

Revision history

Rev	Date	Description
00	<td>	LPC111x preliminary user manual

Contact information

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

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

1. Introduction

The LPC111x are ARM Cortex-M0 based microcontrollers for embedded applications featuring a high level of integration and low power consumption. The ARM Cortex-M0 is a next generation core that offers a simplified instruction set with deterministic behavior.

The LPC111x operate at CPU frequencies of up to 50 MHz.

The peripheral complement of the LPC111x includes up to 32 kB of flash memory, up to 8 kB of data memory, one Fast-mode Plus I²C-bus interface, one RS-485/EIA-485 UART, two SSP interfaces, four general purpose timers, and up to 42 general purpose I/O pins.

2. Features

- ARM Cortex-M0 processor, running at frequencies of up to 50 MHz.
- ARM Cortex-M0 built-in Nested Vectored Interrupt Controller (NVIC).
- 32 kB (LPC1113), 16 kB (LPC1112), or 8 kB (LPC1111) on-chip flash programming memory.
- Up to 8 kB of static RAM.
- In-System Programming (ISP) and In-Application Programming (IAP) via on-chip bootloader software.
- Serial interfaces:
 - UART with fractional baud rate generation, internal FIFO and RS-485/EIA-485 support, and modem control.
 - Two SSP controllers with FIFO and multi-protocol capabilities (second SSP on LQFP48 and PLCC44 packages only).
 - I²C-bus interface supporting the full I²C-bus specification and Fast-mode Plus with a data rate of 1 Mbit/s with multiple address recognition and monitor mode.
- Other peripherals:
 - Up to 42 General Purpose I/O (GPIO) pins with configurable pull-up/pull-down resistors.
 - High-current output driver (20 mA) on one pin.
 - High-current sink drivers (20 mA) on two I²C-bus pins in Fast-mode Plus.
 - Four general purpose timers/counters, with a total of four capture inputs and 13 compare outputs.
 - Watchdog Timer (WDT).
 - System tick timer.
- Serial Wire Debug and Serial Wire Trace Port.
- Integrated PMU (Power Management Unit) automatically adjusts internal regulators to minimize power consumption during Sleep, Deep-sleep, and Deep power-down modes.
- Three reduced power modes: Sleep, Deep-sleep, and Deep power-down.

- Single 3.3 V power supply (2.0 V to 3.6 V).
- 10-bit ADC with input multiplexing among 8 pins.
- GPIO pins can be used as edge and level sensitive interrupt sources.
- Clock output function with divider that can reflect the main oscillator clock, IRC clock, CPU clock, or the Watchdog clock.
- Processor wake-up from Deep-sleep mode via a dedicated start logic using up to 13 functional pins.
- Brownout detect with four separate thresholds for interrupt and one threshold for forced reset.
- Power-On Reset (POR).
- Crystal oscillator with an operating range of 1 MHz to 25 MHz.
- 12 MHz internal RC oscillator trimmed to 1 % accuracy; can optionally be used as a system clock.
- 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 Watchdog oscillator.
- Available in LQFP48, PLCC44, and HVQFN33 packages.

3. Ordering information

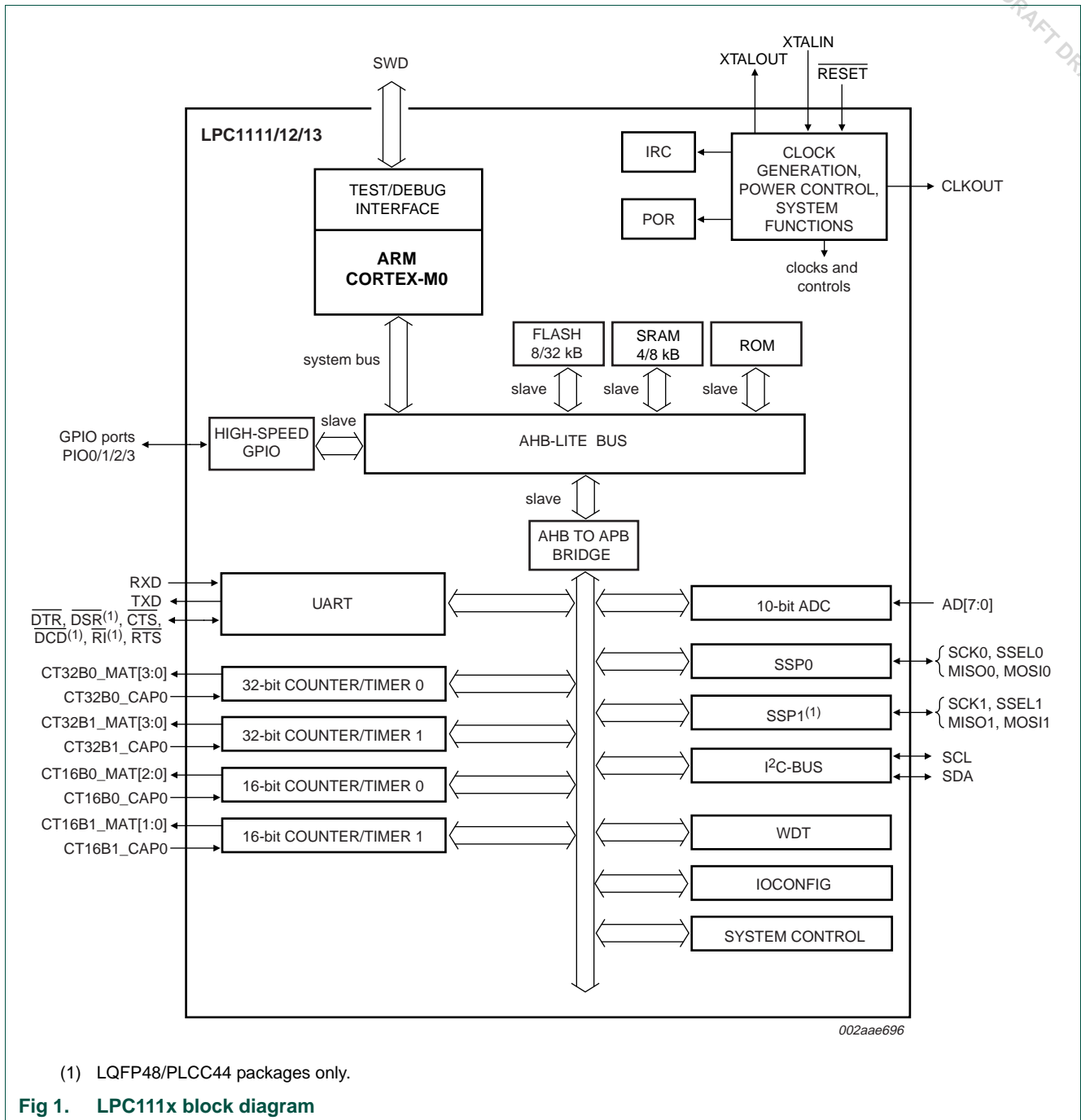
Table 1. Ordering information

Type number	Package		
	Name	Description	Version
LPC1111FHN33	HVQFN33	HVQFN: plastic thermal enhanced very thin quad flat package; no leads; 33 terminals; body 7 x 7 x 0.85 mm	n/a
LPC1112FHN33	HVQFN33	HVQFN: plastic thermal enhanced very thin quad flat package; no leads; 33 terminals; body 7 x 7 x 0.85 mm	n/a
LPC1113FBD48	LQFP48	LQFP48: plastic low profile quad flat package; 48 leads; body 7 x 7 x 1.4 mm	sot313-2
LPC1113FA44	PLCC44	PLCC44; plastic leaded chip carrier; 44 leads	sot187-2
LPC1113FHN33	HVQFN33	HVQFN: plastic thermal enhanced very thin quad flat package; no leads; 33 terminals; body 7 x 7 x 0.85 mm	n/a

Table 2. Ordering options for UM10398

Type number	Flash	Total SRAM	UART RS-485	I ² C/ Fast+	SSP	ADC channels	Package
LPC1111FHN33	8 kB	4 kB	1	1	1	8	HVQFN33
LPC1112FHN33	16 kB	4 kB	1	1	1	8	HVQFN33
LPC1113FBD48	32 kB	8 kB	1	1	2	8	LQFP48
LPC1113FA44	32 kB	8 kB	1	1	2	8	PLCC44
LPC1113FHN33	32 kB	8 kB	1	1	1	8	HVQFN33

4. Block diagram



1. How to read this chapter

[Table 2–3](#) shows the memory configurations for different LPC111x parts.

Table 3. LPC111x memory configuration

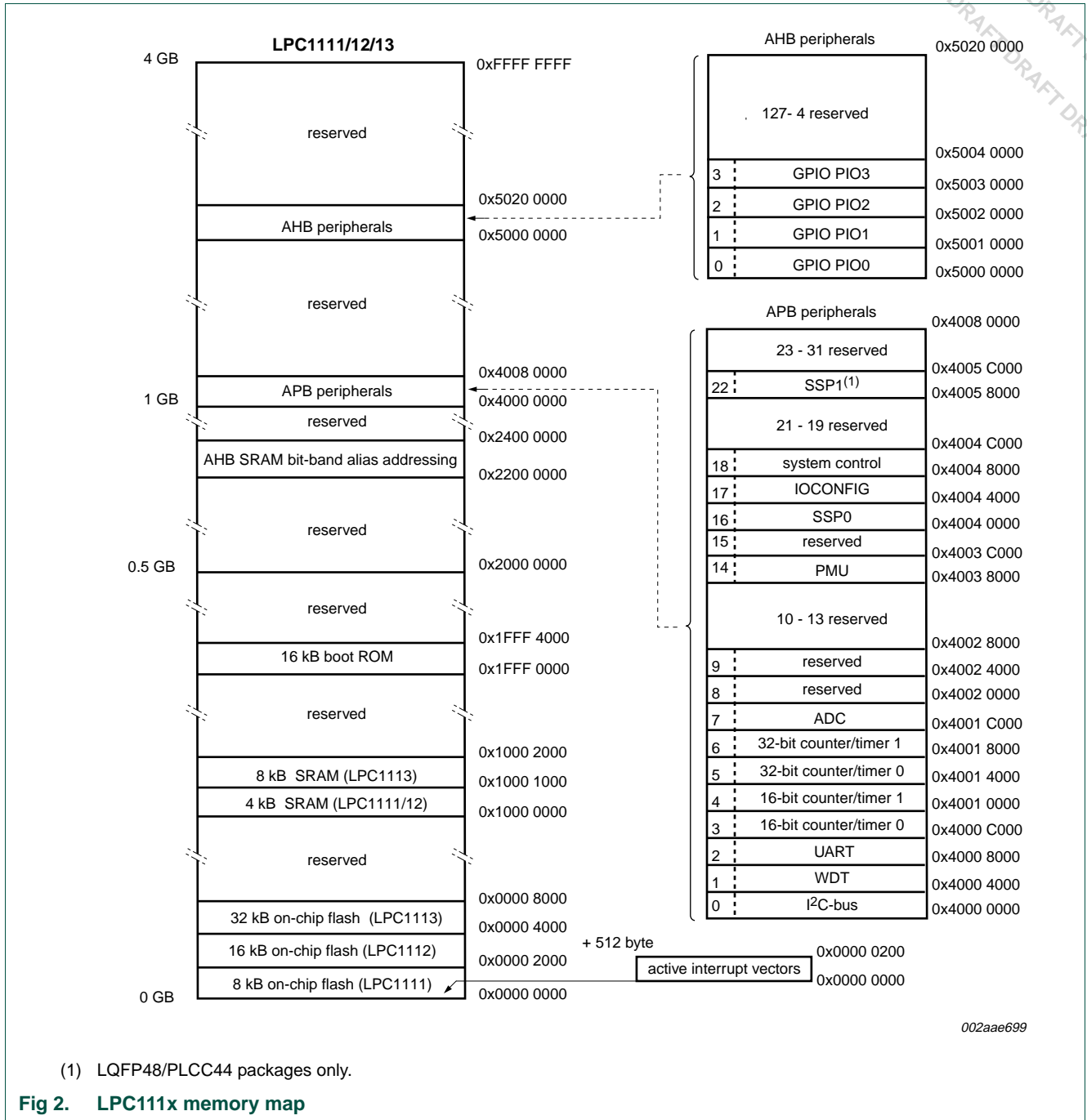
Part	Flash	SRAM
LPC1111	8 kB	4 kB
LPC1112	16 kB	4 kB
LPC1113	32 kB	8 kB

2. Memory map

[Figure 2–2](#) shows the memory and peripheral address space of the LPC111x.

The AHB peripheral area is 2 MB in size and is divided to allow for up to 128 peripherals. On the LPC111x, the GPIO ports are the only AHB peripherals. The APB peripheral area is 512 kB in size and is divided to allow for up to 32 peripherals. Each peripheral of either type is allocated 16 kB of space. This allows simplifying the address decoding for each peripheral.

All peripheral register addresses are 32-bit word aligned regardless of their size. 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.



(1) LQFP48/PLCC44 packages only.

Fig 2. LPC111x memory map

1. Introduction

The system configuration block controls oscillators, start logic, and clock generation of the LPC111x. Also included in this block are registers for setting the priority for AHB access and a register for remapping flash, SRAM, and ROM memory areas.

2. Pin description

[Table 3–4](#) shows pins that are associated with system control block functions.

Table 4. Pin summary

Pin name	Pin direction	Pin description
CLKOUT	O	Clockout pin
PIO0_0 to PIO0_11	I	Start logic wake-up pins port 0
PIO1_0	I	Start logic wake-up pin port 1

3. Clocking and power control

See [Figure 3–3](#) for an overview of the LPC111x Clock Generation Unit (CGU).

The LPC111x include three independent oscillators. These are the system oscillator, the Internal RC oscillator (IRC), and the watchdog oscillator. Each oscillator can be used for more than one purpose as required in a particular application.

Following reset, the LPC111x will operate from the Internal RC oscillator until switched by software. This allows systems to operate without any external crystal and the bootloader code to operate at a known frequency.

The SYSAHBCLKCTRL register gates the system clock to the various peripherals and memories. UART, SSP0/1, and the SysTick timer have individual clock dividers to derive peripheral clocks from the main clock.

The main clock, and the clock outputs from the IRC, the system oscillator, and the watchdog oscillator can be observed directly on the CLKOUT pin.

For details on power control see [Section 3–7](#).

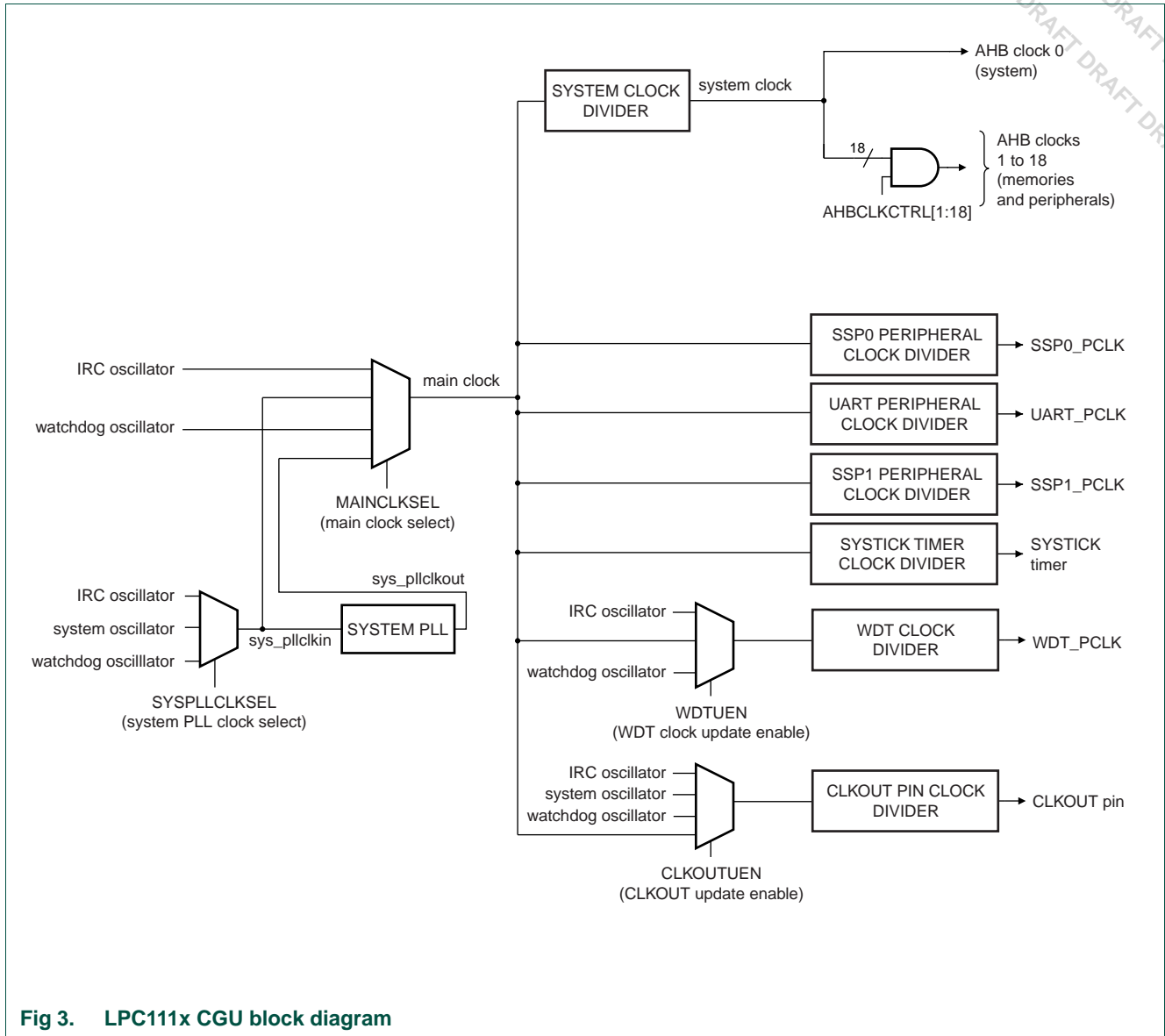


Fig 3. LPC111x CGU block diagram

4. Register description

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

Table 5. Register overview: system control block (base address 0x4004 8000)

Name	Access	Address offset	Description	Reset value	Reference
SYMEMREMAP	R/W	0x000	System memory remap	0x00	Table 3-6
PRESETCTRL	R/W	0x004	Peripheral reset control	0x00	Table 3-7
SYSPLLCTRL	R/W	0x008	System PLL control	0x00	Table 3-8
SYSPLLSTAT	R	0x00C	System PLL status	0x00	Table 3-9
-	-	0x010 - 0x01C	Reserved	-	-

Table 5. Register overview: system control block (base address 0x4004 8000) ...continued

Name	Access	Address offset	Description	Reset value	Reference
SYSOSCCTRL	R/W	0x020	System oscillator control	0x00	Table 3–10
WDTOSCCTRL	R/W	0x024	Watchdog oscillator control	0x00	Table 3–11
IRCCTRL	R/W	0x028	IRC control	0x80	Table 3–12
-	-	0x02C	Reserved	-	-
SYSRESSTAT	R	0x030	System reset status register	0x00	Table 3–13
-	-	0x034 - 0x03C	Reserved	-	-
SYSPLLCLKSEL	R/W	0x040	System PLL clock source select	0x00	Table 3–14
SYSPLLCLKUEN	R/W	0x044	System PLL clock source update enable	0x00	Table 3–15
-	-	0x048 - 0x06C	Reserved	-	-
MAINCLKSEL	R/W	0x070	Main clock source select	0x00	Table 3–16
MAINCLKUEN	R/W	0x074	Main clock source update enable	0x00	Table 3–17
SYSAHBCLKDIV	R/W	0x078	System AHB clock divider	0x01	Table 3–18
-	-	0x07C	Reserved	-	-
SYSAHBCLKCTRL	R/W	0x080	System AHB clock control	0x1F	Table 3–19
-	-	0x084 - 0x090	Reserved	-	-
SSP0CLKDIV	R/W	0x094	SSP0 clock divider	0x00	Table 3–20
UARTCLKDIV	R/W	0x098	UART clock divider	0x00	Table 3–21
SSP1CLKDIV	R/W	0x09C	SSP1 clock divider	0x00	Table 3–22
-	-	0x0A0-0x0AC	Reserved	-	-
SYSTICKCLKDIV	R/W	0x0B0	SYSTICK clock divider	0x00	Table 3–23
-	-	0x0B4 - 0x0CC	Reserved	-	-
WDTCLKSEL	R/W	0x0D0	WDT clock source select	0x00	Table 3–24
WDTCLKUEN	R/W	0x0D4	WDT clock source update enable	0x00	Table 3–25
WDTCLKDIV	R/W	0x0D8	WDT clock divider	0x00	Table 3–26
-	-	0x0DC	Reserved	-	-
CLKOUTCLKSEL	R/W	0x0E0	CLKOUT clock source select	0x00	Table 3–27
CLKOUTUEN	R/W	0x0E4	CLKOUT clock source update enable	0x00	Table 3–28
CLKOUTDIV	R/W	0x0E8	CLKOUT clock divider	0x00	Table 3–29
-	-	0x0EC - 0x0FC	Reserved	-	-
PIOPORCAP0	R	0x100	POR captured PIO status 0	user dependent	Table 3–30
PIOPORCAP1	R	0x104	POR captured PIO status 1	user dependent	Table 3–31
-	R	0x108 - 0x14C	Reserved	-	-
BODCTRL	R/W	0x150	BOD control	0x00	Table 3–32
-	-	0x154	Reserved	-	-
SYSTCKCAL	R/W	0x158	System tick counter calibration	<td>	Table 3–33
-	-	0x15C - 0x1FC	Reserved	-	-
STARTAPRP0	R/W	0x200	Start logic edge control register 0		Table 3–34
STARTERP0	R/W	0x204	Start logic signal enable register 0		Table 3–35
STARTRSRP0CLR	W	0x208	Start logic reset register 0	n/a	Table 3–36

Table 5. Register overview: system control block (base address 0x4004 8000) ...continued

Name	Access	Address offset	Description	Reset value	Reference
STARTSRP0	R	0x20C	Start logic status register 0	n/a	Table 3–37
-	-	0x210 - 0x22C	Reserved	-	-
PDSLEEPCFG	R/W	0x230	Power-down states in Deep-sleep mode	<td>	Table 3–38
PDAWAKECFG	R/W	0x234	Power-down states after wake-up from Deep-sleep mode	<td>	Table 3–39
PDRUNCFG	R/W	0x238	Power-down configuration register	<td>	Table 3–40
-	-	0x23C - 0x3F0	Reserved	-	-
DEVICE_ID	R	0x3F4	Device ID	part dependent	Table 3–41

4.1 System memory remap register

The system memory remap register selects whether the ARM interrupt vectors are read from the boot ROM, the flash, or the SRAM.

Table 6. System memory remap register (SYSTEMREMAP, address 0x4004 8000) bit description

Bit	Symbol	Value	Description	Reset value
1:0	MAP		System memory remap	0x00
		00	Boot Loader Mode. Interrupt vectors are re-mapped to Boot ROM.	
		01	User RAM Mode. Interrupt vectors are re-mapped to Static RAM.	
		10 or 11	User Flash Mode. Interrupt vectors are not re-mapped and reside in Flash.	
31:2	-	-	Reserved	0x00

4.2 Peripheral reset control register

This register allows software to reset the SSP and I2C peripherals. Writing a 0 to the SSP0/1_RST_N or I2C_RST_N bits resets the SSP0/1 or I2C peripheral. Writing a 1 de-asserts the reset.

Table 7. Peripheral reset control register (PRESETCTRL, address 0x4004 8004) bit description

Bit	Symbol	Value	Description	Reset value
0	SSP0_RST_N		SSP0 reset control	0x1
		0	Resets the SSP0 peripheral.	
		1	SSP0 reset de-asserted.	
1	I2C_RST_N		I2C reset control	0x1
		0	Resets the I2C peripheral.	
		1	I2C reset de-asserted.	

Table 7. Peripheral reset control register (PRESETCTRL, address 0x4004 8004) bit description

Bit	Symbol	Value	Description	Reset value
2	SSP1_RST_N		SSP1 reset control	0x1
		0	Resets the SSP1 peripheral.	
		1	SSP1 reset de-asserted.	
31:3	-	-	Reserved	0x00

4.3 System PLL control register

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

The PLL operating mode is set by the DIRECT and BYPASS bits (see [Table 3-43](#)).

Table 8. System PLL control register (SYSPLLCTRL, address 0x4004 8008) bit description

Bit	Symbol	Value	Description	Reset value
4:0	MSEL		Feedback divider value. The division value M is the programmed MSEL value + 1.	0x000
		00000	Division ratio M = 1	
		...		
		11111	Division ration M = 32	
6:5	PSEL		Post divider ratio P. The division ratio is 2 × P.	0x00
		00	P = 1	
		01	P = 2	
		10	P = 4	
		11	P = 8	
7	DIRECT		Direct CCO clock output control	0x0
		0	Clock signal goes through post divider.	
		1	Clock signal goes directly to output(s).	
8	BYPASS		Input clock bypass control.	0x0
		0	CCO clock is sent to post dividers,	
		1	PLL input clock (sys_pllckin) is sent to post dividers.	
31:9	-	-	Reserved	0x00

4.4 System PLL status register

This register is a Read-only register and supplies the PLL lock status (see [Section 3-9.1](#)).

Table 9. System PLL status register (SYSPLLSTAT, address 0x4004 800C) bit description

Bit	Symbol	Value	Description	Reset value
0	LOCK		PLL lock status	0x0
		0	PLL not locked	
		1	PLL locked	
31:1	-	-	Reserved	0x00

4.5 System oscillator control register

This register configures the frequency range for the system oscillator.

Table 10. System oscillator control register (SYSOSCCTRL, address 0x4004 8020) bit description

Bit	Symbol	Value	Description	Reset value
0	BYPASS		Bypass system oscillator	0x0
		0	Oscillator is not bypassed.	
		1	Bypass enabled. PLL input (sys_osc_clk) is fed directly from the XTALIN and XTALOUT pins.	
1	FREQRANGE		Determines frequency range for Low-power oscillator.	0x0
		0	1 - 20 MHz frequency range.	
		1	15 - 25 MHz frequency range	
31:2	-	-	Reserved	0x00

4.6 Watchdog oscillator control register

This register configures the watchdog oscillator. The oscillator consists of an analog and a digital part. The analog part contains the oscillator function and generates an analog clock (Fclkana). With the digital part, the analog output clock (Fclkana) can be divided to the required output clock frequency wdt_osc_clk. The analog output frequency (Fclkana) can be adjusted with the FREQSEL bits between 500 kHz and 3.7 MHz. With the digital part Fclkana will be divided (divider ratios = 2, 4,...,64) to wdt_osc_clk using the DIVSEL bits.

The output clock frequency of the watchdog oscillator can be calculated as $wdt_osc_clk = \frac{Fclkana}{2 \times (1 + DIVSEL)}$. The reset value of the watchdog oscillator is $wdt_osc_clk = 1.6 \text{ MHz} / 6 = 270 \text{ kHz}$.

Remark: Any setting of the FREQSEL bits will yield a Fclkana value within ± 25% of the listed frequency value.

Table 11. Watchdog oscillator control register (WDTOSCCTRL, address 0x4004 8024) bit description

Bit	Symbol	Value	Description	Reset value
4:0	DIVSEL		Select divider for Fclkana to create wdt_osc_clk.	0x000
		00000	2	
		00001	4	
		00010	6	
		
		11111	64	
8:5	FREQSEL		Select watchdog oscillator analog output frequency (Fclkana).	0x05
		0001	0.5 MHz	
		0010	0.8 MHz	
		0011	1.1 MHz	
		0100	1.4 MHz	
		0101	1.6 MHz (Reset value)	
		0110	1.8 MHz	
		0111	2.0 MHz	
		1000	2.2 MHz	
		1001	2.4 MHz	
		1010	2.6 MHz	
		1011	2.7 MHz	
		1100	2.9 MHz	
		1101	3.1 MHz	
1110	3.2 MHz			
1111	3.4 MHz			
31:9	-	-	Reserved	0x00

4.7 Internal resonant crystal control register

This register is used to trim the on-chip 12 MHz oscillator. The trim value is factory-preset and written by the boot code on start-up.

Table 12. Internal resonant crystal control register (IRCCTRL, address 0x4004 8028) bit description

Bit	Symbol	Value	Description	Reset value
7:0	TRIM		Trim value	0x1000 0000, then flash will reprogram
31:9	-	-	Reserved	0x00

4.8 System reset status register

The SYSRSTSTAT register shows the source of the latest reset event. The bits are cleared by writing a one to any of the bits. The POR event clears all other bits in this register, but if another reset signal (e.g., EXTRST) remains asserted after the POR signal is negated, then its bit is set to detected.

Table 13. System reset status register (SYSRSTSTAT, address 0x4004 8030) bit description

Bit	Symbol	Value	Description	Reset value
0	POR		POR reset status	0x0
		0	no POR detected	
		1	POR detected	
1	EXTRST		Status of the external $\overline{\text{RESET}}$ pin	0x0
		0	$\overline{\text{no RESET}}$ event detected	
		1	$\overline{\text{RESET}}$ detected	
2	WDT		Status of the Watchdog reset	0x0
		0	no WDT reset detected	
		1	WDT reset detected	
3	BOD		Status of the Brown-out detect reset	0x0
		0	no BOD reset detected	
		1	BOD reset detected	
4	SYSRST		Status of the software system reset	0x0
		0	no System reset detected	
		1	System reset detected	
31:5	-	-	Reserved	0x00

4.9 System PLL clock source select register

This register selects the clock source for the system PLL. The SYSPLLCLKUEN register (see [Section 3–4.10](#)) must be toggled from LOW to HIGH for the update to take effect.

Table 14. System PLL clock source select register (SYSPLLCLKSEL, address 0x4004 8040) bit description

Bit	Symbol	Value	Description	Reset value
1:0	SEL		System PLL clock source	0x00
		00	IRC oscillator	
		01	System oscillator	
		10	WDT oscillator	
		11	Reserved	
31:2	-	-	Reserved	0x00

4.10 System PLL clock source update enable register

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

Table 15. System PLL clock source update enable register (SYSPLLUEN, address 0x4004 8044) bit description

Bit	Symbol	Value	Description	Reset value
0	ENA		Enable system PLL clock source update	0x0
		0	No change	
		1	Update clock source	
31:1	-	-	Reserved	0x00

4.11 Main clock source select register

This register selects the main system clock which can be either any input to the system PLL, the output from the system PLL (sys_pllclkout), or the watchdog or IRC oscillators directly. The main system clock clocks the core, the peripherals, and the memories.

The MAINCLKUEN register (see [Section 3–4.12](#)) must be toggled from LOW to HIGH for the update to take effect.

Table 16. Main clock source select register (MAINCLKSEL, address 0x4004 8070) bit description

Bit	Symbol	Value	Description	Reset value
1:0	SEL		Clock source for main clock	0x00
		00	IRC oscillator	
		01	Input clock to system PLL	
		10	WDT oscillator	
		11	System PLL clock out	
31:2	-	-	Reserved	0x00

4.12 Main clock source update enable register

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

Table 17. Main clock source update enable register (MAINCLKUEN, address 0x4004 8074) bit description

Bit	Symbol	Value	Description	Reset value
0	ENA		Enable main clock source update	0x0
		0	No change	
		1	Update clock source	
31:1	-	-	Reserved	0x00

4.13 System AHB clock divider register

This register divides the main clock to provide the system clock to the core, memories, and the peripherals. The system clock can be shut down completely by setting the DIV bits to 0x0.

Table 18. System AHB clock divider register (SYSAHBCLKDIV, address 0x4004 8078) bit description

Bit	Symbol	Value	Description	Reset value
7:0	DIV		System AHB clock divider values	0x01
		0	System clock disabled.	
		1	Divide by 1	
		to ...		
		255	Divide by 255	
31:8	-	-	Reserved	0x00

4.14 System AHB clock control register

The AHBCLKCTRL register enables the clocks to individual system and peripheral blocks. The system clock (sys_ahb_clk[0], bit 0 in the AHBCLKCTRL register) provides the clock for the AHB to APB bridge, the AHB matrix, the ARM Cortex-M0, the Syscon block, and the PMU. This clock cannot be disabled.

Table 19. System AHB clock control register (AHBCLKCTRL, address 0x4004 8080) bit description

Bit	Symbol	Value	Description	Reset value
0	SYS		Enables clock for AHB to APB bridge, to the AHB matrix, to the Cortex-M0 FCLK and HCLK, to the SysCon, and to the PMU. This bit is read only.	1
		0	Reserved	
		1	Enable	
1	ROM	0	Disables clock for ROM.	1
		1	Enable	
2	RAM	0	Disables clock for RAM.	1
		1	Enable	
3	FLASHREG	0	Disables clock for flash register interface.	1
		1	Enabled	
4	FLASHARRAY	0	Disables clock for flash array access.	1
		1	Enabled	
5	I2C	0	Disables clock for I2C.	0
		1	Enable	

Table 19. System AHB clock control register (AHBCLKCTRL, address 0x4004 8080) bit description ...continued

Bit	Symbol	Value	Description	Reset value
6	GPIO		Enables clock for GPIO.	0
		0	Disable	
		1	Enable	
7	CT16B0		Enables clock for 16-bit counter/timer 0.	0
		0	Disable	
		1	Enable	
8	CT16B1		Enables clock for 16-bit counter/timer 1.	0
		0	Disable	
		1	Enable	
9	CT32B0		Enables clock for 32-bit counter/timer 0.	0
		0	Disable	
		1	Enable	
10	CT32B1		Enables clock for 32-bit counter/timer 1.	0
		0	Disable	
		1	Enable	
11	SSP0		Enables clock for SSP0.	0
		0	Disable	
		1	Enable	
12	UART		Enables clock for UART. Note that the UART pins must be configured in the IOCON block before the UART clock can be enabled.	0
		0	Disable	
		1	Enable	
13	ADC		Enables clock for ADC.	0
		0	Disable	
		1	Enable	
14	-		Reserved	0
15	WDT		Enables clock for WDT.	0
		0	Disable	
		1	Enable	
16	IOCON		Enables clock for IO configuration block.	0
		0	Disable	
		1	Enable	
17	-	-	Reserved	0
18	SSP1		Enables clock for SSP1.	0
		0	Disable	
		1	Enable	
31:19	-	-	Reserved	0x00

4.15 SSP0 clock divider register

This register configures the SSP0 peripheral clock SSP0_PCLK. The SSP0_PCLK can be shut down by setting the DIV bits to 0x0.

Table 20. SSP0 clock divider register (SSP0CLKDIV, address 0x4004 8094) bit description

Bit	Symbol	Value	Description	Reset value
7:0	DIV		SSP0_PCLK clock divider values	0x00
		0	Disable SSP0_PCLK.	
		1	Divide by 1.	
		to	...	
		255	Divide by 255.	
31:8	-	-	Reserved	0x00

4.16 UART clock divider register

This register configures the UART peripheral clock UART_PCLK. The UART_PCLK can be shut down by setting the DIV bits to 0x0.

Remark: Note that the UART pins must be configured in the IOCON block before the UART clock can be enabled.

Table 21. UART clock divider register (UARTCLKDIV, address 0x4004 8098) bit description

Bit	Symbol	Value	Description	Reset value
7:0	DIV		UART_PCLK clock divider values	0x00
		0	Disable UART_PCLK.	
		1	Divide by 1.	
		to	...	
		255	Divide by 255.	
31:8	-	-	Reserved	0x00

4.17 SSP1 clock divider register

This register configures the SSP1 peripheral clock SSP1_PCLK. The SSP1_PCLK can be shut down by setting the DIV bits to 0x0.

Table 22. SSP1 clock divider register (SSP1CLKDIV, address 0x4004 809C) bit description

Bit	Symbol	Value	Description	Reset value
7:0	DIV		SSP1_PCLK clock divider values	0x00
		0	Disable SSP1_PCLK.	
		1	Divide by 1.	
		to	...	
		255	Divide by 255.	
31:8	-	-	Reserved	0x00

4.18 SYSTICK clock divider register

This register configures the SYSTICK peripheral clock. The SYSTICK timer clock can be shut down by setting the DIV bits to 0x0.

Table 23. SYSTICK clock divider register (SYSTICKCLKDIV, address 0x4004 80B0) bit description

Bit	Symbol	Value	Description	Reset value
7:0	DIV		SYSTICK clock divider values	0x00
		0	Disable SYSTICK timer clock.	
		1	Divide by 1.	
		to	...	
		255	Divide by 255.	
31:8	-	-	Reserved	0x00

4.19 WDT clock source select register

This register selects the clock source for the watchdog timer. The WDTCLKUEN register (see [Section 3-4.20](#)) must be toggled from LOW to HIGH for the update to take effect.

Table 24. WDT clock source select register (WDTCLKSEL, address 0x4004 80D0) bit description

Bit	Symbol	Value	Description	Reset value
1:0	SEL		WDT clock source	0x00
		00	IRC oscillator	
		01	Main clock	
		10	Watchdog oscillator	
		11	Reserved	
31:2	-	-	Reserved	0x00

4.20 WDT clock source update enable register

This register updates the clock source of the watchdog timer with the new input clock after the WDTCLKSEL register has been written to. In order for the update to take effect at the input of the watchdog timer, first write a zero to the WDTCLKUEN register and then write a one to WDTCLKUEN.

Table 25. WDT clock source update enable register (WDTCLKUEN, address 0x4004 80D4) bit description

Bit	Symbol	Value	Description	Reset value
0	ENA		Enable WDT clock source update	0x0
		0	No change	
		1	Update clock source	
31:1	-	-	Reserved	0x00

4.21 WDT clock divider register

This register determines the divider values for the watchdog clock wdt_clk.

Table 26. WDT clock divider register (WDTCLKDIV, address 0x4004 80D8) bit description

Bit	Symbol	Value	Description	Reset value
7:0	DIV		WDT clock divider values	0x00
		0	Gate	
		1	Divide by 1	
		to	...	
		255	Divide by 255	
31:8	-	-	Reserved	0x00

4.22 CLKOUT clock source select register

This register configures the `clkout_clk` signal to be output on the CLKOUT pin. All three oscillators and the main clock can be selected for the `clkout_clk` clock.

The CLKOUTCLKUEN register (see [Section 3–4.23](#)) must be toggled from LOW to HIGH for the update to take effect.

Table 27. CLKOUT clock source select register (CLKOUTCLKSEL, address 0x4004 80E0) bit description

Bit	Symbol	Value	Description	Reset value
1:0	SEL		CLKOUT clock source	0x00
		00	IRC oscillator	
		01	System oscillator	
		10	Watchdog oscillator	
		11	Main clock	
31:2	-	-	Reserved	0x00

4.23 CLKOUT clock source update enable register

This register updates the clock source of the CLKOUT pin with the new clock after the CLKOUTCLKSEL register has been written to. In order for the update to take effect at the input of the CLKOUT pin, first write a zero to the CLKCLKUEN register and then write a one to CLKCLKUEN.

Table 28. CLKOUT clock source update enable register (CLKOUTUEN, address 0x4004 80E4) bit description

Bit	Symbol	Value	Description	Reset value
0	ENA		Enable CLKOUT clock source update	0x0
		0	No change	
		1	Update clock source	
31:1	-	-	Reserved	0x00

4.24 CLKOUT clock divider register

This register determines the divider value for the `clkout_clk` signal on the CLKOUT pin.

Table 29. CLKOUT clock divider registers (CLKOUTCLKDIV, address 0x4004 80E8) bit description

Bit	Symbol	Value	Description	Reset value
7:0	DIV		Clock divider values	0x00
		0	Gate	
		1	Divide by 1	
		to	...	
		255	Divide by 255	
31:8	-	-	Reserved	0x00

4.25 POR captured PIO status register 0

The PIOPORCAP0 register captures the state (HIGH or LOW) of the PIO pins of ports 0, 1, and 2 (pins PIO2_0 to PIO2_7) at power-on-reset. Each bit represents the reset state of one GPIO pin. This register is a read-only status register.

Table 30. POR captured PIO status registers 0 (PIOPORCAP0, address 0x4004 8100) bit description

Bit	Symbol	Description	Reset value
0	CAPPIO0_0	Raw reset status input PIO0_0	User implementation dependent
1	CAPPIO0_1	Raw reset status input PIO0_1	User implementation dependent
11:2	CAPPIO0_11 to CAPPIO0_2	Raw reset status input PIO0_11 to PIO0_2	User implementation dependent
23:12	CAPPIO1_11 to CAPPIO1_0	Raw reset status input PIO1_11 to PIO1_0	User implementation dependent
31:24	CAPPIO2_7 to CAPPIO2_0	Raw reset status input PIO2_7 to PIO2_0	User implementation dependent

4.26 POR captured PIO status register 1

The PIOPORCAP1 register captures the state (HIGH or LOW) of the PIO pins of port 2 (PIO2_8 to PIO2_11) and port 3 at power-on-reset. Each bit represents the reset state of one PIO pin. This register is a read-only status register.

Table 31. POR captured PIO status registers 1 (PIOPORCAP1, address 0x4004 8104) bit description

Bit	Symbol	Description	Reset value
0	CAPPIO2_8	Raw reset status input PIO2_8	User implementation dependent
1	CAPPIO2_9	Raw reset status input PIO2_9	User implementation dependent
2	CAPPIO2_10	Raw reset status input PIO2_10	User implementation dependent
3	CAPPIO2_11	Raw reset status input PIO2_11	User implementation dependent
4	CAPPIO3_0	Raw reset status input PIO3_0	User implementation dependent
5	CAPPIO3_1	Raw reset status input PIO3_1	User implementation dependent
6	CAPPIO3_2	Raw reset status input PIO3_2	User implementation dependent
7	CAPPIO3_3	Raw reset status input PIO3_3	User implementation dependent

Table 31. POR captured PIO status registers 1 (PIOPORCAP1, address 0x4004 8104) bit description

Bit	Symbol	Description	Reset value
8	CAPPIO3_4	Raw reset status input PIO3_4	User implementation dependent
9	CAPPIO3_5	Raw reset status input PIO3_5	User implementation dependent
31:10	-	Reserved	-

4.27 BOD control register

The BOD control register selects four separate threshold values for sending a BOD interrupt to the NVIC. Only one level is allowed for forced reset.

Table 32. BOD control register (BODCTRL, address 0x4004 8150) bit description

Bit	Symbol	Value	Description	Reset value
1:0	BODRSTLEV		BOD reset level	0x00
		00	The reset assertion threshold voltage is 1.49 V; the reset de-assertion threshold voltage is 1.64 V.	
		01 -11	Reserved	
3:2	BODINTVAL		BOD interrupt level	0x00
		00	The interrupt assertion threshold voltage is 1.69 V; the interrupt de-assertion threshold voltage is 1.84 V.	
		01	The interrupt assertion threshold voltage is 2.29 V; the interrupt de-assertion threshold voltage is 2.44 V.	
		10	The interrupt assertion threshold voltage is 2.59 V; the interrupt de-assertion threshold voltage is 2.74 V.	
		11	The interrupt assertion threshold voltage is 2.87 V; the interrupt de-assertion threshold voltage is 2.98 V.	
4	BODRSTENA		BOD reset enable	0x0
		0	Disable reset function.	
		1	Enable reset function.	
31:5	-	-	Reserved	0x00

4.28 System tick counter calibration register

Table 33. System tick timer calibration register (SYSTCKCAL, address 0x4004 8158) bit description

Bit	Symbol	Value	Description	Reset value
25:0	CAL		System tick timer calibration value	<tb>
31:26	-	-	Reserved	0x00

4.29 Start logic edge control register 0

The STARTAPRP0 register controls the start logic inputs of ports 0 (PIO0_0 to PIO0_11) and 1 (PIO1_0). This register selects a falling or rising edge on the corresponding PIO input to produce a falling or rising clock edge, respectively, for the start logic (see [Section 3–8.3](#)).

Every bit in the STARTAPRP0 register controls one port input and is connected to one wake-up interrupt in the NVIC. Bit 0 in the STARTAPRP0 register corresponds to interrupt 0, bit 1 to interrupt 1, etc. (see [Table 5–49](#)), up to a total of 13 interrupts.

Remark: Each interrupt connected to a start logic input must be enabled in the NVIC if the corresponding PIO pin is used to wake up the chip from Deep-sleep mode.

Table 34. Start logic edge control register 0 (STARTAPRP0, address 0x4004 8200) bit description

Bit	Symbol	Value	Description	Reset value
11:0	APRPIO0_11 to APRPIO0_0	0	Falling edge	0x0
		1	Rising edge	
12	APRPIO1_0	0	Falling edge	0x0
		1	Rising edge	
31:13	-	-	Reserved	0x0

4.30 Start logic signal enable register 0

This STARTERP0 register enables or disables the start signal bits in the start logic. The bit assignment is identical to [Table 3–34](#).

Table 35. Start logic signal enable register 0 (STARTERP0, address 0x4004 8204) bit description

Bit	Symbol	Value	Description	Reset value
11:0	ERPIO0_11 to ERPIO0_0	0	Disabled	0x0
		1	Enabled	
12	ERPIO1_0	0	Disabled	0x0
		1	Enabled	
31:13	-	-	Reserved	0x0

4.31 Start logic reset register 0

Writing a one to a bit in the STARTRSRP0CLR register resets the start logic state. The bit assignment is identical to [Table 3–34](#). The start-up logic uses the input signals to generate a clock edge for registering a start signal. This clock edge (falling or rising) sets the interrupt for waking up from Deep-sleep mode. Therefore, the start-up logic states must be cleared before being used.

Table 36. Start logic reset register 0 (STARTRSRP0CLR, address 0x4004 8208) bit description

Bit	Symbol	Value	Description	Reset value
11:0	RSRPIO0_11 to RSRPIO0_0	0	Start signal reset for start logic input PIO0_11 to PIO0_0	n/a
		1	-	
12	RSRPIO1_0	0	Start signal reset for start logic input PIO1_0	n/a
		1	-	
31:13	-	-	Reserved	n/a

4.32 Start logic status register 0

This register reflects the status of the enabled start signal bits. The bit assignment is identical to [Table 3–34](#). Each bit (if enabled) reflects the state of the start logic, i.e. whether or not a wake-up signal has been received for a given pin.

Table 37. Start logic status register 0 (STARTSRP0, address 0x4004 820C) bit description

Bit	Symbol	Value	Description	Reset value
11:0	SRPIO0_11 to SRPIO0_0	0	Start signal status for start logic input PIO0_11 to PIO0_0	n/a
		1	No start signal received	
12	SRPIO1_0	0	Start signal status for start logic input PIO1_0	n/a
		1	No start signal received	
31:13	-	-	Reserved	n/a

4.33 Deep-sleep mode configuration register

The bits in this register can be programmed to indicate the state the chip must enter when the Deep-sleep mode is asserted by the ARM. The value of the PDSLEEPCFG register will be automatically loaded into the PDRUNCFG register when the Sleep mode is entered.

Table 38. Deep-sleep configuration register (PDSLEEPCFG, address 0x4004 8230) bit description

Bit	Symbol	Value	Description	Reset value
0	IRCOUT_PD	0	IRC oscillator output power-down control in Deep-sleep mode	0
		1	Powered down	
		0	Powered	

Table 38. Deep-sleep configuration register (PDSLEEPCFG, address 0x4004 8230) bit description ...continued

Bit	Symbol	Value	Description	Reset value
1	IRC_PD		IRC oscillator power-down control in Deep-sleep mode	0
		1	Powered down	
		0	Powered	
2	FLASH_PD		Flash power-down control in Deep-sleep mode	0
		1	Powered down	
		0	Powered	
3	BOD_PD		BOD power-down control in Deep-sleep mode	0
		1	Powered down	
		0	Powered	
4	ADC_PD		ADC power-down control in Deep-sleep mode	0
		1	Powered down	
		0	Powered	
5	SYSOSC_PD		System oscillator power-down control in Deep-sleep mode	0
		1	Powered down	
		0	Powered	
6	WDTOSC_PD		Watchdog oscillator power-down control in Deep-sleep mode	0
		1	Powered down	
		0	Powered	
7	SYSPLL_PD		System PLL power-down control in Deep-sleep mode	0
		1	Powered down	
		0	Powered	
8	-	-	Reserved	<tbd>
9	MAINREG_PD		Main regulator power mode in Deep-sleep mode	<tbd>
		1	Main regulator in reduced power mode	
		0	Main regulator in normal power mode	
10	-	-	Reserved	<tbd>
31:11	-	-	Reserved	0

4.34 Wake-up configuration register

The bits in this register can be programmed to indicate the state the chip must enter when it is waking up from Deep-sleep mode.

Table 39. Wake-up configuration register (PDAWAKECFG, address 0x4004 8234) bit description

Bit	Symbol	Value	Description	Reset value
0	IRCOUT_PD		IRC oscillator output wake-up configuration	0
		1	Powered down	
		0	Powered	
1	IRC_PD		IRC oscillator power-down wake-up configuration	0
		1	Powered down	
		0	Powered	
2	FLASH_PD		Flash wake-up configuration	0
		1	Powered down	
		0	Powered	
3	BOD_PD		BOD wake-up configuration	0
		1	Powered down	
		0	Powered	
4	ADC_PD		ADC wake-up configuration	1
		1	Powered down	
		0	Powered	
5	SYSOSC_PD		System oscillator wake-up configuration	1
		1	Powered down	
		0	Powered	
6	WDTOSC_PD		Watchdog oscillator wake-up configuration	1
		1	Powered down	
		0	Powered	
7	SYSPLL_PD		System PLL wake-up configuration	1
		1	Powered down	
		0	Powered	
8	-	-	Reserved	<tbd>
9	MAINREG_PD		Main regulator power mode wake-up configuration	<tbd>
		1	Main regulator in reduced power mode	
		0	Main regulator in normal power mode	
10	-	-	Reserved	<tbd>
31:11	-	-	Reserved	0

4.35 Power-down configuration register

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

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

Table 40. Power-down configuration register (PDRUNCFG, address 0x4004 8238) bit description

Bit	Symbol	Value	Description	Reset value
0	IRCOUT_PD		IRC oscillator output power-down	0
		1	Powered down	
		0	Powered	
1	IRC_PD		IRC oscillator power-down	0
		1	Powered down	
		0	Powered	
2	FLASH_PD ^[1]		Flash power-down	0
		1	Powered down	
		0	Powered	
3	BOD_PD		BOD power-down	0
		1	Powered down	
		0	Powered	
4	ADC_PD		ADC power-down	1
		1	Powered down	
		0	Powered	
5	SYSOSC_PD		System oscillator power-down	1
		1	Powered down	
		0	Powered	
6	WDTOSC_PD		Watchdog oscillator power-down	1
		1	Powered down	
		0	Powered	
7	SYSPLL_PD		System PLL power-down	1
		1	Powered down	
		0	Powered	
8	-	-	Reserved	<tbd>
9	MAINREG_PD		Main regulator power mode	<tbd>
		1	Main regulator in reduced power mode	
		0	Main regulator in normal power mode	
10	-	-	Reserved	<tbd>
31:11	-	-	Reserved	0

[1] The flash power-up sequence for waking up from Deep-sleep mode takes 100 μs. Note that the flash does not need to be initialized in this case. If the flash is powered down, the user must wait for this time period before resuming flash operations. The power-up sequence after reset takes slightly longer to allow for the flash to initialize.

4.36 Device ID register

This device ID register is a read-only register and contains the device ID for each LPC111x part. This register is also read by the ISP/IAP commands (see [Section 17–7.11](#) and [Section 17–8.9](#)).

Table 41. Device ID register (DEVICE_ID, address 0x4004 83F4) bit description

Bit	Symbol	Value	Description	Reset value
31:0	DEVICEID	Device ID for LPC111x parts <td>		part-dependent

5. Reset

Reset has four sources on the LPC111x: the $\overline{\text{RESET}}$ pin, Watchdog Reset, Power-On Reset (POR), and Brown Out Detect (BOD). In addition, there is a software reset.

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

On the assertion of a reset source external to the Cortex-M0 CPU (POR, BOD reset, External reset, and Watchdog reset), the IRC starts up. After the IRC-start-up time (maximum of 6 μs on power-up), the IRC provides a stable clock output

1. The boot code in the ROM starts. The boot code performs the boot tasks and may jump to the flash.
2. The flash is powered up. This takes approximately 100 μs . Then the flash initialization sequence is started, which takes about 250 cycles.

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

6. Brown-out detection

The LPC111x includes four levels for monitoring the voltage on the $V_{\text{DD}(3\text{V}3)}$ pin. If this voltage falls below one of the four selected levels, 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 a dedicated status register. An additional threshold level can be selected to cause a forced reset of the chip.

7. Power management

The LPC111x support a variety of power control features. Power and clocks to selected blocks of the LPC111x can be optimized for power consumption when the chip is running.

In addition, there are three special modes of processor power reduction: Sleep mode, Deep-sleep mode, and Deep power-down mode. The PMU controls whether the Sleep mode or the Deep power-down mode is entered (see [Section 4–2.1](#)). For Sleep mode, the user can configure the remaining power-consumption to a large extent by selecting various analog blocks as well as the flash and the oscillators to remain powered or to be shut down. This mode with reduced power to the analog blocks is called the Deep-sleep mode.

The CPU clock rate may also be controlled as needed by changing clock sources, re-configuring PLL values, and/or altering the system clock divider value. This allows a trade-off of power versus processing speed based on application requirements.

Run-time 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. Selected peripherals (UART, SSPs, SysTick timer, Watchdog timer) have their own clock divider which provides even better power control.

Table 42. LPC111x power and clock control options

Register	Power/clock control function	Applies to modes
Power control		
PDRUNCFG	Table 3–40 Controls power to ADC and analog blocks in any mode (running or Deep-sleep).	Run, Deep-sleep
PDSLEEPCFG	Table 3–38 Selects which analog blocks are shut down in Deep-sleep mode. The contents of this register will get loaded into PDRUNCFG when the chip enters Sleep mode.	Deep-sleep
PDAWAKECFG	Table 3–39 Selects which analog blocks are powered when the chip wakes up from Deep-sleep mode. The contents of this register will get loaded into PDRUNCFG when the chip exits Deep-sleep mode.	Run
Clock control		
AHBCLKCTRL	Table 3–19 Controls clocks to the ARM Cortex-M0 CPU, memories, and individual APB peripherals.	Run
SysAHBCLKDIV	Table 3–18 Disables or configures the system clock.	Run
SSP0CLKDIV	Table 3–20 Disables or configures the SSP0 peripheral clock.	Run
UARTCLKDIV	Table 3–21 Disables or configures the UART peripheral clock.	Run
SSP1CLKDIV	Table 3–22 Disables or configures the SSP1 peripheral clock.	Run
WDTCLKDIV	Table 3–26 Disables or configures the watchdog timer clock.	Run
CLKOUTDIV	Table 3–29 Disables or configures the clock on the CLKOUT pin.	Run
Power-down modes control (PMU)		
PCON	Table 4–46 Controls which power-down mode is entered.	Sleep, Deep power-down

7.1 Sleep mode

When Sleep mode is entered, the clock to the ARM Cortex-M0 core is stopped. The Sleep mode is entered by using the ARM Cortex-M0 Wait-for-Interrupt (WFI) instruction. Sleep mode is exited automatically when an interrupt arrives at the processor.

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. Resumption from the Sleep mode does not need any special sequence other than re-enabling the clock to the ARM core.

7.2 Deep-sleep mode

In Deep-sleep mode (see [Section 3–8](#)), the chip is in Sleep mode and in addition selected analog blocks are powered down. The user can configure which blocks to power down when Sleep mode is entered and which blocks will be running on wake-up from Deep-sleep mode.

The advantage of the Deep-sleep mode is that the user can power down clock generating blocks such as oscillators and PLLs thereby gaining far greater dynamic power savings over Sleep mode. In addition, the flash may be powered down in Deep-sleep mode resulting in savings in static leakage power - however at the expense of longer wake-up times for the flash memory.

The LPC111x uses a start logic (see [Section 3–8.3](#)) that connects the general purpose PIO inputs to the NVIC to wake up from Deep-sleep mode. The Deep-sleep mode is exited when the start logic indicates an interrupt to the ARM Cortex-M0.

7.3 Deep power-down mode

In Deep power-down mode, power is shut off to the entire chip with the exception of the WAKEUP pin, which the LPC111x can use to wake up from Deep power-down. During Deep power-down mode, the chip can retain data in four general purpose registers. For details, see [Section 4–1](#).

8. Deep-sleep mode

In Deep-sleep mode, the clock to the ARM core is switched off (the LPC111x is in Sleep mode), and in addition various analog blocks can be selected for power-down. Entering Deep-sleep mode is controlled by the Deep-sleep negotiator, which is part of the ARM Cortex-M0 core, and the Deep-sleep finite state machine. The wake-up process from Deep-sleep mode is initiated by the start logic. After wake-up, the power state of the analog blocks is determined by the PDAWAKECFG register.

8.1 Entering Deep-sleep mode

The Deep-sleep negotiator causes the LPC111x to hold entering Deep-sleep mode until the ARM Cortex-M0 core acknowledges the sleep hold request. During the hold time, the ARM core can still exit the Power-down sequence. Furthermore, the ARM core can choose to de-assert the hold request during sleep mode (e.g. if required to do so by the debugger) in which case the Deep-sleep request will be de-asserted as well.

The Deep-sleep finite state machine ensures that while entering Deep-sleep mode, the start logic's wake-up signals are ignored. This guarantees that the Deep-sleep mode is not entered for too short a time, which could cause a glitch on the Power-down signals.

Once the LPC111x Deep-sleep request is asserted, the Syscon block will power down the core, the PDRUNCFG register will be loaded with the PDSLEEPCFG value, and the selected analog blocks will be powered down on subsequent clock edges. After a further 30 ns delay, the LPC111x is in Deep-sleep mode and can now accept start signals from the start logic to wake up.

Remark: If the IRC is selected for power-down, the Deep-sleep finite state machine will wait for a signal asserting that the IRC has been switched off safely before starting the 30 ns delay time (see [Section 3–8.2](#)).

8.2 Powering down the 12 MHz IRC oscillator

The IRC employs a mechanism that ensures that the 12 MHz oscillator is always switched off without a glitch. Once the 12 MHz oscillator is switched off (within two 12 MHz clock cycles), an acknowledge signal will be sent to the Syscon block.

Remark: The IRC is the only oscillator on the LPC111x that can always shut down glitch-free. Therefore it is recommended that the user switches the clock source to the 12 MHz IRC before the chip enters Deep-sleep mode - unless another clock source is selected to remain powered during Deep-sleep mode.

8.3 Start logic

The Deep-sleep mode is exited when the start logic indicates an interrupt to the ARM core. All PIO port inputs except PIO3_4 and PIO3_5 are connected to the start logic and serve as wake-up pins. The user must program the start logic registers for each input to set the appropriate edge polarity for the corresponding wake-up event. Furthermore, the interrupts corresponding to each input must be enabled in the NVIC. Interrupts 0 to 39 in the NVIC correspond to 40 PIO pins (see [Section 3–4.29](#)).

The start logic does not require a clock to run because it uses the PIO input signals to generate a clock edge when enabled. Therefore, the start logic signals should be cleared (see [Table 3–36](#)) before use.

The start logic can also be used in normal run mode (i.e. not in Sleep or Deep-sleep mode) to provide a vectored interrupt using the LPC111x's input pins.

9. System PLL functional description

The LPC111x uses the system PLL to create the clocks for the core and peripherals.

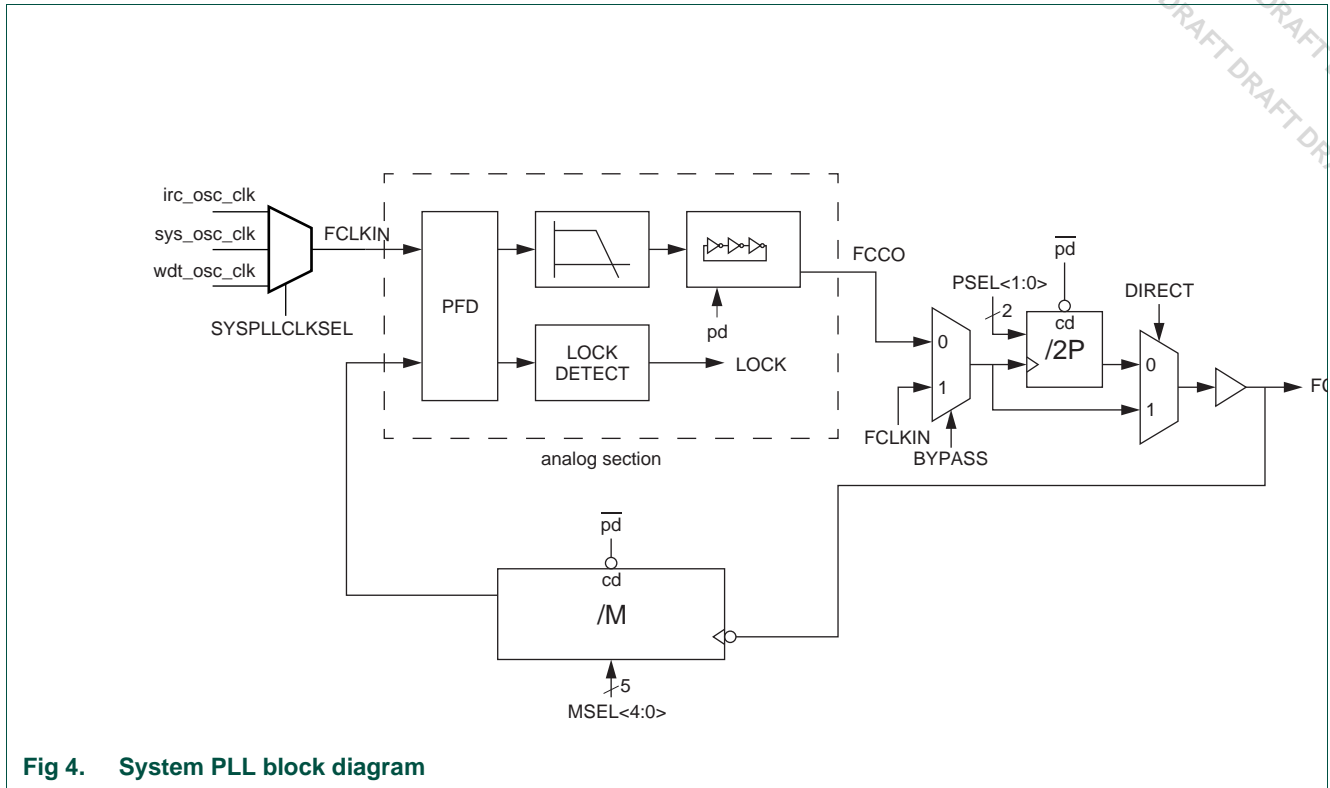


Fig 4. System PLL block diagram

The block diagram of this PLL is shown in [Figure 3-4](#). The input frequency range is 10 MHz to 25 MHz. The input clock is fed directly to the Phase-Frequency Detector (PFD). This block compares the phase and frequency of its inputs, and generates a control signal when phase and/ or frequency do not match. The loop filter filters these control signals and drives the current controlled oscillator (CCO), which generates the main clock and optionally two additional phases. The CCO frequency range is 156 MHz to 320 MHz. These clocks are either divided by $2 \times P$ by the programmable post divider to create the output clock(s), or are sent directly to the output(s). The main output clock is then divided by M by the programmable feedback divider to generate the feedback clock. The output signal of the phase-frequency detector is also monitored by the lock detector, to signal when the PLL has locked on to the input clock.

9.1 Lock detector

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

9.2 Direct output mode

In normal operating mode (with the DIRECT bit set to '0') the CCO clock is divided by 2, 4, 8 or 16 depending on the value on the PSEL bits, giving an output clock with a 50% duty cycle. If a higher output frequency is needed, the CCO clock can be sent directly to the output by setting direct to '1'. As the CCO does not directly generate a 50% duty cycle clock, the output clock duty cycle in this mode can deviate from 50%.

9.3 Power-down control

To reduce the power consumption when the PLL clock is not needed, a Power-down mode has been incorporated. This mode is enabled by setting the SYS_PLL_PD bits to one in the Power-down configuration register ([Table 3-40](#)). In this mode, the internal current reference will be turned off, the oscillator and the phase-frequency detector will be stopped and the dividers will enter a reset state. While in Power-down mode, the lock output will be low to indicate that the PLL is not in lock. When the Power-down mode is terminated by setting the SYS_PLL_PD bits to zero, the PLL will resume its normal operation and will make the lock signal high once it has regained lock on the input clock.

9.4 Operating modes

Table 43. PLL operating modes

Mode	Description	PD bit	BYPASS bit	DIRECT bit
1	Normal mode	0	0	0
2	Direct CCO mode	0	0	0
3	Power-down mode	1	0	x
4	Bypass mode ^[1]	x	1	0
5	Direct bypass mode ^[1]	x	1	1

[1] Analog part of the PLL is powered down automatically.

9.5 Divider ratio programming

Post divider

The division ratio of the post divider is controlled by the PSEL bits. The division ratio is two times the value of P selected by PSEL bits as shown in [Table 3-8](#). This guarantees an output clock with a 50% duty cycle.

Feedback divider

The feedback divider's division ratio is controlled by the MSEL bits. The division ratio between the PLL's output clock and the input clock is the decimal value on MSEL bits plus one, as specified in [Table 3-8](#).

Changing the divider values

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

9.6 Frequency selection

The PLL frequency equations use the following parameters (also see [Figure 3-3](#)):

Table 44. PLL frequency parameters

Parameter	System PLL
FCLKIN	Frequency of sys_pllclk (input clock to the system PLL) from the SYSPLLCLKSEL multiplexer (see Section 3-4.9).
FCCO	Frequency of the Current Controlled Oscillator (CCO); 156 to 320 MHz.
FCLKOUT	Frequency of sys_pllclkout
P	System PLL post divider ratio; PSEL bits in SYSPLLCTRL (see Section 3-4.3).
M	System PLL feedback divider register; MSEL bits in SYSPLLCTRL (see Section 3-4.3).

9.6.1 Mode 1 (Normal mode)

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

(1)

$$F_{clkout} = M \times F_{clk} = (FCCO)/(2 \times P)$$

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

1. Specify the input clock frequency F_{clk} .
2. Calculate M to obtain the desired output frequency F_{clkout} with $M = F_{clkout} / F_{clk}$.
3. Find a value so that $FCCO = 2 \times P \times F_{clkout}$.
4. Verify that all frequencies and divider values conform to the limits specified in [Table 3-8](#).

9.6.2 Mode 2 (Direct CCO mode)

In this mode the post divider is bypassed and the CCO clock is sent directly to the output(s), leading to the following

frequency equation:

(2)

$$F_{clkout} = M \times F_{clk} = FCCO$$

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

1. Specify the input clock frequency F_{clk} .

2. Calculate M to obtain the desired output frequency F_{clkout} with $M = F_{clkout} / F_{clkin}$.
3. Verify that all frequencies and divider values conform to the limits specified in [Table 3–8](#).

Note that although the post divider is not used, it is still running in this mode. To reduce the current consumption to the lowest possible value, it is recommended to set PSEL bits to '00'. This will set the post divider to divide by two, which causes it to consume the least amount of current.

9.6.3 Mode 3 (Power-down mode)

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

9.6.4 Mode 4 (Bypass mode)

(3)

$$F_{clkout} = F_{clkin} / (2 \times P)$$

For $M > 1$:

(4)

$$F_{divo} = F_{clkout} / M$$

Due to the particular construction of the feedback divider the divo output is not the feedback clock but only a signal that masks the feedback divider's input clock to generate the actual feedback clock. As a consequence the divo output signal is '1' when M is set to 1.

9.6.5 Mode 5 (Direct bypass mode)

In this mode, the analog part is placed in Power-down, the post divider is disabled and the input clock is sent directly to the output(s). This mode can e.g. be used to perform either a function test and/or a scan test on the feedback divider. In this mode, the frequency of the feedback clock output is given by:

For $M > 1$:

(5)

$$F_{divo} = F_{clkin}/M$$

1. Introduction

The PMU controls the Deep power-down mode. Four general purpose register in the PMU can be used to retain data during Deep power-down mode.

2. Register description

Table 45. Register overview: PMU (base address 0x4003 8000)

Name	Access	Address offset	Description	Reset value
PCON	R/W	0x000	Power control register	0x0
GPREG0	R/W	0x004	General purpose register 0	0x0
GPREG1	R/W	0x008	General purpose register 1	0x0
GPREG2	R/W	0x00C	General purpose register 2	0x0
GPREG3	R/W	0x010	General purpose register 3	0x0
GPREG4	R/W	0x014	General purpose register 4	0x0

2.1 Power control register

The power control register selects whether Sleep mode or Deep-sleep mode is entered when using the ARM WFI instruction.

Table 46. Power control register (PCON, address 0x4003 8000) bit description

Bit	Symbol	Value	Description	Reset value
0	-	-	Reserved. Do not write 1 to this bit.	0x0
1	DPDEN	-	Deep power-down mode enable	0x0
		1	ARM WFI will enter Deep-power down mode (ARM Cortex-M0 core powered-down).	0x0
		0	ARM WFI will enter Sleep mode (clock to ARM Cortex-M0 core turned off).	0x0
10:2	-	-	Reserved. Do not write ones to this bit.	0x0
11	DPDFLAG	-	Deep power-down flag	0x0
		1	Read: Deep power-down mode entered. Write: Clear the Deep power-down flag.	0x0
		0	Read: Deep power-down mode not entered. Write: No effect.	0x0
31:12	-	-	Reserved. Do not write ones to this bit.	0x0

2.2 General purpose registers 0 to 3

The general purpose registers retain data through the Deep power-down mode when power is still applied to the $V_{DD(3V3)}$ pin but the chip has entered Deep power-down mode. Only a “cold” boot when all power has been completely removed from the chip will reset the general purpose registers.

Table 47. General purpose registers 0 to 3 (GPREG0 - GPREG3, address 0x4003 8004 to 0x4003 8010) bit description

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

2.3 General purpose register 4

The general purpose register 4 retains data through the Deep power-down mode when power is still applied to the $V_{DD(3V3)}$ pin but the chip has entered Deep power-down mode. Only a “cold” boot, when all power has been completely removed from the chip, will reset the general purpose registers.

Remark: If the external voltage applied on pin $V_{DD(3V3)}$ drops below $\langle tbd \rangle V$, the hysteresis of the WAKEUP input pin has to be disabled in order for the chip to wake up from Deep power-down mode.

Table 48. General purpose register 4 (GPREG4, address 0x4003 8014) bit description

Bit	Symbol	Value	Description	Reset value
9:0	-	-	Reserved. Do not write ones to this bit.	0x0
10	WAKEUPHYS		WAKEUP pin hysteresis enable	0x0
		1	Hysteresis for WAKEUP pin enabled.	
		0	Hysteresis for WAKUP pin disabled.	
31:11	GPDATA		Data retained during Deep power-down mode.	0x0

3. Functional description

3.1 Entering Deep power-down mode

Follow these steps to enter Deep power-down mode from normal Run mode:

1. (optional) Save data to be retained during Deep power-down to the DATA bits in the four general purpose registers ([Table 4–47](#) and [Table 4–48](#)).
2. Set the DPDEN bit to one on the PCON register ([Table 4–46](#)) to enable Deep power-down mode.
3. Issue ARM Cortex-M0 WFI/WFE instruction.

After step 3, the PMU turns off the on-chip voltage regulator and waits for a wake-up signal on the WAKEUP pin.

3.2 Exiting Deep power-down mode

Follow these steps to wake up the chip from Deep power-down mode:

1. On the WAKEUP pin, transition from HIGH to LOW.
 - The PMU will turn on the on-chip voltage regulator. When the core voltage reaches the power-on-reset (POR) trip point, a system reset will be triggered and the chip re-boots.
 - All registers except the GPREG0 to GPREG4 and PCON will be in their reset state.

2. Once the chip has booted, read the deep power-down flag in the PCON register ([Table 4-45](#)) to verify that the reset was caused by a wake-up event from Deep power-down and was not a cold reset.
3. Clear the deep power-down flag in the PCON register ([Table 4-45](#)).
4. (Optional) Read the stored data in the general purpose registers ([Table 4-47](#) and [Table 4-48](#)).
5. Set up the PMU for the next Deep power-down cycle (see [Section 4-3.1](#)).

1. Introduction

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

2. Features

- Nested Vectored Interrupt Controller that is an integral part of the ARM Cortex-M0
- Tightly coupled interrupt controller provides low interrupt latency
- Controls system exceptions and peripheral interrupts
- In the LPC111x, the NVIC supports 32 vectored interrupts
- 8 programmable interrupt priority levels with hardware priority level masking
- Relocatable vector table
- Non-Maskable Interrupt (NMI)
- Software interrupt generation

3. Interrupt sources

[Table 5–49](#) 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. There is no significance or priority about what line is connected where, except for certain standards from ARM.

Table 49. Connection of interrupt sources to the Vectored Interrupt Controller

Exception Number	Vector Offset	Function	Flag(s)
12 to 0		start logic wake-up interrupts;	Each interrupt is connected to a PIO input pin serving as wake-up pin from Deep-sleep mode; PIO0_0 to PIO0_11 and PIO1_0; see Table 3–34 .
13	-		Reserved
14	SSP1		Tx FIFO half empty Rx FIFO half full Rx Timeout Rx Overrun
15	I ² C		SI (state change)
16	CT16B0		Match 0 - 2 Capture 0
17	CT16B1		Match 0 - 1 Capture 0
18	CT32B0		Match 0 - 3 Capture 0

Table 49. Connection of interrupt sources to the Vectored Interrupt Controller

Exception Number	Vector Offset	Function	Flag(s)
19		CT32B1	Match 0 - 3 Capture 0
20		SSP0	Tx FIFO half empty Rx FIFO half full Rx Timeout Rx Overrun
21		UART	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)
22		-	Reserved
23		-	Reserved
24		ADC	A/D Converter end of conversion
25		WDT	Watchdog interrupt (WDINT)
26		BOD	Brown-out detect
27		-	Reserved
28		PIO_3	GPIO interrupt status of port 3
29		PIO_2	GPIO interrupt status of port 2
30		PIO_1	GPIO interrupt status of port 1
31		PIO_0	GPIO interrupt status of port 0

1. How to read this chapter

The number of GPIO pins available on each port depends on the LPC111x part and the package. See [Table 6–50](#) for available GPIO pins:

Table 50. GPIO configuration

Part	Package	GPIO port 0	GPIO port 1	GPIO port 2	GPIO port 3	Total GPIO pins
LPC1111	HVQFN33	PIO0_0 to PIO0_11	PIO1_0 to PIO1_11	PIO2_0	PIO3_2; PIO3_4; PIO3_5	28
LPC1112	HVQFN33	PIO0_0 to PIO0_11	PIO1_0 to PIO1_11	PIO2_0	PIO3_2; PIO3_4; PIO3_5	28
LPC1113	HVQFN33	PIO0_0 to PIO0_11	PIO1_0 to PIO1_11	PIO2_0	PIO3_2; PIO3_4; PIO3_5	28
	PLCC44	PIO0_0 to PIO0_11	PIO1_0 to PIO1_11	PIO2_0 to PIO2_11	PIO3_4 and PIO3_5	38
	LQFP48	PIO0_0 to PIO0_11	PIO1_0 to PIO1_11	PIO2_0 to PIO2_11	PIO3_0 to PIO3_5	42

Register bits corresponding to PION_m pins which are not available are reserved.

2. Introduction

2.1 Features

- Digital ports can be configured input/output by software.
- Each individual port pin can serve as external interrupt input pin.
- Interrupts can be configured on single falling or rising edges and on both edges.
- Individual interrupt levels can be programmed.
- All GPIO pins are inputs by default.
- Read and write data operations from/to the port pins are maskable.

3. Register description

Each GPIO register is 32 bit wide and can be accessed in byte, half-word, and word sizes. Individual bits (i.e. GPIO ports) can also be set by writing directly to the port pin's address.

Table 51. Register overview: GPIO (base address port 0: 0x5000 0000; port 1: 0x5001 0000, port 2: 0x5002 0000; port 3: 0x5003 0000)

Name	Access	Address offset	Description	Reset value
GPIO n DATA	R/W	0x0000 to 0x3FFC	Port n data register for pins PION_0 to PION_11; 4096 locations; each data register is 32 bit wide;	0x00
-	-	0x4000 to 0x7FFC	reserved	-
GPIO n DIR	R/W	0x8000	Data direction register for port n	0x00
GPIO n IS	R/W	0x8004	Interrupt sense register for port n	0x00

Table 51. Register overview: GPIO (base address port 0: 0x5000 0000; port 1: 0x5001 0000, port 2: 0x5002 0000; port 3: 0x5003 0000)

Name	Access	Address offset	Description	Reset value
GPIO_nIBE	R/W	0x8008	Interrupt both edges register for port n	0x00
GPIO_nIEV	R/W	0x800C	Interrupt event register for port n	0x00
GPIO_nIE	R/W	0x8010	Interrupt mask register for port n	0x00
GPIO_nRIS	R	0x8014	Raw interrupt status register for port n	0x00
GPIO_nMIS	R	0x8018	Masked interrupt status register for port n	0x00
GPIO_nIC	W	0x801C	Interrupt clear register for port n	0x00
-	-	0x8020 - 0x8FFF	reserved	0x00

3.1 GPIO data register

The data register allows to read the values on the pins programmed as inputs and to program the values on pins configured as outputs. The same data register appears at 4096 locations in the GPIO address space and 12 bits of the address bus can be used for bit masking (see [Section 6–4.1](#)).

Table 52. GPIO_nDATA register (GPIO0DATA, address 0x5000 0000 to 0x5000 3FFC; GPIO1DATA, address 0x5001 0000 to 0x5001 3FFC; GPIO2DATA, address 0x5002 0000 to 0x5002 3FFC; GPIO3DATA, address 0x5003 0000 to 0x5003 3FFC) bit description

Bit	Symbol	Access	Description	Reset value
11:0	DATA	R/W	Input data (read) or output data (write) for pins PION_0 to PION_11.	0x00
31:12	-	-	Reserved	0x00

3.2 GPIO data direction register

Table 53. GPIO_nDIR register (GPIO0DIR, address 0x5000 8000 to GPIO3DIR, address 0x5003 8000) bit description

Bit	Symbol	Access	Value	Description	Reset value
11:0	IO	R/W		Selects pin x as input or output (x = 0 to 11).	0x00
			0	Pin PION_x is configured as input.	
			1	Pin PION_x is configured as output.	
31:12	-	-	-	Reserved	-

3.3 GPIO interrupt sense register

Table 54. GPIOInIS register (GPIO0IS, address 0x5000 8004 to GPIO3IS, address 0x5003 8004) bit description

Bit	Symbol	Access	Value	Description	Reset value
11:0	ISENSE	R/W		Selects interrupt on pin x as level or edge sensitive (x = 0 to 11).	0x00
			0	Interrupt on pin PION_x is configured as edge sensitive.	
			1	Interrupt on pin PION_x is configured as level sensitive.	
31:12	-	-	-	Reserved	-

3.4 GPIO interrupt both edges sense register

Table 55. GPIOInIBE register (GPIO0IBE, address 0x5000 8008 to GPIO3IBE, address 0x5003 8008) bit description

Bit	Symbol	Access	Value	Description	Reset value
11:0	IBE	R/W		Selects interrupt on pin x to be triggered on both edges (x = 0 to 11).	0x00
			0	Interrupt on pin PION_x is controlled through register GPIOInIEV.	
			1	Both edges on pin PION_x trigger an interrupt.	
31:12	-	-	-	Reserved	-

3.5 GPIO interrupt event register

Table 56. GPIOInIEV register (GPIO0IEV, address 0x5000 800C to GPIO3IEV, address 0x5003 800C) bit description

Bit	Symbol	Access	Value	Description	Reset value
11:0	IEV	R/W		Selects interrupt on pin x to be triggered rising or falling edges (x = 0 to 11).	0x00
			0	Depending on setting in register GPIOInIS (see Table 6-54), Rising edges or HIGH level on pin PION_x trigger an interrupt.	
			1	Depending on setting in register GPIOInIS (see Table 6-54), falling edges or LOW level on pin PION_x trigger an interrupt.	
31:12	-	-	-	Reserved	-

3.6 GPIO interrupt mask register

Bits set to HIGH in the GPIOInIE register allow the corresponding pins to trigger their individual interrupts and the combined GPIOInINTR line. Clearing a bit disables interrupt triggering on that pin.

Table 57. GPIOIE register (GPIO0IE, address 0x5000 8010 to GPIO3IE, address 0x5003 8010) bit description

Bit	Symbol	Access	Value	Description	Reset value
11:0	MASK	R/W		Selects interrupt on pin x to be masked (x = 0 to 11).	0x00
			0	Interrupt on pin PION_x is masked.	
			1	Interrupt on pin PION_x is not masked.	
31:12	-	-	-	Reserved	-

3.7 GPIO raw interrupt status register

Bits read HIGH in the GPIOIRS register reflect the raw (prior to masking) interrupt status of the corresponding pins indicating that all the requirements have been met before they are allowed to trigger the GPIOIE. Bits read as zero indicate that the corresponding input pins have not initiated an interrupt. The register is read-only.

Table 58. GPIOIRS register (GPIO0IRS, address 0x5000 8014 to GPIO3IRS, address 0x5003 8014) bit description

Bit	Symbol	Access	Value	Description	Reset value
11:0	MASK	R		Selects interrupt on pin x to be masked (x = 0 to 11).	0x00
			0	No interrupt on pin PION_x.	
			1	Interrupt requirements met on PION_x.	
31:12	-	-	-	Reserved	-

3.8 GPIO masked interrupt status register

Bits read HIGH in the GPIOMIS register reflect the status of the input lines triggering an interrupt. Bits read as LOW indicate that either no interrupt on the corresponding input pins has been generated or that the interrupt is masked. GPIOMIS is the state of the interrupt after masking. The register is read-only.

Table 59. GPIOMIS register (GPIO0MIS, address 0x5000 8018 to GPIO3MIS, address 0x5003 8018) bit description

Bit	Access	Symbol	Value	Description	Reset value
11:0	R	MASK		Selects interrupt on pin x to be masked (x = 0 to 11).	0x00
			0	No interrupt or interrupt masked on pin PION_x.	
			1	Interrupt on PION_x.	
31:12	-	-	-	Reserved	-

3.9 GPIO interrupt clear register

Table 60. GPIO_nIC register (GPIO0IC, address 0x5000 801C to GPIO3IC, address 0x5003 801C) bit description

Bit	Symbol	Access	Value	Description	Reset value
11:0	CLR	W		Selects interrupt on pin x to be cleared (x = 0 to 11). Clears the interrupt edge detection logic. This register is write-only. Remark: The synchronizer between the GPIO and the NVIC blocks causes a delay of 2 clocks. It is recommended to add two NOPs after the clear of the interrupt edge detection logic before the exit of the interrupt service routine.	0x00
			0	No effect.	
			1	Clears edge detection logic for pin PION _x .	
31:12	-	-	-	Reserved	-

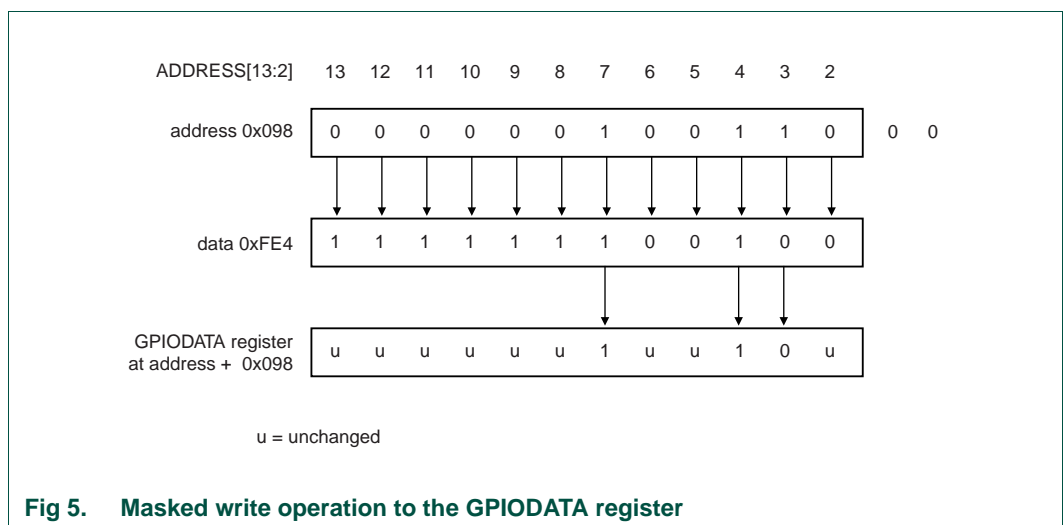
4. Functional description

4.1 Write/read data operation

In order for software to be able to set GPIO bits without affecting any other pins in a single write operation, bits [13:2] of a 14-bit wide address bus are used to create a 12-bit wide mask for write and read operations on the 12 GPIO pins for each port. The masked GPIODATA register can be located anywhere between address offsets 0x0000 to 0x3FFC in the GPIO_n address space depending on the address chosen.

Write operation

If the address bit associated with the GPIO data bit to be written is HIGH, the value of the GPIODATA register bit is updated from the GPIO data bit. If the address bit is LOW, the corresponding GPIODATA register bit is left unchanged.



Read operation

If the address bit associated with the GPIO data bit is HIGH, the value is read. If the address bit is LOW, the GPIO data bit is read as 0.

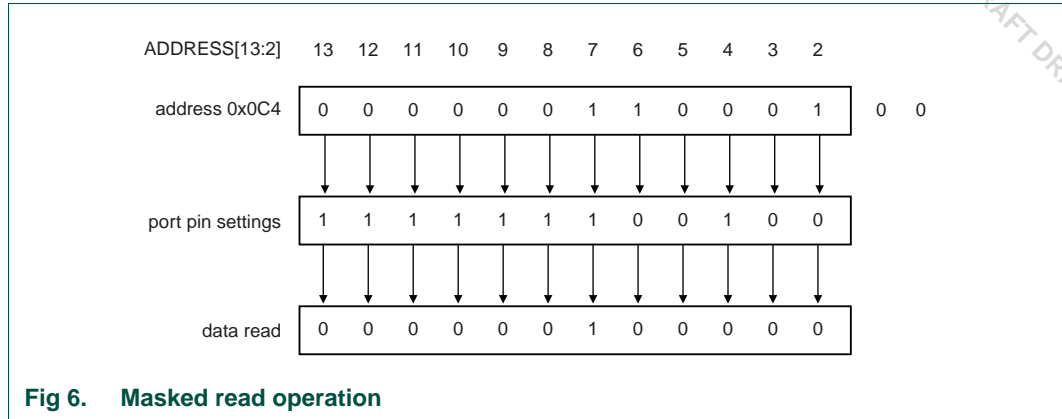


Fig 6. Masked read operation

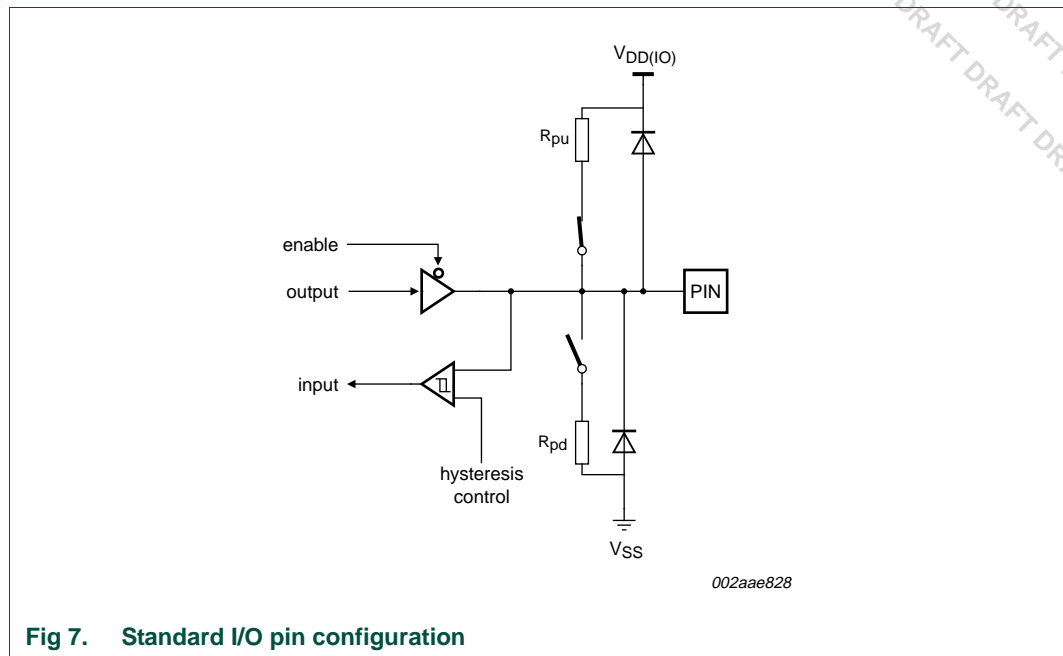


Fig 7. Standard I/O pin configuration

3.1 Pin function

The FUNC bits in the IOCON registers can be set to GPIO (FUNC = 000) or to a peripheral function. If the pins are GPIO pins, the GPIO nDIR registers determine whether the pin is configured as an input or output (see [Table 6-53](#)). For any peripheral function, the pin direction is controlled automatically depending on the pin's functionality. The GPIO nDIR registers have no effect for peripheral functions.

3.2 Pin mode

The MODE bits in the IOCON register allow the selection of on-chip pull-up or pull-down resistors for each pin or select the repeater mode.

The possible on-chip resistor configurations are pull-up enabled, pull-down enabled, or no pull-up/pull-down. The default value is pull-up enabled.

The 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.

3.3 Hysteresis

The input buffer for digital functions can be configured with hysteresis or as plain buffer through the IOCON registers (see the *LPC111x data sheet* for details).

If the external pad supply voltage $V_{DD(IO)}$ is between 2.5 V and 3.6 V, the hysteresis buffer can be enabled or disabled. If $V_{DD(IO)}$ is below 2.5 V, the hysteresis buffer must be **disabled** to use the pin in input mode.

3.4 A/D-mode

In A/D-mode, the digital receiver is disconnected to obtain an accurate input voltage for analog-to-digital conversions. This mode can be selected in those IOCON registers that control pins with an analog function. If A/D mode is selected, Hysteresis and Pin mode settings have no effect.

For pins without analog functions, the A/D-mode setting has no effect.

3.5 I²C mode

If the I²C function is selected by the FUNC bits of registers IOCON_PIO0_4 (Table 7-73) and IOCON_PIO0_5 (Table 7-74), then the I²C-bus pins can be configured for different I²C-modes:

- Standard mode/Fast-mode I²C with input glitch filter (this includes an open-drain output according to the I²C-bus specification).
- Fast-mode Plus with input glitch filter (this includes an open-drain output according to the I²C-bus specification). In this mode, the pins function as high-current sinks.
- Standard I/O functionality without input filter.

Remark: Either Standard mode/Fast-mode I²C or Standard I/O functionality should be selected if the pin is used as GPIO pin.

4. Register description

The I/O configuration registers control the PIO port pins, the inputs and outputs of all peripherals and functional blocks, the I²C-bus pins, and the ADC input pins.

Each port pin PION_m has one IOCON register assigned to control the pin's function and electrical characteristics.

Some input functions (SCK0, DSR0, DCD0, and RI0) are multiplexed to several physical pins. The IOCON_LOC registers select the pin location for each of these functions.

Remark: The IOCON registers are listed in order of their memory locations in Table 7-61, which correspond to the order of their physical pin numbers in the LQFP48 package starting at the upper left corner with pin 1 (PIO2_6). See Table 7-62 for a listing of IOCON registers ordered by port number.

Table 61. Register overview: I/O configuration (base address 0x4004 4000)

Name	Access	Address offset	Description	Reset value	Reference
IOCON_PIO2_6	R/W	0x000	I/O configuration for pin PIO2_6	0xD0	Table 7-63
-	R/W	0x004	Reserved	-	-
IOCON_PIO2_0	R/W	0x008	I/O configuration for pin PIO2_0/DTR/SSEL1	0xD0	Table 7-64
IOCON_RESET_PIO0_0	R/W	0x00C	I/O configuration for pin RESET/PIO0_0	0xD0	Table 7-65
IOCON_PIO0_1	R/W	0x010	I/O configuration for pin PIO0_1/CLKOUT/CT32B0_MAT2	0xD0	Table 7-66
IOCON_PIO1_8	R/W	0x014	I/O configuration for pin PIO1_8/CT16B1_CAP0	0xD0	Table 7-67

Table 61. Register overview: I/O configuration (base address 0x4004 4000)

Name	Access	Address offset	Description	Reset value	Reference
-	R/W	0x018	Reserved	-	-
IOCON_PIO0_2	R/W	0x01C	I/O configuration for pin PIO0_2/SSEL0/CT16B0_CAP0	0xD0	Table 7-68
IOCON_PIO2_7	R/W	0x020	I/O configuration for pin PIO2_7	0xD0	Table 7-69
IOCON_PIO2_8	R/W	0x024	I/O configuration for pin PIO2_8	0xD0	Table 7-70
IOCON_PIO2_1	R/W	0x028	I/O configuration for pin PIO2_1/DSR/SCK1	0xD0	Table 7-71
IOCON_PIO0_3	R/W	0x02C	I/O configuration for pin PIO0_3	0xD0	Table 7-72
IOCON_PIO0_4	R/W	0x030	I/O configuration for pin PIO0_4/SCL	0xC0	Table 7-73
IOCON_PIO0_5	R/W	0x034	I/O configuration for pin PIO0_5/SDA	0xC0	Table 7-74
IOCON_PIO1_9	R/W	0x038	I/O configuration for pin PIO1_9/CT16B1_MAT0	0xD0	Table 7-75
IOCON_PIO3_4	R/W	0x03C	I/O configuration for pin PIO3_4	0xD0	Table 7-76
IOCON_PIO2_4	R/W	0x040	I/O configuration for pin PIO2_4	0xD0	Table 7-77
IOCON_PIO2_5	R/W	0x044	I/O configuration for pin PIO2_5	0xD0	Table 7-78
IOCON_PIO3_5	R/W	0x048	I/O configuration for pin PIO3_5	0xD0	Table 7-79
IOCON_PIO0_6	R/W	0x04C	I/O configuration for pin PIO0_6/SCK0	0xD0	Table 7-80
IOCON_PIO0_7	R/W	0x050	I/O configuration for pin PIO0_7/CTS	0xD0	Table 7-81
IOCON_PIO2_9	R/W	0x054	I/O configuration for pin PIO2_9	0xD0	Table 7-82
IOCON_PIO2_10	R/W	0x058	I/O configuration for pin PIO2_10	0xD0	Table 7-83
IOCON_PIO2_2	R/W	0x05C	I/O configuration for pin PIO2_2/DCD/MISO1	0xD0	Table 7-84
IOCON_PIO0_8	R/W	0x060	I/O configuration for pin PIO0_8/MISO0/CT16B0_MAT0	0xD0	Table 7-85
IOCON_PIO0_9	R/W	0x064	I/O configuration for pin PIO0_9/MOSI0/CT16B0_MAT1	0xD0	Table 7-86
IOCON_JTAG_TCK_PIO0_10	R/W	0x068	I/O configuration for pin SWCLK/PIO0_10/SCK0/CT16B0_MAT2	0xD0	Table 7-87
IOCON_PIO1_10	R/W	0x06C	I/O configuration for pin PIO1_10/AD6/CT16B1_MAT1	0xD0	Table 7-88
IOCON_PIO2_11	R/W	0x070	I/O configuration for pin PIO2_11/SCK0	0xD0	Table 7-89
IOCON_JTAG_TDI_PIO0_11	R/W	0x074	I/O configuration for pin TDI/PIO0_11/AD0/CT32B0_MAT3	0xD0	Table 7-90
IOCON_JTAG_TMS_PIO1_0	R/W	0x078	I/O configuration for pin TMS/PIO1_0/AD1/CT32B1_CAP0	0xD0	Table 7-91
IOCON_JTAG_TDO_PIO1_1	R/W	0x07C	I/O configuration for pin TDO/PIO1_1/AD2/CT32B1_MAT0	0xD0	Table 7-92
IOCON_JTAG_nTRST_PIO1_2	R/W	0x080	I/O configuration for pin TRST/PIO1_2/AD3/CT32B1_MAT1	0xD0	Table 7-93
IOCON_PIO3_0	R/W	0x084	I/O configuration for pin PIO3_0/DTR	0xD0	Table 7-94
IOCON_PIO3_1	R/W	0x088	I/O configuration for pin PIO3_1/DSR	0xD0	Table 7-95
IOCON_PIO2_3	R/W	0x08C	I/O configuration for pin PIO2_3/RI/MOSI1	0xD0	Table 7-96

Table 61. Register overview: I/O configuration (base address 0x4004 4000)

Name	Access	Address offset	Description	Reset value	Reference
IOCON_SWDIO_PIO1_3	R/W	0x090	I/O configuration for pin SWDIO/PIO1_3/AD4/CT32B1_MAT2	0xD0	Table 7-97
IOCON_PIO1_4	R/W	0x094	I/O configuration for pin PIO1_4/AD5/CT32B1_MAT3	0xD0	Table 7-98
IOCON_PIO1_11	R/W	0x098	I/O configuration for pin PIO1_11/AD7	0xD0	Table 7-99
IOCON_PIO3_2	R/W	0x09C	I/O configuration for pin PIO3_2/ $\overline{\text{DCD}}$	0xD0	Table 7-100
IOCON_PIO1_5	R/W	0x0A0	I/O configuration for pin PIO1_5/RTS/CT32B0_CAP0	0xD0	Table 7-101
IOCON_PIO1_6	R/W	0x0A4	I/O configuration for pin PIO1_6/RXD/CT32B0_MAT0	0xD0	Table 7-102
IOCON_PIO1_7	R/W	0x0A8	I/O configuration for pin PIO1_7/TXD/CT32B0_MAT1	0xD0	Table 7-103
IOCON_PIO3_3	R/W	0x0AC	I/O configuration for pin PIO3_3/ $\overline{\text{RI}}$	0xD0	Table 7-104
IOCON_SCK_LOC	R/W	0x0B0	SCK pin location select register	0x00	Table 7-105
IOCON_DSR_LOC	R/W	0x0B4	DSR pin location select register	0x00	Table 7-106
IOCON_DCD_LOC	R/W	0x0B8	$\overline{\text{DCD}}$ pin location select register	0x00	Table 7-107
IOCON_RI_LOC	R/W	0x0BC	$\overline{\text{RI}}$ pin location register	0x00	Table 7-108

Table 62. I/O configuration registers ordered by port number

Port pin	Register name	LPC1111 HVQFN33	LPC1112 HVQFN33	LPC1113 HVQFN33	LPC1113 PLCC44	LPC1113 LQFP48	Reference
PIO0_0	IOCON_RESET_PIO0_0	yes	yes	yes	yes	yes	Table 7-65
PIO0_1	IOCON_PIO0_1	yes	yes	yes	yes	yes	Table 7-66
PIO0_2	IOCON_PIO0_2	yes	yes	yes	yes	yes	Table 7-68
PIO0_3	IOCON_PIO0_3	yes	yes	yes	yes	yes	Table 7-72
PIO0_4	IOCON_PIO0_4	yes	yes	yes	yes	yes	Table 7-73
PIO0_5	IOCON_PIO0_5	yes	yes	yes	yes	yes	Table 7-74
PIO0_6	IOCON_PIO0_6	yes	yes	yes	yes	yes	Table 7-80
PIO0_7	IOCON_PIO0_7	yes	yes	yes	yes	yes	Table 7-81
PIO0_8	IOCON_PIO0_8	yes	yes	yes	yes	yes	Table 7-85
PIO0_9	IOCON_PIO0_9	yes	yes	yes	yes	yes	Table 7-86
PIO0_10	IOCON_JTAG_TCK_PIO0_10	yes	yes	yes	yes	yes	Table 7-87
PIO0_11	IOCON_JTAG_TDI_PIO0_11	yes	yes	yes	yes	yes	Table 7-90
PIO1_0	IOCON_JTAG_TMS_PIO1_0	yes	yes	yes	yes	yes	Table 7-91
PIO1_1	IOCON_JTAG_TDO_PIO1_1	yes	yes	yes	yes	yes	Table 7-92
PIO1_2	IOCON_JTAG_nTRST_PIO1_2	yes	yes	yes	yes	yes	Table 7-93
PIO1_3	IOCON_SWDIO_PIO1_3	yes	yes	yes	yes	yes	Table 7-97
PIO1_4	IOCON_PIO1_4	yes	yes	yes	yes	yes	Table 7-98
PIO1_5	IOCON_PIO1_5	yes	yes	yes	yes	yes	Table 7-101
PIO1_6	IOCON_PIO1_6	yes	yes	yes	yes	yes	Table 7-102
PIO1_7	IOCON_PIO1_7	yes	yes	yes	yes	yes	Table 7-103
PIO1_8	IOCON_PIO1_8	yes	yes	yes	yes	yes	Table 7-67

Table 62. I/O configuration registers ordered by port number

Port pin	Register name	LPC1111 HVQFN33	LPC1112 HVQFN33	LPC1113 HVQFN33	LPC1113 PLCC44	LPC1113 LQFP48	Reference
PIO1_9	IOCON_PIO1_9	yes	yes	yes	yes	yes	Table 7-75
PIO1_10	IOCON_PIO1_10	yes	yes	yes	yes	yes	Table 7-88
PIO1_11	IOCON_PIO1_11	yes	yes	yes	yes	yes	Table 7-99
PIO2_0	IOCON_PIO2_0	yes	yes	yes	yes	yes	Table 7-64
PIO2_1	IOCON_PIO2_1	no	no	no	yes	yes	Table 7-71
PIO2_2	IOCON_PIO2_2	no	no	no	yes	yes	Table 7-84
PIO2_3	IOCON_PIO2_3	no	no	no	yes	yes	Table 7-96
PIO2_4	IOCON_PIO2_4	no	no	no	yes	yes	Table 7-77
PIO2_5	IOCON_PIO2_5	no	no	no	yes	yes	Table 7-78
PIO2_6	IOCON_PIO2_6	no	no	no	yes	yes	Table 7-63
PIO2_7	IOCON_PIO2_7	no	no	no	yes	yes	Table 7-69
PIO2_8	IOCON_PIO2_8	no	no	no	yes	yes	Table 7-70
PIO2_9	IOCON_PIO2_9	no	no	no	yes	yes	Table 7-82
PIO2_10	IOCON_PIO2_10	no	no	no	yes	yes	Table 7-83
PIO2_11	IOCON_PIO2_11	no	no	no	yes	yes	Table 7-89
PIO3_0	IOCON_PIO3_0	no	no	no	no	yes	Table 7-94
PIO3_1	IOCON_PIO3_1	no	no	no	no	yes	Table 7-95
PIO3_2	IOCON_PIO3_2	yes	yes	yes	no	yes	Table 7-100
PIO3_3	IOCON_PIO3_3	no	no	no	no	yes	Table 7-104
PIO3_4	IOCON_PIO3_4	yes	yes	yes	yes	yes	Table 7-76
PIO3_5	IOCON_PIO3_5	yes	yes	yes	yes	yes	Table 7-79
-	IOCON_SCK_LOC	yes (SCKLOC = 01 reserved)	yes (SCKLOC = 01 reserved)	yes (SCKLOC = 01 reserved)	yes	yes	Table 7-105
-	IOCON_DSR_LOC	no	no	no	no	yes	Table 7-106
-	IOCON_DCD_LOC	no	no	no	no	yes	Table 7-107
-	IOCON_RI_LOC	no	no	no	no	yes	Table 7-108

4.1 I/O configuration registers IOCON_PIO_n

For details on the I/O configuration settings, see [Section 7-3](#).

Table 63. IOCON_PIO2_6 register (IOCON_PIO2_6, address 0x4004 4000) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO2_6.	
		001 to 111	Reserved.	

Table 63. IOCON_PIO2_6 register (IOCON_PIO2_6, address 0x4004 4000) bit description

Bit	Symbol	Value	Description	Reset value
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 64. IOCON_PIO2_0 register (IOCON_PIO2_0, address 0x4004 4008) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO2_0.	
		001	Select function \overline{DTR} .	
		010	Select function SSEL1.	
		011 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 65. IOCON_nRESET_PIO0_0 register (IOCON_nRESET_PIO0_0, address 0x4004 400C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function \overline{RESET} .	
		001	Selects function PIO0_0.	
		010 to 111	Reserved.	

Table 65. IOCON_nRESET_PIO0_0 register (IOCON_nRESET_PIO0_0, address 0x4004 400C) bit description

Bit	Symbol	Value	Description	Reset value
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 66. IOCON_PIO0_1 register (IOCON_PIO0_1, address 0x4004 4010) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO0_1.	
		001	Selects function CLKOUT.	
		010	Selects function CT32B0_MAT2.	
		011 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 67. IOCON_PIO1_8 register (IOCON_PIO1_8, address 0x4004 4014) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO1_8.	
		001	Selects function CT16B1_CAP0.	
		010 to 111	Reserved.	

Table 67. IOCON_PIO1_8 register (IOCON_PIO1_8, address 0x4004 4014) bit description

Bit	Symbol	Value	Description	Reset value
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 68. IOCON_PIO0_2 register (IOCON_PIO0_2, address 0x4004 401C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO0_2.	
		001	Selects function SSEL0.	
		010	Selects function CT16B0_CAP0.	
		011 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 69. IOCON_PIO2_7 register (IOCON_PIO2_7, address 0x4004 4020) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO2_7.	
		001 to 111	Reserved.	

Table 69. IOCON_PIO2_7 register (IOCON_PIO2_7, address 0x4004 4020) bit description

Bit	Symbol	Value	Description	Reset value
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 70. IOCON_PIO2_8 register (IOCON_PIO2_8, address 0x4004 4024) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO2_8.	
		001 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
	11	Repeater mode.		
5	HYS		Hysteresis.	0
		0	Disable.	
	1	Enable.		
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 71. IOCON_PIO2_1 register (IOCON_PIO2_1, address 0x4004 4028) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO2_1.	
		001	Select function $\overline{\text{DSR}}$.	
		010	Select function SCK1.	
		011 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 72. IOCON_PIO0_3 register (IOCON_PIO0_3 address 0x4004 402C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO0_3.	
		001 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 73. IOCON_PIO0_4 register (IOCON_PIO0_4 address 0x4004 4030) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO0_4.	
		001	Selects I2C function SCL.	
		010 to 111	Reserved.	
7:3		-	Reserved.	10000
9:8	I2CMODE		Selects I2C mode.	00
		00 ^[1]	Standard mode/ Fast-mode I2C.	
		01 ^[1]	Standard I/O functionality	
		10	Fast-mode Plus I2C	
		11	Reserved.	
31:10	-	-	Reserved.	-

[1] Select Standard mode (I2CMODE = 00, default) or Standard I/O functionality (I2CMODE = 01) if the pin function is GPIO (FUNC = 000).

Table 74. IOCON_PIO0_5 register (IOCON_PIO0_5 address 0x4004 4034) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO0_5.	
		001	Selects I2C function SDA.	
		010 to 111	Reserved.	
7:3		-	Reserved.	10000
9:8	I2CMODE		Selects I2C mode.	00
		00 ^[1]	Standard mode/ Fast-mode I2C.	
		01 ^[1]	Standard mode/ Fast-mode I2C.	
		10	Standard I/O functionality	
		11	Reserved.	
31:10	-	-	Reserved.	-

[1] Select Standard mode (I2CMODE = 00, default) or Standard I/O functionality (I2CMODE = 01) if the pin function is GPIO (FUNC = 000).

Table 75. IOCON_PIO1_9 register (IOCON_PIO1_9 address 0x4004 4038) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO1_9.	
		001	Selects function CT16B1_MAT0.	
		010 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 76. IOCON_PIO3_4 register (IOCON_PIO3_4, address 0x4004 403C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO3_4.	
		001 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 77. IOCON_PIO2_4 register (IOCON_PIO2_4, address 0x4004 4040) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO2_4.	
		001 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
	11	Repeater mode.		
5	HYS		Hysteresis.	0
		0	Disable.	
	1	Enable.		
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 78. IOCON_PIO2_5 register (IOCON_PIO2_5, address 0x4004 4044) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO2_5.	
		001 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
	11	Repeater mode.		
5	HYS		Hysteresis.	0
		0	Disable.	
	1	Enable.		
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 79. IOCON_PIO3_5 register (IOCON_PIO3_5, address 0x4004 4048) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO3_5.	
		001 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 80. IOCON_PIO0_6 register (IOCON_PIO0_6 address 0x4004 404C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO0_6.	
		001	Reserved.	
		010	Selects function SCK0 (only if pin PIO0_6/SCK0 selected in Table 7–105).	
		011 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 81. IOCON_PIO0_7 register (IOCON_PIO0_7, address 0x4004 4050) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO0_7.	
		001	Select function $\overline{\text{CTS}}$.	
		010 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 82. IOCON_PIO2_9 register (IOCON_PIO2_9, address 0x4004 4054) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO2_9.	
		001 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 83. IOCON_PIO2_10 register (IOCON_PIO2_10, address 0x4004 4058) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO2_10.	
		001 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 84. IOCON_PIO2_2 register (IOCON_PIO2_2, address 0x4004 405C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO2_2.	
		001	Select function \overline{DCD} .	
		010	Select function MISO1.	
		011 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 85. IOCON_PIO0_8 register (IOCON_PIO0_8, address 0x4004 4060) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO0_8.	
		001	Selects function MISO0.	
		010	Selects function CT16B0_MAT0.	
		011	Reserved.	
		100 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 86. IOCON_PIO0_9 register (IOCON_PIO0_9, address 0x4004 4064) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO0_9.	
		001	Selects function MOSI0.	
		010	Selects function CT16B0_MAT1.	
		011 to 111	Reserved.	
		4:3	MODE	
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 87. IOCON_JTAG_TCK_PIO0_10 register (IOCON_JTAG_TCK_PIO0_10, address 0x4004 4068) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function SWCLK.	
		001	Selects function PIO0_10.	
		010	Selects function SCK0 (only if pin SWCLK/PIO0_10/SCK0/CT16B0_MAT2 selected in Table 7-105).	
		011	Selects function CT16B0_MAT2.	
		100 to 111	Reserved.	
		4:3	MODE	
00	Inactive (no pull-down/pull-up resistor enabled).			
01	Pull-down resistor enabled.			
10	Pull-up resistor enabled.			
11	Repeater mode.			
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 88. IOCON_PIO1_10 register (IOCON_PIO1_10, address 0x4004 406C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO1_10.	
		001	Selects function AD6.	
		010	Selects function CT16B1_MAT1.	
		011 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
6	-	-	Reserved.	1
7	ADMODE		Select Analog/Digital mode.	1
		0	Analog input mode.	
		1	Digital functional mode.	
31:8	-	-	Reserved.	-

Table 89. IOCON_PIO2_11 register (IOCON_PIO2_11, address 0x4004 4070) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO2_11.	
		001	Select function SCK0 (only if pin PIO2_11/SCK0 selected in Table 7-105).	
		010 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 90. IOCON_JTAG_TDI_PIO0_11 register (IOCON_JTAG_TDI_PIO0_11, address 0x40044074) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function TDI.	
		001	Selects function PIO0_11.	
		010	Selects function AD0.	
		011	Selects function CT32B0_MAT3.	
		100 to 111	Reserved.	
		4:3	MODE	
00	Inactive (no pull-down/pull-up resistor enabled).			
01	Pull-down resistor enabled.			
10	Pull-up resistor enabled.			
11	Repeater mode.			
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
6	-	-	Reserved.	1
7	ADMODE		Select Analog/Digital mode.	1
		0	Analog input mode.	
		1	Digital functional mode.	
31:8	-	-	Reserved.	-

Table 91. IOCON_JTAG_TMS_PIO1_0 register (IOCON_JTAG_TMS_PIO1_0, address 0x40044078) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function TMS.	
		001	Selects function PIO1_0.	
		010	Selects function AD1.	
		011	Selects function CT32B1_CAP0.	
		100 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
6	-	-	Reserved.	1
7	ADMODE		Select Analog/Digital mode.	1
		0	Analog input mode.	
		1	Digital functional mode.	
31:8	-	-	Reserved.	-

Table 92. IOCON_JTAG_TDO_PIO1_1 register (IOCON_JTAG_TDO_PIO1_1, address 0x4004407C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function TDO.	
		001	Selects function PIO1_1.	
		010	Selects function AD2.	
		011	Selects function CT32B1_MAT0.	
		100 to 111	Reserved.	
		4:3	MODE	
00	Inactive (no pull-down/pull-up resistor enabled).			
01	Pull-down resistor enabled.			
10	Pull-up resistor enabled.			
11	Repeater mode.			
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
6	-	-	Reserved.	1
7	ADMODE		Select Analog/Digital mode.	1
		0	Analog input mode.	
		1	Digital functional mode.	
31:8	-	-	Reserved.	-

Table 93. IOCON_JTAG_nTRST_PIO1_2 register (IOCON_JTAG_nTRST_PIO1_2, address 0x4004 4080) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function $\overline{\text{TRST}}$.	
		001	Selects function PIO1_2.	
		010	Selects function AD3.	
		011	Selects function CT32B1_MAT1.	
		100 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
5	HYS		Hysteresis.	0
		0	Disable.	
6	-	-	Reserved.	1
7	ADMODE		Select Analog/Digital mode.	1
		0	Analog input mode.	
7	-	-	1	Digital functional mode.
31:8	-	-	Reserved.	-

Table 94. IOCON_PIO3_0 register (IOCON_PIO3_0, address 0x4004 4084) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO3_0.	
		001	Selects function $\overline{\text{DTR}}$.	
		001 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
4:3	-	-	11	Repeater mode.

Table 94. IOCON_PIO3_0 register (IOCON_PIO3_0, address 0x4004 4084) bit description

Bit	Symbol	Value	Description	Reset value
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 95. IOCON_PIO3_1 register (IOCON_PIO3_1, address 0x4004 4088) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO3_1/ $\overline{\text{DSR}}$.	
		001 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 96. IOCON_PIO2_3 register (IOCON_PIO2_3, address 0x4004 408C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO2_3/MOSI1.	
		001	Selects function $\overline{\text{RI}}$.	
		010 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	

Table 96. IOCON_PIO2_3 register (IOCON_PIO2_3, address 0x4004 408C) bit description

Bit	Symbol	Value	Description	Reset value
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 97. IOCON_SWDIO_PIO1_3 register (IOCON_SWDIO_PIO1_3, address 0x4004 4090) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function SWDIO.	
		001	Selects function PIO1_3.	
		010	Selects function AD4.	
		011	Selects function CT32B1_MAT2.	
		100 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
6	-	-	Reserved.	1
7	ADMODE		Select Analog/Digital mode.	1
		0	Analog input mode.	
		1	Digital functional mode.	
31:8	-	-	Reserved.	-

Table 98. IOCON_PIO1_4 register (IOCON_PIO1_4, address 0x4004 4094) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC ^[1]		Selects pin function.	000
		000	Selects function PIO1_4.	
		001	Selects function AD5.	
		010	Selects function CT32B1_MAT3.	
		100 to 011	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
6	-	-	Reserved.	1
7	ADMODE		Select Analog/Digital mode.	1
		0	Analog input mode.	
		1	Digital functional mode.	
31:8	-	-	Reserved.	-

[1] This pin functions as WAKEUP pin if the LPC111x is in Deep power-down mode regardless of the value of FUNC.

Table 99. IOCON_PIO1_11 register (IOCON_PIO1_11 address 0x4004 4098) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO1_11.	
		001	Selects function AD7.	
		010 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
6	-	-	Reserved.	1
7	ADMODE		Select Analog/Digital mode.	1
		0	Analog input mode.	
		1	Digital functional mode.	
31:8	-	-	Reserved.	-

Table 100. IOCON_PIO3_2 register (IOCON_PIO3_2, address 0x4004 409C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO3_2.	
		001	Selects function $\overline{\text{DCD}}$.	
		010 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 101. IOCON_PIO1_5 register (IOCON_PIO1_5, address 0x4004 40A0) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO1_5.	
		001	Selects function $\overline{\text{RTS}}$.	
		010	Selects function CT32B0_CAP0.	
		011 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 102. IOCON_PIO1_6 register (IOCON_PIO1_6, address 0x4004 40A4) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO1_6.	
		001	Selects function RXD.	
		010	Selects function CT32B0_MAT0.	
		011 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 103. IOCON_PIO1_7 register (IOCON_PIO1_7, address 0x4004 40A8) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO1_7.	
		001	Selects function TXD.	
		010	Selects function CT32B0_MAT1.	
		011 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

Table 104. IOCON_PIO3_3 register (IOCON_PIO3_3, address 0x4004 40AC) bit description

Bit	Symbol	Value	Description	Reset value
2:0	FUNC		Selects pin function.	000
		000	Selects function PIO3_3.	
		001	Selects function \overline{RI} .	
		010 to 111	Reserved.	
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	10
		00	Inactive (no pull-down/pull-up resistor enabled).	
		01	Pull-down resistor enabled.	
		10	Pull-up resistor enabled.	
		11	Repeater mode.	
5	HYS		Hysteresis.	0
		0	Disable.	
		1	Enable.	
7:6	-	-	Reserved.	11
31:8	-	-	Reserved.	0

4.2 IOCON location registers

The IOCON location registers are used to select a physical pin for multiplexed functions.

Remark: Note that once the pin location has been selected, the function still must be configured in the corresponding IOCON registers for the function to be usable on that pin.

Table 105. IOCON SCK location register (IOCON_SCK_LOC, address 0x4004 40B0) bit description

Bit	Symbol	Value	Description	Reset value
1:0	SCKLOC		Selects pin location for SCK0 pin.	00
		00	Selects SCK0 function in pin location SWCLK/PIO0_10/SCK0/CT16B0_MAT2 (see Table 7-87).	
		01	Selects SCK0 function in pin location PIO2_11/SCK0 (see Table 7-89).	
		10	Selects SCK0 function in pin location PIO0_6/SCK0 (see Table 7-80).	
		11	Reserved.	
31:2	-	-	Reserved.	-

Table 106. IOCON DSR location register (IOCON_DSR_LOC, address 0x4004 40B4) bit description

Bit	Symbol	Value	Description	Reset value
1:0	DSRLOC		Selects pin location for DSR0 pin.	00
		00	Selects $\overline{\text{DSR}}$ function in pin location PIO2_1/ $\overline{\text{DSR}}$ /SCK1.	
		01	Selects $\overline{\text{DSR}}$ function in pin location PIO3_1/ $\overline{\text{DSR}}$.	
		10	Reserved.	
		11	Reserved.	
31:2	-	-	Reserved.	-

Table 107. IOCON DCD location register (IOCON_DCD_LOC, address 0x4004 40B8) bit description

Bit	Symbol	Value	Description	Reset value
1:0	DCDLOC		Selects pin location for DCD pin.	00
		00	Selects $\overline{\text{DCD}}$ function in pin location PIO2_2/ $\overline{\text{DCD}}$ /MISO1.	
		01	Selects $\overline{\text{DCD}}$ function in pin location PIO3_2/ $\overline{\text{DCD}}$.	
		10	Reserved.	
		11	Reserved.	
31:2	-	-	Reserved.	-

Table 108. IOCON $\overline{\text{RI}}$ location register (IOCON_RI_LOC, address 0x4004 40BC) bit description

Bit	Symbol	Value	Description	Reset value
1:0	RILOC		Selects pin location for RI pin.	00
		00	Selects $\overline{\text{RI}}$ function in pin location PIO2_3/ $\overline{\text{RI}}$ /MOSI1.	
		01	Selects $\overline{\text{RI}}$ function in pin location PIO3_3/ $\overline{\text{RI}}$.	
		10	Reserved.	
		11	Reserved.	
31:2	-	-	Reserved.	-

1. How to read this chapter

The LPC111x are available in three packages: LQFP48 (LPC1113), PLCC44 (LPC1113), and HVQFN33 (LPC1111, LPC1112, LPC1113).

Table 109. LPC111x pin configurations

Part		LQFP48	PLCC44	HVQFN33
LPC1111	Pin configuration	-	-	Figure 8–10
	Pin description	-	-	Table 8–112
LPC1112	Pin configuration	-	-	Figure 8–10
	Pin description	-	-	Table 8–112
LPC1113	Pin configuration	Figure 8–8	Figure 8–9	Figure 8–10
	Pin description	Table 8–110	Table 8–111	Table 8–112

2. LPC111x Pin configuration

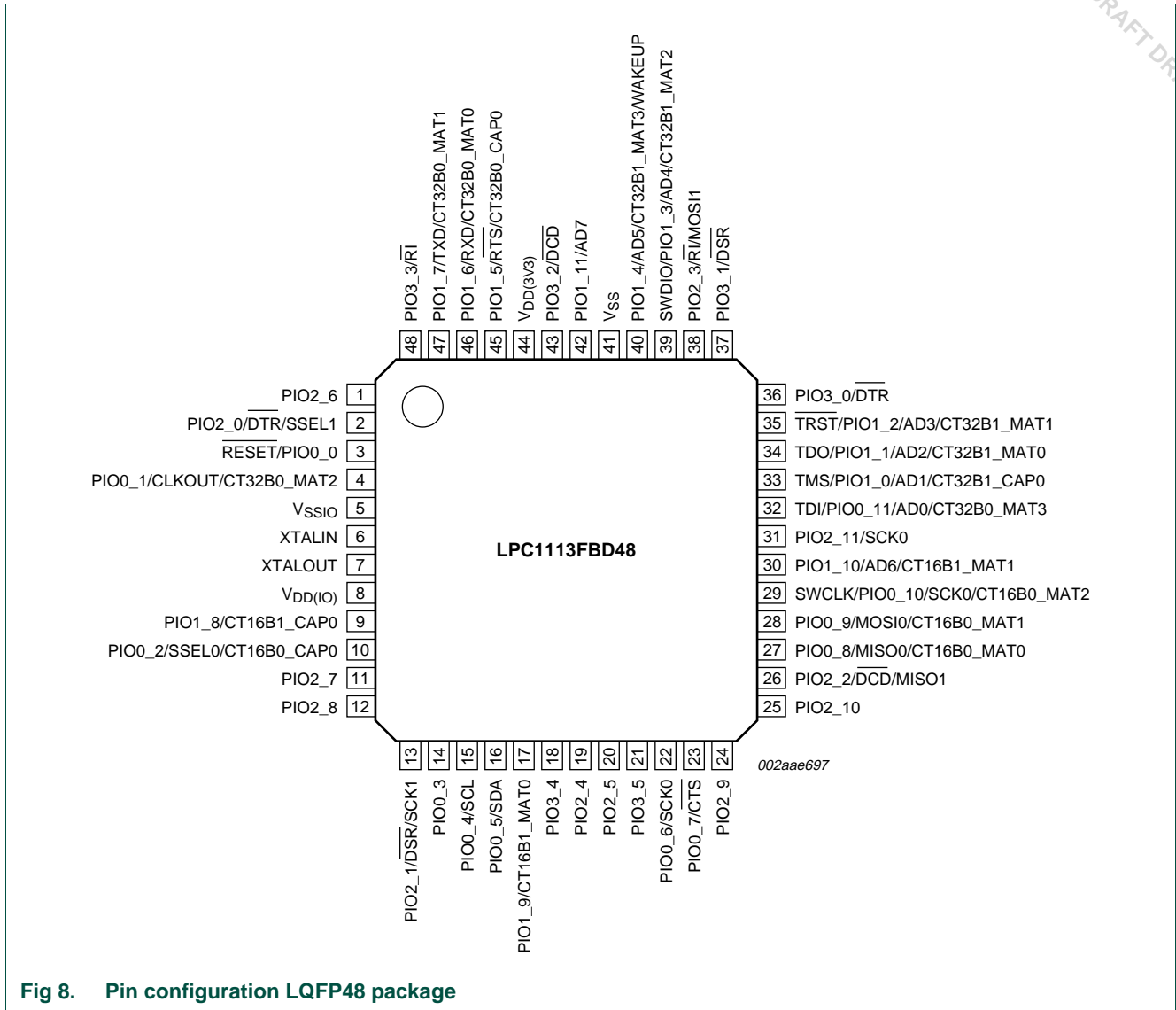


Fig 8. Pin configuration LQFP48 package

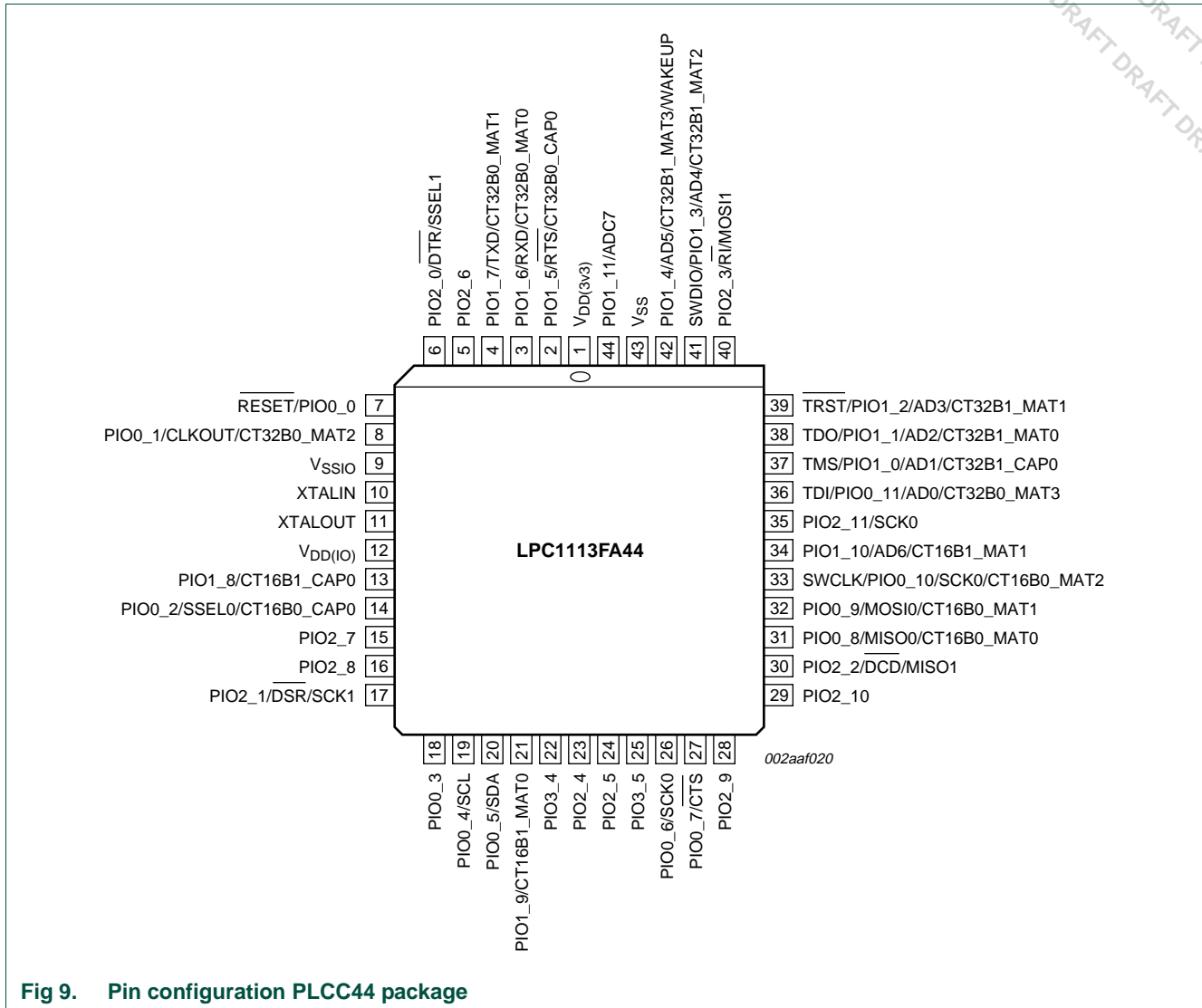


Fig 9. Pin configuration PLCC44 package

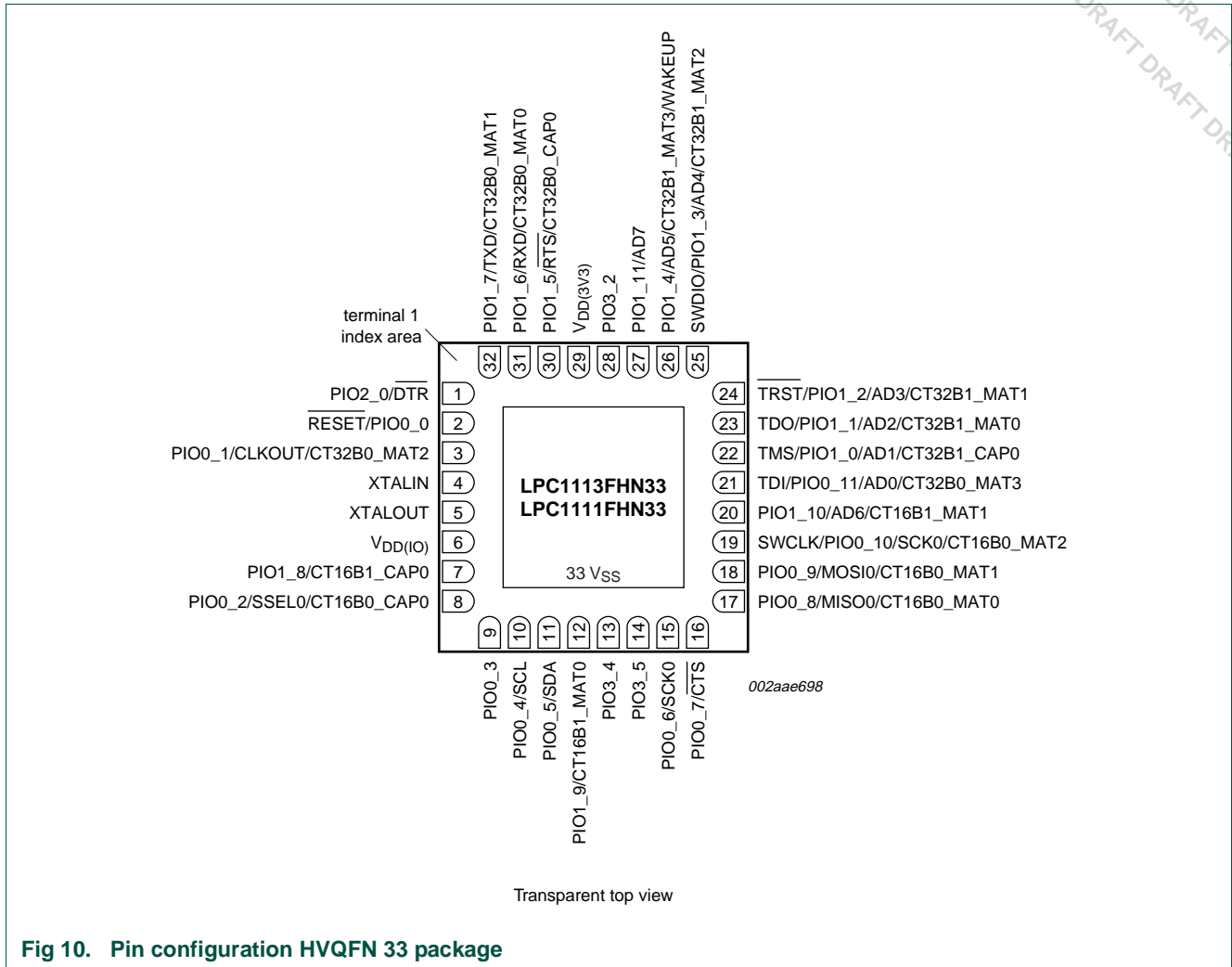


Fig 10. Pin configuration HVQFN 33 package

3. LPC111x Pin description

Table 110. LPC1113 pin description table (LQFP48 package)

Symbol	Pin	Type	Description
RESET/PIO0_0	3	I	RESET — External reset input: A LOW 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.
		I/O	PIO0_0 — General purpose digital input/output pin.
PIO0_1/CLKOUT/ CT32B0_MAT2	4 ^[1]	I/O	PIO0_1 — General purpose digital input/output pin. A LOW level on this pin during reset starts the ISP command handler.
		O	CLKOUT — Clockout pin.
		O	CT32B0_MAT2 — Match output 2 for 32-bit timer 0.
PIO0_2/SSEL0/ CT16B0_CAP0	10 ^[1]	I/O	PIO0_2 — General purpose digital input/output pin.
		O	SSEL0 — Slave Select for SSP0.
		I	CT16B0_CAP0 — Capture input 0 for 16-bit timer 0.
PIO0_3	14 ^[1]	I/O	PIO0_3 — General purpose digital input/output pin.

Table 110. LPC1113 pin description table (LQFP48 package) ...continued

Symbol	Pin	Type	Description
PIO0_4/SCL	15 ^[2]	I/O	PIO0_4 — General purpose digital input/output pin.
		I/O	SCL — I ² C-bus clock input/output. High-current sink only if I ² C Fast-mode Plus is selected in the I/O configuration register.
PIO0_5/SDA	16 ^[2]	I/O	PIO0_5 — General purpose digital input/output pin.
		I/O	SDA — I ² C-bus data input/output. High-current sink only if I ² C Fast-mode Plus is selected in the I/O configuration register.
PIO0_6/SCK0	22 ^[1]	I/O	PIO0_6 — General purpose digital input/output pin.
		I/O	SCK0 — Serial clock for SSP0.
PIO0_7/ $\overline{\text{CTS}}$	23 ^[1]	I/O	PIO0_7 — General purpose digital input/output pin (high-current output driver).
		I	CTS — Clear To Send input for UART.
PIO0_8/MISO0/ CT16B0_MAT0	27 ^[1]	I/O	PIO0_8 — General purpose digital input/output pin.
		I/O	MISO0 — Master In Slave Out for SSP0.
		O	CT16B0_MAT0 — Match output 0 for 16-bit timer 0.
PIO0_9/MOSI0/ CT16B0_MAT1	28 ^[1]	I/O	PIO0_9 — General purpose digital input/output pin.
		I/O	MOSI0 — Master Out Slave In for SSP0.
		O	CT16B0_MAT1 — Match output 1 for 16-bit timer 0.
SWCLK/PIO0_10/ SCK0/CT16B0_MAT2	29 ^[1]	I	SWCLK — Serial wire clock and test clock TCK for JTAG interface.
		I/O	PIO0_10 — General purpose digital input/output pin.
		O	SCK0 — Serial clock for SSP0.
		O	CT16B0_MAT2 — Match output 2 for 16-bit timer 0.
TDI/PIO0_11/ AD0/CT32B0_MAT3	32 ^[3]	I	TDI — Test Data In for JTAG interface.
		I/O	PIO0_11 — General purpose digital input/output pin.
		I	AD0 — A/D converter, input 0.
		O	CT32B0_MAT3 — Match output 3 for 32-bit timer 0.
TMS/PIO1_0/ AD1/CT32B1_CAP0	33 ^[3]	I	TMS — Test Mode Select for JTAG interface.
		I/O	PIO1_0 — General purpose digital input/output pin.
		I	AD1 — A/D converter, input 1.
		I	CT32B1_CAP0 — Capture input 0 for 32-bit timer 1.
TDO/PIO1_1/ AD2/CT32B1_MAT0	34 ^[3]	O	TDO — Test Data Out for JTAG interface.
		I/O	PIO1_1 — General purpose digital input/output pin.
		I	AD2 — A/D converter, input 2.
		O	CT32B1_MAT0 — Match output 0 for 32-bit timer 1.
$\overline{\text{TRST}}$ /PIO1_2/ AD3/CT32B1_MAT1	35 ^[3]	I	TRST — Test Reset for JTAG interface.
		I/O	PIO1_2 — General purpose digital input/output pin.
		I	AD3 — A/D converter, input 3.
		O	CT32B1_MAT1 — Match output 1 for 32-bit timer 1.
SWDIO/PIO1_3/AD4/ CT32B1_MAT2	39 ^[3]	I/O	SWDIO — Serial wire debug input/output.
		I/O	PIO1_3 — General purpose digital input/output pin.
		I	AD4 — A/D converter, input 4.
		O	CT32B1_MAT2 — Match output 2 for 32-bit timer 1.

Table 110. LPC1113 pin description table (LQFP48 package) ...continued

Symbol	Pin	Type	Description
PIO1_4/AD5/ CT32B1_MAT3/WAKEUP	40 ^[3]	I/O	PIO1_4 — General purpose digital input/output pin.
		I	AD5 — A/D converter, input 5.
		O	CT32B1_MAT3 — Match output 3 for 32-bit timer 1.
		I	WAKEUP — Deep power-down mode wake-up pin.
PIO1_5/RTS/ CT32B0_CAP0	45 ^[1]	I/O	PIO1_5 — General purpose digital input/output pin.
		O	RTS — Request To Send output for UART.
		I	CT32B0_CAP0 — Capture input 0 for 32-bit timer 0.
PIO1_6/RXD/ CT32B0_MAT0	46 ^[1]	I/O	PIO1_6 — General purpose digital input/output pin.
		I	RXD — Receiver input for UART.
		O	CT32B0_MAT0 — Match output 0 for 32-bit timer 0.
PIO1_7/TXD/ CT32B0_MAT1	47 ^[1]	I/O	PIO1_7 — General purpose digital input/output pin.
		O	TXD — Transmitter output for UART.
		O	CT32B0_MAT1 — Match output 1 for 32-bit timer 0.
PIO1_8/CT16B1_CAP0	9 ^[1]	I/O	PIO1_8 — General purpose digital input/output pin.
		I	CT16B1_CAP0 — Capture input 0 for 16-bit timer 1.
PIO1_9/CT16B1_MAT0	17 ^[1]	I/O	PIO1_9 — General purpose digital input/output pin.
		O	CT16B1_MAT0 — Match output 0 for 16-bit timer 1.
PIO1_10/AD6/ CT16B1_MAT1	30 ^[3]	I/O	PIO1_10 — General purpose digital input/output pin.
		I	AD6 — A/D converter, input 6.
		O	CT16B1_MAT1 — Match output 1 for 16-bit timer 1.
PIO1_11/AD7	42 ^[3]	I/O	PIO1_11 — General purpose digital input/output pin.
		I	AD7 — A/D converter, input 7.
PIO2_0/DTR/SSEL1	2 ^[1]	I/O	PIO2_0 — General purpose digital input/output pin.
		O	DTR — Data Terminal Ready output for UART.
		O	SSEL1 — Slave Select for SSP1.
PIO2_1/DSR/SCK1	13 ^[1]	I/O	PIO2_1 — General purpose digital input/output pin.
		I	DSR — Data Set Ready input for UART.
		I/O	SCK1 — Serial clock for SSP1.
PIO2_2/DCD/MISO1	26 ^[1]	I/O	PIO2_2 — General purpose digital input/output pin.
		I	DCD — Data Carrier Detect input for UART.
		I/O	MISO1 — Master In Slave Out for SSP1.
PIO2_3/RI/MOSI1	38 ^[1]	I/O	PIO2_3 — General purpose digital input/output pin.
		I	RI — Ring Indicator input for UART.
		I/O	MOSI1 — Master Out Slave In for SSP1.
PIO2_4	19 ^[1]	I/O	PIO2_4 — General purpose digital input/output pin.
PIO2_5	20 ^[1]	I/O	PIO2_5 — General purpose digital input/output pin.
PIO2_6	1 ^[1]	I/O	PIO2_6 — General purpose digital input/output pin.
PIO2_7	11 ^[1]	I/O	PIO2_7 — General purpose digital input/output pin.
PIO2_8	12 ^[1]	I/O	PIO2_8 — General purpose digital input/output pin.
PIO2_9	24 ^[1]	I/O	PIO2_9 — General purpose digital input/output pin.
PIO2_10	25 ^[1]	I/O	PIO2_10 — General purpose digital input/output pin.

Table 110. LPC1113 pin description table (LQFP48 package) ...continued

Symbol	Pin	Type	Description
PIO2_11/SCK0	31 ^[1]	I/O	PIO2_11 — General purpose digital input/output pin.
		I/O	SCK0 — Serial clock for SSP0.
PIO3_0/DTR	36 ^[1]	I/O	PIO3_0 — General purpose digital input/output pin.
		O	DTR — Data Terminal Ready output for UART.
PIO3_1/DSR	37 ^[1]	I/O	PIO3_1 — General purpose digital input/output pin.
		I	DSR — Data Set Ready input for UART.
PIO3_2/DCD	43 ^[1]	I/O	PIO3_2 — General purpose digital input/output pin.
		I	DCD — Data Carrier Detect input for UART.
PIO3_3/RI	48 ^[1]	I/O	PIO3_3 — General purpose digital input/output pin.
		I	RI — Ring Indicator input for UART.
PIO3_4	18 ^[1]	I/O	PIO3_4 — General purpose digital input/output pin.
PIO3_5	21 ^[1]	I/O	PIO3_5 — General purpose digital input/output pin.
V _{DD(I/O)}	8	I	3.3 V input/output supply voltage.
V _{DD(3V3)}	44	I	3.3 V supply voltage to the internal regulator and the ADC. Also used as the ADC reference voltage.
V _{SSIO}	5	I	Ground.
XTALIN	6 ^[4]	I	Input to the oscillator circuit and internal clock generator circuits. Input voltage must not exceed 1.8 V.
XTALOUT	7 ^[4]	O	Output from the oscillator amplifier.
V _{SS}	41	I	Ground.

- [1] 5 V tolerant pad providing digital I/O functions with configurable pull-up/pull-down resistors and configurable hysteresis.
- [2] I²C-bus pads compliant with the I²C-bus specification for I²C standard mode and I²C Fast-mode Plus.
- [3] 5 V tolerant pad providing digital I/O functions with configurable pull-up/pull-down resistors, configurable 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.
- [4] When the system oscillator is not used, connect XTALIN and XTALOUT as follows: XTALIN can be left floating or can be grounded (grounding is preferred to reduce susceptibility to noise). XTALOUT should be left floating.

Table 111. LPC1113 pin description table (PLCC44 package)

Symbol	Pin	Type	Description
RESET/PIO0_0	7	I	RESET — External reset input: A LOW 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.
		I/O	PIO0_0 — General purpose digital input/output pin.
PIO0_1/CLKOUT/ CT32B0_MAT2	8 ^[1]	I/O	PIO0_1 — General purpose digital input/output pin. A LOW level on this pin during reset starts the ISP command handler.
		O	CLKOUT — Clockout pin.
		O	CT32B0_MAT2 — Match output 2 for 32-bit timer 0.
PIO0_2/SSEL0/ CT16B0_CAP0	14 ^[1]	I/O	PIO0_2 — General purpose digital input/output pin.
		O	SSEL0 — Slave Select for SSP0.
		I	CT16B0_CAP0 — Capture input 0 for 16-bit timer 0.
PIO0_3	18 ^[1]	I/O	PIO0_3 — General purpose digital input/output pin.

Table 111. LPC1113 pin description table (PLCC44 package) ...continued

Symbol	Pin	Type	Description
PIO0_4/SCL	19 ^[2]	I/O	PIO0_4 — General purpose digital input/output pin.
		I/O	SCL — I ² C-bus clock input/output. High-current sink only if I ² C Fast-mode Plus is selected in the I/O configuration register.
PIO0_5/SDA	20 ^[2]	I/O	PIO0_5 — General purpose digital input/output pin.
		I/O	SDA — I ² C-bus data input/output. High-current sink only if I ² C Fast-mode Plus is selected in the I/O configuration register.
PIO0_6/SCK0	26 ^[1]	I/O	PIO0_6 — General purpose digital input/output pin.
		I/O	SCK0 — Serial clock for SSP0.
PIO0_7/ $\overline{\text{CTS}}$	27 ^[1]	I/O	PIO0_7 — General purpose digital input/output pin (high-current output driver).
		I	$\overline{\text{CTS}}$ — Clear To Send input for UART.
PIO0_8/MISO0/ CT16B0_MAT0	31 ^[1]	I/O	PIO0_8 — General purpose digital input/output pin.
		I/O	MISO0 — Master In Slave Out for SSP0.
		O	CT16B0_MAT0 — Match output 0 for 16-bit timer 0.
PIO0_9/MOSI0/ CT16B0_MAT1	32 ^[1]	I/O	PIO0_9 — General purpose digital input/output pin.
		I/O	MOSI0 — Master Out Slave In for SSP0.
		O	CT16B0_MAT1 — Match output 1 for 16-bit timer 0.
SWCLK/PIO0_10/ SCK0/CT16B0_MAT2	33 ^[1]	I	SWCLK — Serial wire clock and test clock TCK for JTAG interface.
		I/O	PIO0_10 — General purpose digital input/output pin.
		O	SCK0 — Serial clock for SSP0.
		O	CT16B0_MAT2 — Match output 2 for 16-bit timer 0.
TDI/PIO0_11/ AD0/CT32B0_MAT3	36 ^[3]	I	TDI — Test Data In for JTAG interface.
		I/O	PIO0_11 — General purpose digital input/output pin.
		I	AD0 — A/D converter, input 0.
		O	CT32B0_MAT3 — Match output 3 for 32-bit timer 0.
TMS/PIO1_0/ AD1/CT32B1_CAP0	37 ^[3]	I	TMS — Test Mode Select for JTAG interface.
		I/O	PIO1_0 — General purpose digital input/output pin.
		I	AD1 — A/D converter, input 1.
		I	CT32B1_CAP0 — Capture input 0 for 32-bit timer 1.
TDO/PIO1_1/ AD2/CT32B1_MAT0	38 ^[3]	O	TDO — Test Data Out for JTAG interface.
		I/O	PIO1_1 — General purpose digital input/output pin.
		I	AD2 — A/D converter, input 2.
		O	CT32B1_MAT0 — Match output 0 for 32-bit timer 1.
$\overline{\text{TRST}}$ /PIO1_2/ AD3/CT32B1_MAT1	39 ^[3]	I	$\overline{\text{TRST}}$ — Test Reset for JTAG interface.
		I/O	PIO1_2 — General purpose digital input/output pin.
		I	AD3 — A/D converter, input 3.
		O	CT32B1_MAT1 — Match output 1 for 32-bit timer 1.
SWDIO/PIO1_3/AD4/ CT32B1_MAT2	41 ^[3]	I/O	SWDIO — Serial wire debug input/output.
		I/O	PIO1_3 — General purpose digital input/output pin.
		I	AD4 — A/D converter, input 4.
		O	CT32B1_MAT2 — Match output 2 for 32-bit timer 1.

Table 111. LPC1113 pin description table (PLCC44 package) ...continued

Symbol	Pin	Type	Description
PIO1_4/AD5/ CT32B1_MAT3/WAKEUP	42 ^[3]	I/O	PIO1_4 — General purpose digital input/output pin.
		I	AD5 — A/D converter, input 5.
		O	CT32B1_MAT3 — Match output 3 for 32-bit timer 1.
		I	WAKEUP — Deep power-down mode wake-up pin.
PIO1_5/RTS/ CT32B0_CAP0	2 ^[1]	I/O	PIO1_5 — General purpose digital input/output pin.
		O	RTS — Request To Send output for UART.
		I	CT32B0_CAP0 — Capture input 0 for 32-bit timer 0.
PIO1_6/RXD/ CT32B0_MAT0	3 ^[1]	I/O	PIO1_6 — General purpose digital input/output pin.
		I	RXD — Receiver input for UART.
		O	CT32B0_MAT0 — Match output 0 for 32-bit timer 0.
PIO1_7/TXD/ CT32B0_MAT1	4 ^[1]	I/O	PIO1_7 — General purpose digital input/output pin.
		O	TXD — Transmitter output for UART.
		O	CT32B0_MAT1 — Match output 1 for 32-bit timer 0.
PIO1_8/CT16B1_CAP0	13 ^[1]	I/O	PIO1_8 — General purpose digital input/output pin.
		I	CT16B1_CAP0 — Capture input 0 for 16-bit timer 1.
PIO1_9/CT16B1_MAT0	21 ^[1]	I/O	PIO1_9 — General purpose digital input/output pin.
		O	CT16B1_MAT0 — Match output 0 for 16-bit timer 1.
PIO1_10/AD6/ CT16B1_MAT1	34 ^[3]	I/O	PIO1_10 — General purpose digital input/output pin.
		I	AD6 — A/D converter, input 6.
		O	CT16B1_MAT1 — Match output 1 for 16-bit timer 1.
PIO1_11/AD7	44 ^[3]	I/O	PIO1_11 — General purpose digital input/output pin.
		I	AD7 — A/D converter, input 7.
PIO2_0/DTR/SSEL1	6 ^[1]	I/O	PIO2_0 — General purpose digital input/output pin.
		O	DTR — Data Terminal Ready output for UART.
		O	SSEL1 — Slave Select for SSP1.
PIO2_1/DSR/SCK1	17 ^[1]	I/O	PIO2_1 — General purpose digital input/output pin.
		I	DSR — Data Set Ready input for UART.
		I/O	SCK1 — Serial clock for SSP1.
PIO2_2/DCD/MISO1	30 ^[1]	I/O	PIO2_2 — General purpose digital input/output pin.
		I	DCD — Data Carrier Detect input for UART.
		I/O	MISO1 — Master In Slave Out for SSP1.
PIO2_3/R \bar{I} /MOSI1	40 ^[1]	I/O	PIO2_3 — General purpose digital input/output pin.
		I	R\bar{I} — Ring Indicator input for UART.
		I/O	MOSI1 — Master Out Slave In for SSP1.
PIO2_4	23 ^[1]	I/O	PIO2_4 — General purpose digital input/output pin.
PIO2_5	24 ^[1]	I/O	PIO2_5 — General purpose digital input/output pin.
PIO2_6	5 ^[1]	I/O	PIO2_6 — General purpose digital input/output pin.
PIO2_7	15 ^[1]	I/O	PIO2_7 — General purpose digital input/output pin.
PIO2_8	16 ^[1]	I/O	PIO2_8 — General purpose digital input/output pin.
PIO2_9	28 ^[1]	I/O	PIO2_9 — General purpose digital input/output pin.
PIO2_10	29 ^[1]	I/O	PIO2_10 — General purpose digital input/output pin.

Table 111. LPC1113 pin description table (PLCC44 package) ...continued

Symbol	Pin	Type	Description
PIO2_11/SCK0	35 ^[1]	I/O	PIO2_11 — General purpose digital input/output pin.
		I/O	SCK0 — Serial clock for SSP0.
PIO3_4	22 ^[1]	I/O	PIO3_4 — General purpose digital input/output pin.
PIO3_5	25 ^[1]	I/O	PIO3_5 — General purpose digital input/output pin.
V _{DD(I/O)}	12	I	3.3 V input/output supply voltage.
V _{DD(3V3)}	1	I	3.3 V supply voltage to the internal regulator and the ADC. Also used as the ADC reference voltage.
V _{SSIO}	9	I	Ground.
XTALIN	10 ^[4]	I	Input to the oscillator circuit and internal clock generator circuits. Input voltage must not exceed 1.8 V.
XTALOUT	11 ^[4]	O	Output from the oscillator amplifier.
V _{SS}	43	I	Ground.

- [1] 5 V tolerant pad providing digital I/O functions with configurable pull-up/pull-down resistors and configurable hysteresis.
- [2] I²C-bus pads compliant with the I²C-bus specification for I²C standard mode and I²C Fast-mode Plus.
- [3] 5 V tolerant pad providing digital I/O functions with configurable pull-up/pull-down resistors, configurable 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.
- [4] When the system oscillator is not used, connect XTALIN and XTALOUT as follows: XTALIN can be left floating or can be grounded (grounding is preferred to reduce susceptibility to noise). XTALOUT should be left floating.

Table 112. LPC1111/1112/1113 pin description table (HVQFN33 package)

Symbol	Pin	Type	Description
RESET/PIO0_0	2	I	RESET — External reset input: A LOW 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.
		I/O	PIO0_0 — General purpose digital input/output pin.
PIO0_1/CLKOUT/ CT32B0_MAT2	3 ^[1]	I/O	PIO0_1 — General purpose digital input/output pin. A LOW level on this pin during reset starts the ISP command handler.
		O	CLKOUT — Clock out pin.
		O	CT32B0_MAT2 — Match output 2 for 32-bit timer 0.
PIO0_2/SSEL0/ CT16B0_CAP0	8 ^[1]	I/O	PIO0_2 — General purpose digital input/output pin.
		O	SSEL0 — Slave select for SSP0.
		I	CT16B0_CAP0 — Capture input 0 for 16-bit timer 0.
PIO0_3	9 ^[1]	I/O	PIO0_3 — General purpose digital input/output pin.
PIO0_4/SCL	10 ^[2]	I/O	PIO0_4 — General purpose digital input/output pin.
		I/O	SCL — I ² C-bus clock input/output. High-current sink only if I ² C Fast-mode Plus is selected in the I/O configuration register.
PIO0_5/SDA	11 ^[2]	I/O	PIO0_5 — General purpose digital input/output pin.
		I/O	SDA — I ² C-bus data input/output. High-current sink only if I ² C Fast-mode Plus is selected in the I/O configuration register.
PIO0_6/SCK0	15 ^[1]	I/O	PIO0_6 — General purpose digital input/output pin.
		I/O	SCK0 — Serial clock for SSP0.
PIO0_7/ $\overline{\text{CTS}}$	16 ^[1]	I/O	PIO0_7 — General purpose digital input/output pin (high-current output driver).
		I	CTS — Clear To Send input for UART.

Table 112. LPC1111/1112/1113 pin description table (HVQFN33 package) ...continued

Symbol	Pin	Type	Description
PIO0_8/MISO0/ CT16B0_MAT0	17 ^[1]	I/O	PIO0_8 — General purpose digital input/output pin.
		I/O	MISO0 — Master In Slave Out for SSP0.
		O	CT16B0_MAT0 — Match output 0 for 16-bit timer 0.
PIO0_9/MOSI0/ CT16B0_MAT1	18 ^[1]	I/O	PIO0_9 — General purpose digital input/output pin.
		I/O	MOSI0 — Master Out Slave In for SSP0.
		O	CT16B0_MAT1 — Match output 1 for 16-bit timer 0.
SWCLK/PIO0_10/SCK0/ CT16B0_MAT2	19 ^[1]	I	SWCLK — Serial wire clock and test clock TCK for JTAG interface.
		I/O	PIO0_10 — General purpose digital input/output pin.
		O	SCK0 — Serial clock for SSP0.
		O	CT16B0_MAT2 — Match output 2 for 16-bit timer 0.
TDI/PIO0_11/AD0/ CT32B0_MAT3	21 ^[3]	I	TDI — Test Data In for JTAG interface.
		I/O	PIO0_11 — General purpose digital input/output pin.
		I	AD0 — A/D converter, input 0.
		O	CT32B0_MAT3 — Match output 3 for 32-bit timer 0.
TMS/PIO1_0/AD1/ CT32B1_CAP0	22 ^[3]	I	TMS — Test Mode Select for JTAG interface.
		I/O	PIO1_0 — General purpose digital input/output pin.
		I	AD1 — A/D converter, input 1.
		I	CT32B1_CAP0 — Capture input 0 for 32-bit timer 1.
TDO/PIO1_1/AD2/ CT32B1_MAT0	23 ^[3]	O	TDO — Test Data Out for JTAG interface.
		I/O	PIO1_1 — General purpose digital input/output pin.
		I	AD2 — A/D converter, input 2.
		O	CT32B1_MAT0 — Match output 0 for 32-bit timer 1.
TRST/PIO1_2/AD3/ CT32B1_MAT1	24 ^[3]	I	TRST — Test Reset for JTAG interface.
		I/O	PIO1_2 — General purpose digital input/output pin.
		I	AD3 — A/D converter, input 3.
		O	CT32B1_MAT1 — Match output 1 for 32-bit timer 1.
SWDIO/PIO1_3/AD4/ CT32B1_MAT2	25 ^[3]	I/O	SWDIO — Serial wire debug input/output.
		I/O	PIO1_3 — General purpose digital input/output pin.
		I	AD4 — A/D converter, input 4.
		O	CT32B1_MAT2 — Match output 2 for 32-bit timer 1.
PIO1_4/AD5/ CT32B1_MAT3/WAKEUP	26 ^[3]	I/O	PIO1_4 — General purpose digital input/output pin.
		I	AD5 — A/D converter, input 5.
		O	CT32B1_MAT3 — Match output 3 for 32-bit timer 1.
		I	WAKEUP — Deep power-down mode wake-up pin.
PIO1_5/RTS/ CT32B0_CAP0	30 ^[1]	I/O	PIO1_5 — General purpose digital input/output pin.
		O	RTS — Request To Send output for UART.
		I	CT32B0_CAP0 — Capture input 0 for 32-bit timer 0.
PIO1_6/RXD/ CT32B0_MAT0	31 ^[1]	I/O	PIO1_6 — General purpose digital input/output pin.
		I	RXD — Receiver input for UART.
		O	CT32B0_MAT0 — Match output 0 for 32-bit timer 0.

Table 112. LPC1111/1112/1113 pin description table (HVQFN33 package) ...continued

Symbol	Pin	Type	Description
PIO1_7/TXD/ CT32B0_MAT1	32 ^[1]	I/O	PIO1_7 — General purpose digital input/output pin.
		O	TXD — Transmitter output for UART.
		O	CT32B0_MAT1 — Match output 1 for 32-bit timer 0.
PIO1_8/CT16B1_CAP0	7 ^[1]	I/O	PIO1_8 — General purpose digital input/output pin.
		I	CT16B1_CAP0 — Capture input 0 for 16-bit timer 1.
PIO1_9/CT16B1_MAT0	12 ^[1]	I/O	PIO1_9 — General purpose digital input/output pin.
		O	CT16B1_MAT0 — Match output 0 for 16-bit timer 1.
PIO1_10/AD6/ CT16B1_MAT1	20 ^[3]	I/O	PIO1_10 — General purpose digital input/output pin.
		I	AD6 — A/D converter, input 6.
		O	CT16B1_MAT1 — Match output 1 for 16-bit timer 1.
PIO1_11/AD7	27 ^[3]	I/O	PIO1_11 — General purpose digital input/output pin.
		I	AD7 — A/D converter, input 7.
PIO2_0/DTR	1 ^[1]	I/O	PIO2_0 — General purpose digital input/output pin.
		O	DTR — Data Terminal Ready output for UART.
PIO3_2	28 ^[1]	I/O	PIO3_2 — General purpose digital input/output pin.
PIO3_4	13 ^[1]	I/O	PIO3_4 — General purpose digital input/output pin.
PIO3_5	14 ^[1]	I/O	PIO3_5 — General purpose digital input/output pin.
V _{DD(I/O)}	6	I	3.3 V input/output supply voltage.
V _{DD(3V3)}	29	I	3.3 V supply voltage to the internal DC-DC converter and the ADC. Also used as the ADC reference voltage.
XTALIN	4 ^[4]	I	Input to the oscillator circuit and internal clock generator circuits. Input voltage must not exceed 1.8 V.
XTALOUT	5 ^[4]	O	Output from the oscillator amplifier.
V _{SS}	33	-	Thermal pad. Connect to ground.

- [1] 5 V tolerant pad providing digital I/O functions with configurable pull-up/pull-down resistors and configurable hysteresis.
- [2] I²C-bus pads compliant with the I²C-bus specification for I²C standard mode and I²C Fast-mode Plus.
- [3] 5 V tolerant pad providing digital I/O functions with configurable pull-up/pull-down resistors, configurable 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.
- [4] When the system oscillator is not used, connect XTALIN and XTALOUT as follows: XTALIN can be left floating or can be grounded (grounding is preferred to reduce susceptibility to noise). XTALOUT should be left floating.

UM10398

Chapter 9: LPC111x Universal Asynchronous Receiver/Transmitter (UART)

Rev. 00.06 — 19 October 2009

User manual

1. How to read this chapter

The UART block is identical for all LPC111x parts. The $\overline{\text{DSR}}$, $\overline{\text{DCD}}$, and $\overline{\text{RI}}$ modem signals are pinned out for LQFP48 and PLCC44 packages only.

2. Features

- 16-byte receive and transmit FIFOs.
- Register locations conform to '550 industry standard.
- Receiver FIFO trigger points at 1, 4, 8, and 14 bytes.
- Built-in baud rate generator.
- UART allows for implementation of either software or hardware flow control.
- RS-485/EIA-485 9-bit mode support with output enable.
- Modem control.

3. Pin description

Table 113. UART pin description

Pin	Type	Description
RXD	Input	Serial Input. Serial receive data.
TXD	Output	Serial Output. Serial transmit data.
RTS	Output	Request To Send. RS-485 direction control pin.
DTR	Output	Data Terminal Ready.
$\overline{\text{DSR}}$ ^[1]	Input	Data Set Ready.
CTS	Input	Clear To Send.
$\overline{\text{DCD}}$ ^[1]	Input	Data Carrier Detect.
$\overline{\text{RI}}$ ^[1]	Input	Ring Indicator.

[1] LQFP48 packages only.

The $\overline{\text{DSR}}$, $\overline{\text{DCD}}$, and $\overline{\text{RI}}$ modem inputs are multiplexed to two different pin locations. Use the IOCON_LOC registers (see [Section 7-4.2](#)) to select a physical location for each function on the LQFP48 pin package in addition to selecting the function in the IOCON registers.

The $\overline{\text{DTR}}$ output is available in two pin locations as well. The output value of the $\overline{\text{DTR}}$ pin is driven in both locations identically, and the $\overline{\text{DTR}}$ function at any location can be selected simply by selecting the function in the IOCON register for that pin location.

4. Clocking and power control

The UART block is gated by the AHCLKCTRL register (see [Table 3–19](#)). The peripheral UART clock, which is used by the UART baud rate generator, is controlled by the UARTCLKDIV register (see [Table 3–21](#)).

The UART_PCLK can be disabled in the UARTCLKDIV register (see [Section 3–4.16](#)) and the UART block can be disabled through the System AHB clock control register bit 12 (see [Section 3–4.14](#)) for power savings.

Remark: The UART pins must be configured in the corresponding IOCON registers **before** the UART clocks are enabled.

5. Register description

The UART contains registers organized as shown in [Table 9–114](#). The Divisor Latch Access Bit (DLAB) is contained in U0LCR[7] and enables access to the Divisor Latches.

Table 114. Register overview: UART (base address: 0x4000 8000)

Name	Access	Address offset	Description	Reset Value ^[1]	Notes
U0RBR	RO	0x000	Receiver Buffer Register. Contains the next received character to be read.	NA	when DLAB=0
U0THR	WO	0x000	Transmit Holding Register. The next character to be transmitted is written here.	NA	when DLAB=0
U0DLL	R/W	0x000	Divisor Latch LSB. Least significant byte of the baud rate divisor value. The full divisor is used to generate a baud rate from the fractional rate divider.	0x01	when DLAB=1
U0DLM	R/W	0x004	Divisor Latch MSB. Most significant byte of the baud rate divisor value. The full divisor is used to generate a baud rate from the fractional rate divider.	0x00	when DLAB=1
U0IER	R/W	0x004	Interrupt Enable Register. Contains individual interrupt enable bits for the 7 potential UART interrupts.	0x00	when DLAB=0
U0IIR	RO	0x008	Interrupt ID Register. Identifies which interrupt(s) are pending.	0x01	-
U0FCR	WO	0x008	FIFO Control Register. Controls UART FIFO usage and modes.	0x00	-
U0LCR	R/W	0x00C	Line Control Register. Contains controls for frame formatting and break generation.	0x00	-
U0MCR	R/W	0x010	Modem control register	0x00	-
U0LSR	RO	0x014	Line Status Register. Contains flags for transmit and receive status, including line errors.	0x60	-
U0MSR	RO	0x018	Modem status register	0x00	-
U0SCR	R/W	0x01C	Scratch Pad Register. Eight-bit temporary storage for software.	0x00	-
U0ACR	R/W	0x020	Auto-baud Control Register. Contains controls for the auto-baud feature.	0x00	-
-	-	0x024	Reserved	-	-
U0FDR	R/W	0x028	Fractional Divider Register. Generates a clock input for the baud rate divider.	0x10	-
-	-	0x02C	Reserved	-	-
U0TER	R/W	0x030	Transmit Enable Register. Turns off UART transmitter for use with software flow control.	0x80	-
-	-	0x034 - 0x048	Reserved	-	-
U0RS485CTRL	R/W	0x04C	RS-485/EIA-485 Control. Contains controls to configure various aspects of RS-485/EIA-485 modes.	0x00	-
U0ADRMATCH	R/W	0x050	RS-485/EIA-485 address match. Contains the address match value for RS-485/EIA-485 mode.	0x00	-
U0RS485DLY	R/W	0x054	RS-485/EIA-485 direction control delay.	0x00	-
U0FIFOLVL	RO	0x058	FIFO Level register. Provides the current fill levels of the transmit and receive FIFOs.	0x00	-

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

5.1 UART Receiver Buffer Register (U0RBR - 0x4000 8000, when DLAB = 0, Read Only)

The U0RBR is the top byte of the UART RX FIFO. The top byte of the RX FIFO contains the oldest character received and can be read via the bus interface. The LSB (bit 0) represents the “oldest” received data bit. If the character received is less than 8 bits, the unused MSBs are padded with zeroes.

The Divisor Latch Access Bit (DLAB) in U0LCR must be zero in order to access the U0RBR. The U0RBR is always Read Only.

Since PE, FE and BI bits (see [Table 9–126](#)) correspond to the byte sitting on the top of the RBR FIFO (i.e. the one that will be read in the next read from the RBR), the right approach for fetching the valid pair of received byte and its status bits is first to read the content of the U0LSR register, and then to read a byte from the U0RBR.

Table 115. UART Receiver Buffer Register (U0RBR - address 0x4000 8000 when DLAB = 0, Read Only) bit description

Bit	Symbol	Description	Reset Value
7:0	RBR	The UART Receiver Buffer Register contains the oldest received byte in the UART RX FIFO.	undefined
31:8	-	Reserved	-

5.2 UART Transmitter Holding Register (U0THR - 0x4000 8000 when DLAB = 0, Write Only)

The U0THR is the top byte of the UART TX FIFO. The top byte is the newest character in the TX FIFO and can be written via the bus interface. The LSB represents the first bit to transmit.

The Divisor Latch Access Bit (DLAB) in U0LCR must be zero in order to access the U0THR. The U0THR is always Write Only.

Table 116. UART Transmitter Holding Register (U0THR - address 0x4000 8000 when DLAB = 0, Write Only) bit description

Bit	Symbol	Description	Reset Value
7:0	THR	Writing to the UART Transmit Holding Register causes the data to be stored in the UART transmit FIFO. The byte will be sent when it reaches the bottom of the FIFO and the transmitter is available.	NA
31:8	-	Reserved	-

5.3 UART Divisor Latch LSB and MSB Registers (U0DLL - 0x4000 8000 and U0DLM - 0x4000 8004, when DLAB = 1)

The UART Divisor Latch is part of the UART Baud Rate Generator and holds the value used, along with the Fractional Divider, to divide the UART_PCLK clock in order to produce the baud rate clock, which must be 16x the desired baud rate. The U0DLL and U0DLM registers together form a 16-bit divisor where U0DLL contains the lower 8 bits of the divisor and U0DLM contains the higher 8 bits of the divisor. A 0x0000 value is treated like a 0x0001 value as division by zero is not allowed. The Divisor Latch Access Bit (DLAB) in U0LCR must be one in order to access the UART Divisor Latches. Details on how to select the right value for U0DLL and U0DLM can be found in [Section 9–5.15](#).

Table 117. UART Divisor Latch LSB Register (U0DLL - address 0x4000 8000 when DLAB = 1) bit description

Bit	Symbol	Description	Reset value
7:0	DLLSB	The UART Divisor Latch LSB Register, along with the U0DLM register, determines the baud rate of the UART.	0x01
31:8	-	Reserved	-

Table 118. UART Divisor Latch MSB Register (U0DLM - address 0x4000 8004 when DLAB = 1) bit description

Bit	Symbol	Description	Reset value
7:0	DLMSB	The UART Divisor Latch MSB Register, along with the U0DLL register, determines the baud rate of the UART.	0x00
31:8	-	Reserved	-

5.4 UART Interrupt Enable Register (U0IER - 0x4000 8004, when DLAB = 0)

The U0IER is used to enable the four UART interrupt sources.

Table 119. UART Interrupt Enable Register (U0IER - address 0x4000 8004 when DLAB = 0) bit description

Bit	Symbol	Value	Description	Reset value
0	RBR Interrupt Enable		Enables the Receive Data Available interrupt for UART. It also controls the Character Receive Time-out interrupt.	0
		0	Disable the RDA interrupt.	
		1	Enable the RDA interrupt.	
1	THRE Interrupt Enable		Enables the THRE interrupt for UART. The status of this interrupt can be read from U0LSR[5].	0
		0	Disable the THRE interrupt.	
		1	Enable the THRE interrupt.	
2	RX Line Interrupt Enable		Enables the UART RX line status interrupts. The status of this interrupt can be read from U0LSR[4:1].	0
		0	Disable the RX line status interrupts.	
		1	Enable the RX line status interrupts.	
3	-	-	Reserved	-
6:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
7	-	-	Reserved	0
8	ABEOIntEn		Enables the end of auto-baud interrupt.	0
		0	Disable end of auto-baud Interrupt.	
		1	Enable end of auto-baud Interrupt.	
9	ABTOIntEn		Enables the auto-baud time-out interrupt.	0
		0	Disable auto-baud time-out Interrupt.	
		1	Enable auto-baud time-out Interrupt.	
31:10	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

5.5 UART Interrupt Identification Register (U0IIR - 0x4004 8008, Read Only)

U0IIR provides a status code that denotes the priority and source of a pending interrupt. The interrupts are frozen during a U0IIR access. If an interrupt occurs during a U0IIR access, the interrupt is recorded for the next U0IIR access.

Table 120. UART Interrupt Identification Register (U0IIR - address 0x4004 8008, Read Only) bit description

Bit	Symbol	Value	Description	Reset value
0	IntStatus		Interrupt status. Note that U0IIR[0] is active low. The pending interrupt can be determined by evaluating U0IIR[3:1].	1
		0	At least one interrupt is pending.	
		1	No interrupt is pending.	
3:1	IntId		Interrupt identification. U0IER[3:1] identifies an interrupt corresponding to the UART Rx FIFO. All other combinations of U0IER[3:1] not listed below are reserved (100,101,111).	0
		011	1 - Receive Line Status (RLS).	
		010	2a - Receive Data Available (RDA).	
		110	2b - Character Time-out Indicator (CTI).	
		001	3 - THRE Interrupt.	
		000	4 - Modem interrupt.	
5:4	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
7:6	FIFO Enable		These bits are equivalent to U0FCR[0].	0
8	ABEOInt		End of auto-baud interrupt. True if auto-baud has finished successfully and interrupt is enabled.	0
9	ABTOInt		Auto-baud time-out interrupt. True if auto-baud has timed out and interrupt is enabled.	0
31:10	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Bits U0IIR[9:8] are set by the auto-baud function and signal a time-out or end of auto-baud condition. The auto-baud interrupt conditions are cleared by setting the corresponding Clear bits in the Auto-baud Control Register.

If the IntStatus bit is one and no interrupt is pending and the IntId bits will be zero. If the IntStatus is 0, a non auto-baud interrupt is pending in which case the IntId bits identify the type of interrupt and handling as described in [Table 9-121](#). Given the status of U0IIR[3:0], an interrupt handler routine can determine the cause of the interrupt and how to clear the active interrupt. The U0IIR must be read in order to clear the interrupt prior to exiting the Interrupt Service Routine.

The UART RLS interrupt (U0IIR[3:1] = 011) is the highest priority interrupt and is set whenever any one of four error conditions occur on the UART RX input: overrun error (OE), parity error (PE), framing error (FE) and break interrupt (BI). The UART Rx error condition that set the interrupt can be observed via U0LSR[4:1]. The interrupt is cleared upon a U0LSR read.

The UART RDA interrupt (U0IIR[3:1] = 010) shares the second level priority with the CTI interrupt (U0IIR[3:1] = 110). The RDA is activated when the UART Rx FIFO reaches the trigger level defined in U0FCR7:6 and is reset when the UART Rx FIFO depth falls below the trigger level. When the RDA interrupt goes active, the CPU can read a block of data defined by the trigger level.

The CTI interrupt (U0IIR[3:1] = 110) is a second level interrupt and is set when the UART Rx FIFO contains at least one character and no UART Rx FIFO activity has occurred in 3.5 to 4.5 character times. Any UART Rx FIFO activity (read or write of UART RSR) will clear the interrupt. This interrupt is intended to flush the UART RBR after a message has been received that is not a multiple of the trigger level size. For example, if a peripheral wished to send a 105 character message and the trigger level was 10 characters, the CPU would receive 10 RDA interrupts resulting in the transfer of 100 characters and 1 to 5 CTI interrupts (depending on the service routine) resulting in the transfer of the remaining 5 characters.

Table 121. UART Interrupt Handling

U0IIR[3:0] value ^[1]	Priority	Interrupt type	Interrupt source	Interrupt reset
0001	-	None	None	-
0110	Highest	RX Line Status / Error	OE ^[2] or PE ^[2] or FE ^[2] or BI ^[2]	U0LSR Read ^[2]
0100	Second	RX Data Available	Rx data available or trigger level reached in FIFO (U0FCR0=1)	U0RBR Read ^[3] or UART FIFO drops below trigger level
1100	Second	Character Time-out indication	Minimum of one character in the RX FIFO and no character input or removed during a time period depending on how many characters are in FIFO and what the trigger level is set at (3.5 to 4.5 character times). The exact time will be: $[(\text{word length}) \times 7 - 2] \times 8 + [(\text{trigger level} - \text{number of characters}) \times 8 + 1] \text{ RCLKs}$	U0RBR Read ^[3]
0010	Third	THRE	THRE ^[2]	U0IIR Read ^[4] (if source of interrupt) or THR write

[1] Values "0000", "0011", "0101", "0111", "1000", "1001", "1010", "1011", "1101", "1110", "1111" are reserved.
 [2] For details see [Section 9–5.9 “UART Line Status Register \(U0LSR - 0x4000 8014, Read Only\)”](#)
 [3] For details see [Section 9–5.1 “UART Receiver Buffer Register \(U0RBR - 0x4000 8000, when DLAB = 0, Read Only\)”](#)
 [4] For details see [Section 9–5.5 “UART Interrupt Identification Register \(U0IIR - 0x4004 8008, Read Only\)”](#) and [Section 9–5.2 “UART Transmitter Holding Register \(U0THR - 0x4000 8000 when DLAB = 0, Write Only\)”](#)

The UART THRE interrupt (U0IIR[3:1] = 001) is a third level interrupt and is activated when the UART THR FIFO is empty provided certain initialization conditions have been met. These initialization conditions are intended to give the UART THR FIFO a chance to fill up with data to eliminate many THRE interrupts from occurring at system start-up. The

initialization conditions implement a one character delay minus the stop bit whenever THRE = 1 and there have not been at least two characters in the U0THR at one time since the last THRE = 1 event. This delay is provided to give the CPU time to write data to U0THR without a THRE interrupt to decode and service. A THRE interrupt is set immediately if the UART THR FIFO has held two or more characters at one time and currently, the U0THR is empty. The THRE interrupt is reset when a U0THR write occurs or a read of the U0IIR occurs and the THRE is the highest interrupt (U0IIR[3:1] = 001).

5.6 UART FIFO Control Register (U0FCR - 0x4000 8008, Write Only)

The U0FCR controls the operation of the UART RX and TX FIFOs.

Table 122. UART FIFO Control Register (U0FCR - address 0x4000 8008, Write Only) bit description

Bit	Symbol	Value	Description	Reset value
0	FIFO Enable	0	UART FIFOs are disabled. Must not be used in the application.	0
		1	Active high enable for both UART Rx and TX FIFOs and U0FCR[7:1] access. This bit must be set for proper UART operation. Any transition on this bit will automatically clear the UART FIFOs.	
1	RX FIFO Reset	0	No impact on either of UART FIFOs.	0
		1	Writing a logic 1 to U0FCR[1] will clear all bytes in UART Rx FIFO, reset the pointer logic. This bit is self-clearing.	
2	TX FIFO Reset	0	No impact on either of UART FIFOs.	0
		1	Writing a logic 1 to U0FCR[2] will clear all bytes in UART TX FIFO, reset the pointer logic. This bit is self-clearing.	
3	-	-	Reserved	0
5:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
7:6	RX Trigger Level	-	These two bits determine how many receiver UART FIFO characters must be written before an interrupt is activated.	0
		00	Trigger level 0 (1 character or 0x01).	
		01	Trigger level 1 (4 characters or 0x04).	
		10	Trigger level 2 (8 characters or 0x08).	
31:8	-	-	11	Trigger level 3 (14 characters or 0x0E).
			-	Reserved

5.7 UART Modem Control Register

The U0MCR enables the modem loopback mode and controls the modem output signals.

Table 123. UART0 Modem Control Register (U0MCR - address 0x4000 8010) bit description

Bit	Symbol	Value	Description	Reset value
0	DTR Control		Source for modem output pin, $\overline{\text{DTR}}$. This bit reads as 0 when modem loopback mode is active.	0
1	RTS Control		Source for modem output pin $\overline{\text{RTS}}$. This bit reads as 0 when modem loopback mode is active.	0
3-2	-	NA	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	0
4	Loopback Mode Select		The modem loopback mode provides a mechanism to perform diagnostic loopback testing. Serial data from the transmitter is connected internally to serial input of the receiver. Input pin, RXD, has no effect on loopback and output pin, TXD is held in marking state. The four modem inputs (CTS, DSR, RI and DCD) are disconnected externally. Externally, the modem outputs (RTS, $\overline{\text{DTR}}$) are set inactive. Internally, the four modem outputs are connected to the four modem inputs. As a result of these connections, the upper four bits of the U0MSR will be driven by the lower four bits of the U0MCR rather than the four modem inputs in normal mode. This permits modem status interrupts to be generated in loopback mode by writing the lower four bits of U0MCR.	0
		0	Disable modem loopback mode.	
		1	Enable modem loopback mode.	
5	-	NA	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	0
6	RTSen	0	Disable auto-rts flow control.	0
		1	Enable auto-rts flow control.	
7	CTSen	0	Disable auto-cts flow control.	0
		1	Enable auto-cts flow control.	

5.7.1 Auto-flow control

If auto-RTS mode is enabled the UART's receiver FIFO hardware controls the $\overline{\text{RTS}}$ output of the UART. If the auto-CTS mode is enabled the UART's U0TSR hardware will only start transmitting if the $\overline{\text{CTS}}$ input signal is asserted.

5.7.1.1 Auto-RTS

The auto-RTS function is enabled by setting the RTSen bit. Auto-RTS data flow control originates in the U0RBR module and is linked to the programmed receiver FIFO trigger level. If auto-RTS is enabled, the data-flow is controlled as follows:

When the receiver FIFO level reaches the programmed trigger level, $\overline{\text{RTS}}$ is deasserted (to a high value). It is possible that the sending UART sends an additional byte after the trigger level is reached (assuming the sending UART has another byte to send) because it might not recognize the deassertion of $\overline{\text{RTS}}$ until after it has begun sending the additional byte. $\overline{\text{RTS}}$ is automatically reasserted (to a low value) once the receiver FIFO has reached the previous trigger level. The reassertion of $\overline{\text{RTS}}$ signals the sending UART to continue transmitting data.

If Auto-RTS mode is disabled, the RTSen bit controls the $\overline{\text{RTS}}$ output of the UART. If Auto-RTS mode is enabled, hardware controls the $\overline{\text{RTS}}$ output, and the actual value of $\overline{\text{RTS}}$ will be copied in the RTS Control bit of the UART. As long as Auto-RTS is enabled, the value of the RTS Control bit is read-only for software.

Example: Suppose the UART operating in type '550 mode has the trigger level in U0FCR set to 0x2, then, if Auto-RTS is enabled, the UART will deassert the $\overline{\text{RTS}}$ output as soon as the receive FIFO contains 8 bytes (Table 9-122 on page 101). The $\overline{\text{RTS}}$ output will be reasserted as soon as the receive FIFO hits the previous trigger level: 4 bytes.

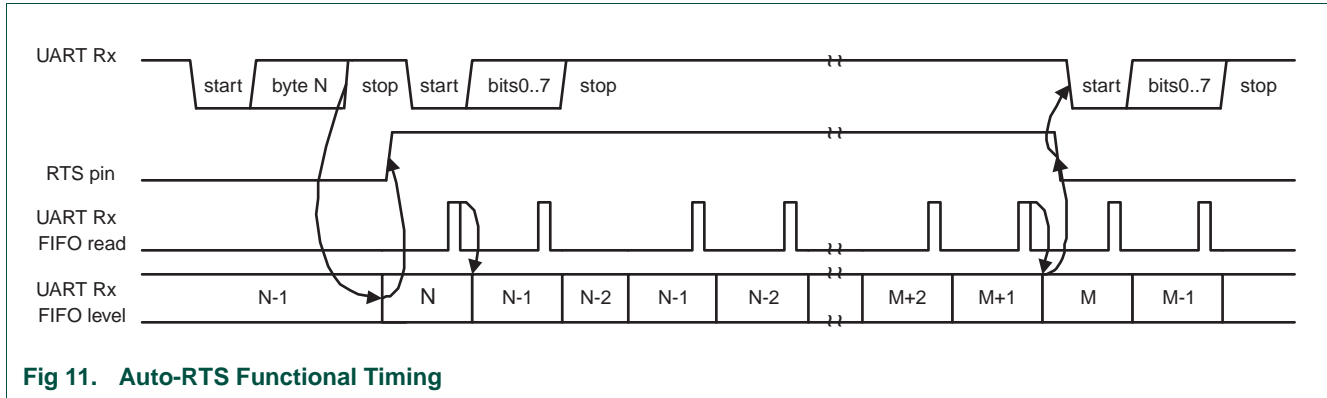


Fig 11. Auto-RTS Functional Timing

5.7.1.2 Auto-CTS

The Auto-CTS function is enabled by setting the CTSen bit. If Auto-CTS is enabled, the transmitter circuitry in the U0TSR module checks $\overline{\text{CTS}}$ input before sending the next data byte. When $\overline{\text{CTS}}$ is active (low), the transmitter sends the next byte. To stop the transmitter from sending the following byte, $\overline{\text{CTS}}$ must be released before the middle of the last stop bit that is currently being sent. In Auto-CTS mode, a change of the $\overline{\text{CTS}}$ signal does not trigger a modem status interrupt unless the CTS Interrupt Enable bit is set, Delta CTS bit in the U0MSR will be set though. Table 9-124 lists the conditions for generating a Modem Status interrupt.

Table 124. Modem status interrupt generation

Enable modem status interrupt (U0ER[3])	CTSen (U0MCR[7])	CTS interrupt enable (U0IER[7])	Delta CTS (U0MSR[0])	Delta DCD or trailing edge RI or Delta DSR (U0MSR[3] or U0MSR[2] or U0MSR[1])	Modem status interrupt
0	x	x	x	x	No
1	0	x	0	0	No
1	0	x	1	x	Yes
1	0	x	x	1	Yes
1	1	0	x	0	No
1	1	0	x	1	Yes
1	1	1	0	0	No
1	1	1	1	x	Yes
1	1	1	x	1	Yes

The auto-CTS function reduces interrupts to the host system. When flow control is enabled, a $\overline{\text{CTS}}$ state change does not trigger host interrupts because the device automatically controls its own transmitter. Without Auto-CTS, the transmitter sends any data present in the transmit FIFO and a receiver overrun error can result. [Figure 9–12](#) illustrates the Auto-CTS functional timing.

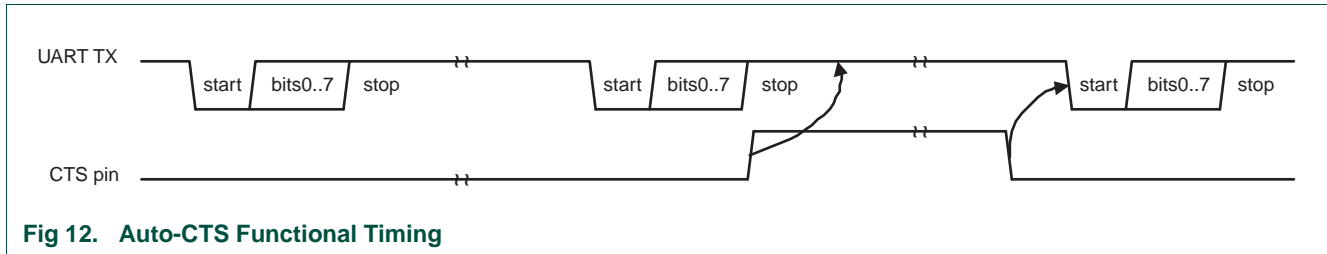


Fig 12. Auto-CTS Functional Timing

While starting transmission of the initial character, the $\overline{\text{CTS}}$ signal is asserted. Transmission will stall as soon as the pending transmission has completed. The UART will continue transmitting a 1 bit as long as $\overline{\text{CTS}}$ is de-asserted (high). As soon as $\overline{\text{CTS}}$ gets de-asserted, transmission resumes and a start bit is sent followed by the data bits of the next character.

5.8 UART Line Control Register (U0LCR - 0x4000 800C)

The U0LCR determines the format of the data character that is to be transmitted or received.

Table 125. UART Line Control Register (U0LCR - address 0x4000 800C) bit description

Bit	Symbol	Value	Description	Reset Value
1:0	Word Length Select	00	5-bit character length.	0
		01	6-bit character length.	
		10	7-bit character length.	
		11	8-bit character length.	
2	Stop Bit Select	0	1 stop bit.	0
		1	2 stop bits (1.5 if U0LCR[1:0]=00).	
3	Parity Enable	0	Disable parity generation and checking.	0
		1	Enable parity generation and checking.	
5:4	Parity Select	00	Odd parity. Number of 1s in the transmitted character and the attached parity bit will be odd.	0
		01	Even Parity. Number of 1s in the transmitted character and the attached parity bit will be even.	
		10	Forced "1" stick parity.	
		11	Forced "0" stick parity.	
6	Break Control	0	Disable break transmission.	0
		1	Enable break transmission. Output pin UART TXD is forced to logic 0 when U0LCR[6] is active high.	

Table 125. UART Line Control Register (U0LCR - address 0x4000 800C) bit description

Bit	Symbol	Value	Description	Reset Value
7	Divisor Latch Access Bit (DLAB)	0	Disable access to Divisor Latches.	0
		1	Enable access to Divisor Latches.	
31: 8	-	-	Reserved	-

5.9 UART Line Status Register (U0LSR - 0x4000 8014, Read Only)

The U0LSR is a Read Only register that provides status information on the UART TX and RX blocks.

Table 126. UART Line Status Register (U0LSR - address 0x4000 8014, Read Only) bit description

Bit	Symbol	Value	Description	Reset Value
0	Receiver Data Ready (RDR)		Receiver Data Ready :U0LSR[0] is set when the U0RBR holds an unread character and is cleared when the UART RBR FIFO is empty.	0
		0	U0RBR is empty.	
		1	U0RBR contains valid data.	
1	Overrun Error (OE)		The overrun error condition is set as soon as it occurs. A U0LSR read clears U0LSR[1]. U0LSR[1] is set when UART RSR has a new character assembled and the UART RBR FIFO is full. In this case, the UART RBR FIFO will not be overwritten and the character in the UART RSR will be lost.	0
		0	Overrun error status is inactive.	
		1	Overrun error status is active.	
2	Parity Error (PE)		When the parity bit of a received character is in the wrong state, a parity error occurs. A U0LSR read clears U0LSR[2]. Time of parity error detection is dependent on U0FCR[0]. Note: A parity error is associated with the character at the top of the UART RBR FIFO.	0
		0	Parity error status is inactive.	
		1	Parity error status is active.	
3	Framing Error (FE)		When the stop bit of a received character is a logic 0, a framing error occurs. A U0LSR read clears U0LSR[3]. The time of the framing error detection is dependent on U0FCR0. Upon detection of a framing error, the RX will attempt to re-synchronize to the data and assume that the bad stop bit is actually an early start bit. However, it cannot be assumed that the next received byte will be correct even if there is no Framing Error. Note: A framing error is associated with the character at the top of the UART RBR FIFO.	0
		0	Framing error status is inactive.	
		1	Framing error status is active.	

Table 126. UART Line Status Register (U0LSR - address 0x4000 8014, Read Only) bit description ...continued

Bit	Symbol	Value	Description	Reset Value
4	Break Interrupt (BI)		When RXD1 is held in the spacing state (all zeros) for one full character transmission (start, data, parity, stop), a break interrupt occurs. Once the break condition has been detected, the receiver goes idle until RXD1 goes to marking state (all ones). A U0LSR read clears this status bit. The time of break detection is dependent on U0FCR[0]. Note: The break interrupt is associated with the character at the top of the UART RBR FIFO.	0
		0	Break interrupt status is inactive.	
		1	Break interrupt status is active.	
5	Transmitter Holding Register Empty (THRE)		THRE is set immediately upon detection of an empty UART THR and is cleared on a U0THR write.	1
		0	U0THR contains valid data.	
		1	U0THR is empty.	
6	Transmitter Empty (TEMT)		TEMT is set when both U0THR and U0TSR are empty; TEMT is cleared when either the U0TSR or the U0THR contain valid data.	1
		0	U0THR and/or the U0TSR contains valid data.	
		1	U0THR and the U0TSR are empty.	
7	Error in RX FIFO (RXFE)		U0LSR[7] is set when a character with a RX error such as framing error, parity error or break interrupt, is loaded into the U0RBR. This bit is cleared when the U0LSR register is read and there are no subsequent errors in the UART FIFO.	0
		0	U0RBR contains no UART RX errors or U0FCR[0]=0.	
		1	UART RBR contains at least one UART RX error.	
31: 8	-	-	Reserved	-

5.10 UART Modem Status Register

The U0MSR is a read-only register that provides status information on the modem input signals. U0MSR[3:0] is cleared on U0MSR read. Note that modem signals have no direct effect on the UART operation. They facilitate the software implementation of modem signal operations.

Table 127. UART Modem Status Register (U0MSR - address 0x4000 8018) bit description

Bit	Symbol	Value	Description	Reset Value
0	Delta CTS		Set upon state change of input $\overline{\text{CTS}}$. Cleared on a U0MSR read.	0
		0	No change detected on modem input $\overline{\text{CTS}}$.	
		1	State change detected on modem input $\overline{\text{CTS}}$.	
1	Delta DSR		Set upon state change of input $\overline{\text{DSR}}$. Cleared on a U0MSR read.	0
		0	No change detected on modem input $\overline{\text{DSR}}$.	
		1	State change detected on modem input $\overline{\text{DSR}}$.	

Table 127. UART Modem Status Register (U0MSR - address 0x4000 8018) bit description

Bit	Symbol	Value	Description	Reset Value
2	Trailing Edge RI		Set upon low to high transition of input \overline{RI} . Cleared on a U0MSR read.	0
		0	No change detected on modem input, \overline{RI} .	
		1	Low-to-high transition detected on \overline{RI} .	
3	Delta DCD		Set upon state change of input \overline{DCD} . Cleared on a U0MSR read.	0
		0	No change detected on modem input \overline{DCD} .	
		1	State change detected on modem input \overline{DCD} .	
4	CTS		Clear To Send State. Complement of input signal \overline{CTS} . This bit is connected to U0MCR[1] in modem loopback mode.	0
5	DSR		Data Set Ready State. Complement of input signal \overline{DSR} . This bit is connected to U0MCR[0] in modem loopback mode.	0
6	RI		Ring Indicator State. Complement of input \overline{RI} . This bit is connected to U0MCR[2] in modem loopback mode.	0
7	DCD		Data Carrier Detect State. Complement of input \overline{DCD} . This bit is connected to U0MCR[3] in modem loopback mode.	0

5.11 UART Scratch Pad Register (U0SCR - 0x4000 801C)

The U0SCR has no effect on the UART operation. This register can be written and/or read at user's discretion. There is no provision in the interrupt interface that would indicate to the host that a read or write of the U0SCR has occurred.

Table 128. UART Scratch Pad Register (U0SCR - address 0x4000 8014) bit description

Bit	Symbol	Description	Reset Value
7:0	Pad	A readable, writable byte.	0x00
31:8	-	Reserved	-

5.12 UART Auto-baud Control Register (U0ACR - 0x4000 8020)

The UART Auto-baud Control Register (U0ACR) controls the process of measuring the incoming clock/data rate for the baud rate generation and can be read and written at user's discretion.

Table 129. Auto-baud Control Register (U0ACR - address 0x4000 8020) bit description

Bit	Symbol	Value	Description	Reset value
0	Start		This bit is automatically cleared after auto-baud completion.	0
		0	Auto-baud stop (auto-baud is not running).	
		1	Auto-baud start (auto-baud is running). Auto-baud run bit. This bit is automatically cleared after auto-baud completion.	
1	Mode		Auto-baud mode select bit.	0
		0	Mode 0.	
		1	Mode 1.	

Table 129. Auto-baud Control Register (U0ACR - address 0x4000 8020) bit description

Bit	Symbol	Value	Description	Reset value
2	AutoRestart	0	No restart	0
		1	Restart in case of time-out (counter restarts at next UART Rx falling edge)	0
7:3	-	NA	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	0
8	ABEOIntClr		End of auto-baud interrupt clear bit (write only accessible).	0
		0	Writing a 0 has no impact.	
		1	Writing a 1 will clear the corresponding interrupt in the U0IIR.	
9	ABTOIntClr		Auto-baud time-out interrupt clear bit (write only accessible).	0
		0	Writing a 0 has no impact.	
		1	Writing a 1 will clear the corresponding interrupt in the U0IIR.	
31:10	-	NA	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	0

5.13 Auto-baud

The UART auto-baud function can be used to measure the incoming baud rate based on the "AT" protocol (Hayes command). If enabled the auto-baud feature will measure the bit time of the receive data stream and set the divisor latch registers U0DLM and U0DLL accordingly.

Auto-baud is started by setting the U0ACR Start bit. Auto-baud can be stopped by clearing the U0ACR Start bit. The Start bit will clear once auto-baud has finished and reading the bit will return the status of auto-baud (pending/finished).

Two auto-baud measuring modes are available which can be selected by the U0ACR Mode bit. In Mode 0 the baud rate is measured on two subsequent falling edges of the UART Rx pin (the falling edge of the start bit and the falling edge of the least significant bit). In Mode 1 the baud rate is measured between the falling edge and the subsequent rising edge of the UART Rx pin (the length of the start bit).

The U0ACR AutoRestart bit can be used to automatically restart baud rate measurement if a time-out occurs (the rate measurement counter overflows). If this bit is set, the rate measurement will restart at the next falling edge of the UART Rx pin.

The auto-baud function can generate two interrupts.

- The U0IIR ABTOInt interrupt will get set if the interrupt is enabled (U0IER ABTOIntEn is set and the auto-baud rate measurement counter overflows).
- The U0IIR ABEOInt interrupt will get set if the interrupt is enabled (U0IER ABEOIntEn is set and the auto-baud has completed successfully).

The auto-baud interrupts have to be cleared by setting the corresponding U0ACR ABTOIntClr and ABEOIntEn bits.

The fractional baud rate generator must be disabled ($DIVADDVAL = 0$) during auto-baud. Also, when auto-baud is used, any write to U0DLM and U0DLL registers should be done before U0ACR register write. The minimum and the maximum baud rates supported by UART are function of $UART_PCLK$, number of data bits, stop bits and parity bits.

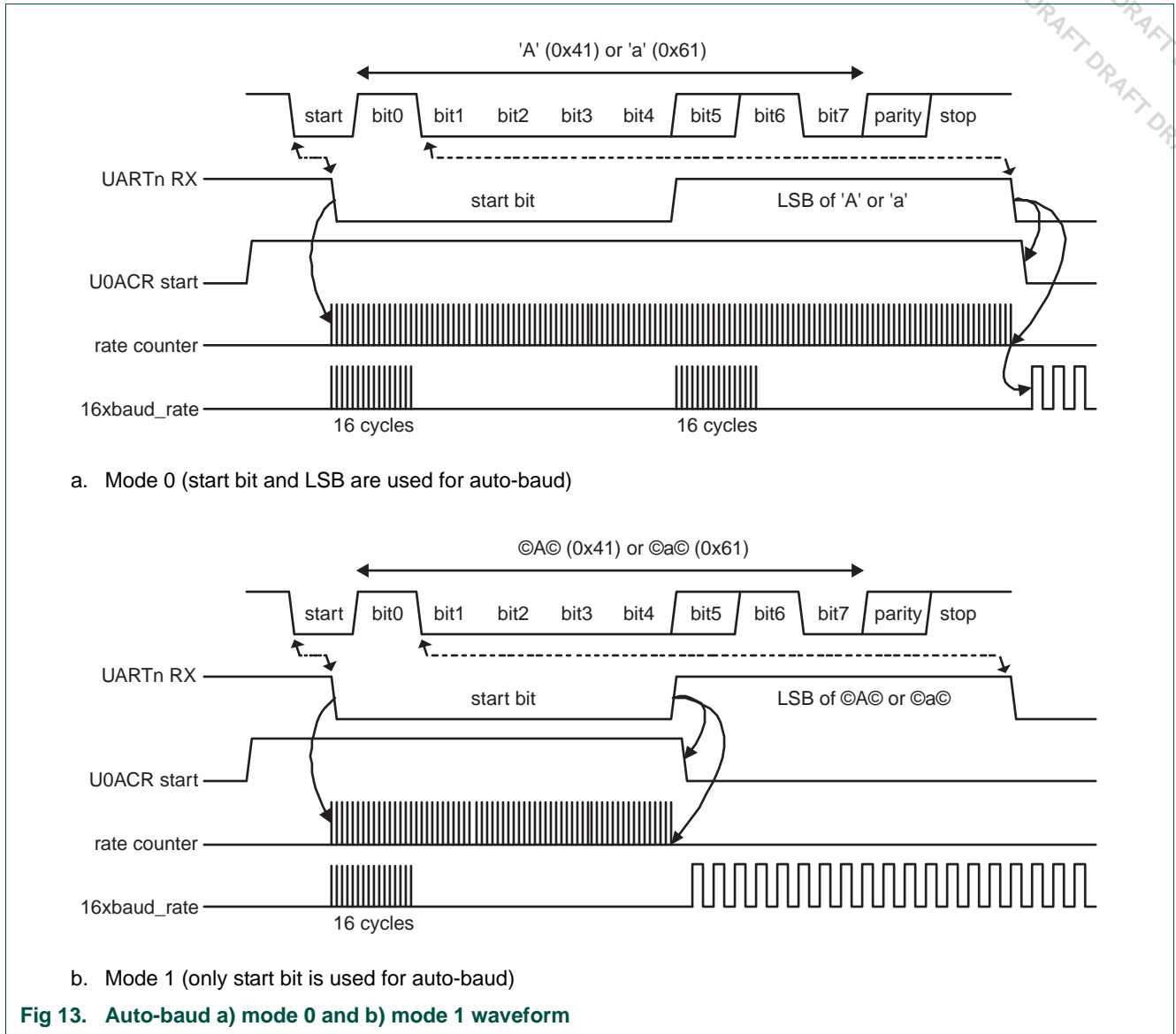
(6)

$$ratemin = \frac{2 \times PCLK}{16 \times 2^{15}} \leq UART_{baudrate} \leq \frac{PCLK}{16 \times (2 + databits + paritybits + stopbits)} = ratemax$$

5.14 Auto-baud modes

When the software is expecting an "AT" command, it configures the UART with the expected character format and sets the U0ACR Start bit. The initial values in the divisor latches U0DLM and U0DLL don't care. Because of the "A" or "a" ASCII coding ("A" = 0x41, "a" = 0x61), the UART Rx pin sensed start bit and the LSB of the expected character are delimited by two falling edges. When the U0ACR Start bit is set, the auto-baud protocol will execute the following phases:

1. On U0ACR Start bit setting, the baud rate measurement counter is reset and the UART U0RSR is reset. The U0RSR baud rate is switched to the highest rate.
2. A falling edge on UART Rx pin triggers the beginning of the start bit. The rate measuring counter will start counting $UART_PCLK$ cycles.
3. During the receipt of the start bit, 16 pulses are generated on the RSR baud input with the frequency of the UART input clock, guaranteeing the start bit is stored in the U0RSR.
4. During the receipt of the start bit (and the character LSB for Mode = 0), the rate counter will continue incrementing with the pre-scaled UART input clock ($UART_PCLK$).
5. If Mode = 0, the rate counter will stop on next falling edge of the UART Rx pin. If Mode = 1, the rate counter will stop on the next rising edge of the UART Rx pin.
6. The rate counter is loaded into U0DLM/U0DLL and the baud rate will be switched to normal operation. After setting the U0DLM/U0DLL, the end of auto-baud interrupt U0IIR ABEOInt will be set, if enabled. The U0RSR will now continue receiving the remaining bits of the "A/a" character.



5.15 UART Fractional Divider Register (U0FDR - 0x4000 8028)

The UART Fractional Divider Register (U0FDR) controls the clock pre-scaler for the baud rate generation and can be read and written at the user's discretion. This pre-scaler takes the APB clock and generates an output clock according to the specified fractional requirements.

Important: If the fractional divider is active (DIVADDDVAL > 0) and DLM = 0, the value of the DLL register must be 3 or greater.

Table 130. UART Fractional Divider Register (U0FDR - address 0x4000 8028) bit description

Bit	Function	Value	Description	Reset value
3:0	DIVADDVAL	0	Baud rate generation pre-scaler divisor value. If this field is 0, fractional baud rate generator will not impact the UART baud rate.	0
7:4	MULVAL	1	Baud rate pre-scaler multiplier value. This field must be greater or equal 1 for UART to operate properly, regardless of whether the fractional baud rate generator is used or not.	1
31:8	-	NA	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	0

This register controls the clock pre-scaler for the baud rate generation. The reset value of the register keeps the fractional capabilities of UART disabled making sure that UART is fully software and hardware compatible with UARTs not equipped with this feature.

The UART baud rate can be calculated as:

(7)

$$UART_{baudrate} = \frac{PCLK}{16 \times (256 \times U0DLM + U0DLL) \times \left(1 + \frac{DivAddVal}{MulVal}\right)}$$

Where UART_PCLK is the peripheral clock, U0DLM and U0DLL are the standard UART baud rate divider registers, and DIVADDVAL and MULVAL are UART fractional baud rate generator specific parameters.

The value of MULVAL and DIVADDVAL should comply to the following conditions:

1. $1 \leq MULVAL \leq 15$
2. $0 \leq DIVADDVAL \leq 14$
3. $DIVADDVAL < MULVAL$

The value of the U0FDR should not be modified while transmitting/receiving data or data may be lost or corrupted.

If the U0FDR register value does not comply to these two requests, then the fractional divider output is undefined. If DIVADDVAL is zero then the fractional divider is disabled, and the clock will not be divided.

5.15.1 Baud rate calculation

UART can operate with or without using the Fractional Divider. In real-life applications it is likely that the desired baud rate can be achieved using several different Fractional Divider settings. The following algorithm illustrates one way of finding a set of DLM, DLL, MULVAL, and DIVADDVAL values. Such set of parameters yields a baud rate with a relative error of less than 1.1% from the desired one.

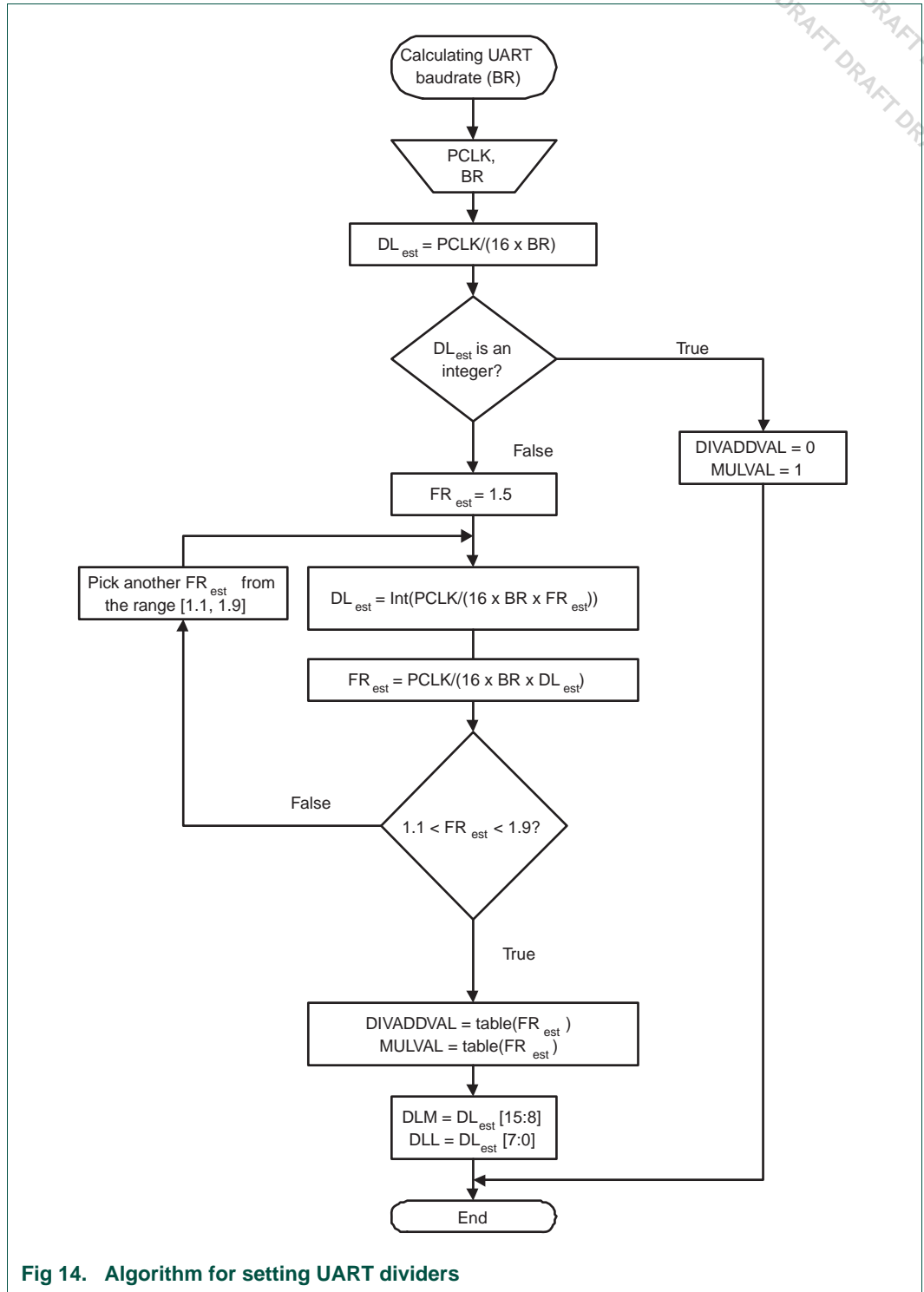


Fig 14. Algorithm for setting UART dividers

Table 131. Fractional Divider setting look-up table

FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal	FR	DivAddVal/ MulVal
1.000	0/1	1.250	1/4	1.500	1/2	1.750	3/4
1.067	1/15	1.267	4/15	1.533	8/15	1.769	10/13
1.071	1/14	1.273	3/11	1.538	7/13	1.778	7/9
1.077	1/13	1.286	2/7	1.545	6/11	1.786	11/14
1.083	1/12	1.300	3/10	1.556	5/9	1.800	4/5
1.091	1/11	1.308	4/13	1.571	4/7	1.818	9/11
1.100	1/10	1.333	1/3	1.583	7/12	1.833	5/6
1.111	1/9	1.357	5/14	1.600	3/5	1.846	11/13
1.125	1/8	1.364	4/11	1.615	8/13	1.857	6/7
1.133	2/15	1.375	3/8	1.625	5/8	1.867	13/15
1.143	1/7	1.385	5/13	1.636	7/11	1.875	7/8
1.154	2/13	1.400	2/5	1.643	9/14	1.889	8/9
1.167	1/6	1.417	5/12	1.667	2/3	1.900	9/10
1.182	2/11	1.429	3/7	1.692	9/13	1.909	10/11
1.200	1/5	1.444	4/9	1.700	7/10	1.917	11/12
1.214	3/14	1.455	5/11	1.714	5/7	1.923	12/13
1.222	2/9	1.462	6/13	1.727	8/11	1.929	13/14
1.231	3/13	1.467	7/15	1.733	11/15	1.933	14/15

5.15.1.1 Example 1: UART_PCLK = 14.7456 MHz, BR = 9600

According to the provided algorithm $DL_{est} = PCLK / (16 \times BR) = 14.7456 \text{ MHz} / (16 \times 9600) = 96$. Since this DL_{est} is an integer number, $DIVADDVAL = 0$, $MULVAL = 1$, $DLM = 0$, and $DLL = 96$.

5.15.1.2 Example 2: UART_PCLK = 12 MHz, BR = 115200

According to the provided algorithm $DL_{est} = PCLK / (16 \times BR) = 12 \text{ MHz} / (16 \times 115200) = 6.51$. This DL_{est} is not an integer number and the next step is to estimate the FR parameter. Using an initial estimate of $FR_{est} = 1.5$ a new $DL_{est} = 4$ is calculated and FR_{est} is recalculated as $FR_{est} = 1.628$. Since $FR_{est} = 1.628$ is within the specified range of 1.1 and 1.9, $DIVADDVAL$ and $MULVAL$ values can be obtained from the attached look-up table.

The closest value for $FR_{est} = 1.628$ in the look-up [Table 9–131](#) is $FR = 1.625$. It is equivalent to $DIVADDVAL = 5$ and $MULVAL = 8$.

Based on these findings, the suggested UART setup would be: $DLM = 0$, $DLL = 4$, $DIVADDVAL = 5$, and $MULVAL = 8$. According to [Equation 9–7](#), the UART's baud rate is 115384. This rate has a relative error of 0.16% from the originally specified 115200.

5.16 UART Transmit Enable Register (U0TER - 0x4000 8030)

In addition to being equipped with full hardware flow control (auto-cts and auto-rts mechanisms described above), U0TER enables implementation of software flow control. When $TxEn = 1$, UART transmitter will keep sending data as long as they are available. As soon as $TxEn$ becomes 0, UART transmission will stop.

Although [Table 9–132](#) describes how to use TxEn bit in order to achieve hardware flow control, it is strongly suggested to let UART hardware implemented auto flow control features take care of this, and limit the scope of TxEn to software flow control.

[Table 9–132](#) describes how to use TXEn bit in order to achieve software flow control.

Table 132. UART Transmit Enable Register (U0TER - address 0x4000 8030) bit description

Bit	Symbol	Description	Reset Value
6:0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
7	TXEN	When this bit is 1, as it is after a Reset, data written to the THR is output on the TXD pin as soon as any preceding data has been sent. If this bit cleared to 0 while a character is being sent, the transmission of that character is completed, but no further characters are sent until this bit is set again. In other words, a 0 in this bit blocks the transfer of characters from the THR or TX FIFO into the transmit shift register. Software can clear this bit when it detects that the a hardware-handshaking TX-permit signal (\overline{CTS}) has gone false, or with software handshaking, when it receives an XOFF character (DC3). Software can set this bit again when it detects that the TX-permit signal has gone true, or when it receives an XON (DC1) character.	1
31:8	-	Reserved	-

5.17 UART RS485 Control register (U0RS485CTRL - 0x4000 804C)

The U0RS485CTRL register controls the configuration of the UART in RS-485/EIA-485 mode.

Table 133. UART RS485 Control register (U0RS485CTRL - address 0x4000 804C) bit description

Bit	Symbol	Value	Description	Reset value
0	NMMEN	0	RS-485/EIA-485 Normal Multidrop Mode (NMM) is disabled.	0
		1	RS-485/EIA-485 Normal Multidrop Mode (NMM) is enabled. In this mode, an address is detected when a received byte causes the UART to set the parity error and generate an interrupt.	
1	RXDIS	0	The receiver is enabled.	0
		1	The receiver is disabled.	
2	AADEN	0	Auto Address Detect (AAD) is disabled.	0
		1	Auto Address Detect (AAD) is enabled.	
3	SEL	0	If direction control is enabled (bit DCTRL = 1), pin RTS is used for direction control.	0
		1	If direction control is enabled (bit DCTRL = 1), pin \overline{DTR} is used for direction control.	
4	DCTRL	0	Disable Auto Direction Control.	0
		1	Enable Auto Direction Control.	
5	OINV		This bit reverses the polarity of the direction control signal on the RTS (or \overline{DTR}) pin.	0

Table 133. UART RS485 Control register (U0RS485CTRL - address 0x4000 804C) bit description ...continued

Bit	Symbol	Value	Description	Reset value
		0	The direction control pin will be driven to logic '0' when the transmitter has data to be sent. It will be driven to logic '1' after the last bit of data has been transmitted.	
		1	The direction control pin will be driven to logic '1' when the transmitter has data to be sent. It will be driven to logic '0' after the last bit of data has been transmitted.	
31:6	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

5.18 UART RS-485 Address Match register (U0RS485ADRMATCH - 0x4000 8050)

The U0RS485ADRMATCH register contains the address match value for RS-485/EIA-485 mode.

Table 134. UART RS-485 Address Match register (U0RS485ADRMATCH - address 0x4000 8050) bit description

Bit	Symbol	Description	Reset value
7:0	ADRMATCH	Contains the address match value.	0x00
31:8	-	Reserved	-

5.19 UART1 RS-485 Delay value register (U0RS485DLY - 0x4000 8054)

The user may program the 8-bit RS485DLY register with a delay between the last stop bit leaving the TXFIFO and the de-assertion of RTS (or DTR). This delay time is in periods of the baud clock. Any delay time from 0 to 255 bit times may be programmed.

Table 135. UART RS-485 Delay value register (U0RS485DLY - address 0x4000 8054) bit description

Bit	Symbol	Description	Reset value
7:0	DLY	Contains the direction control (RTS or DTR) delay value. This register works in conjunction with an 8-bit counter.	0x00
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

5.20 RS-485/EIA-485 modes of operation

The RS-485/EIA-485 feature allows the UART to be configured as an addressable slave. The addressable slave is one of multiple slaves controlled by a single master.

The UART master transmitter will identify an address character by setting the parity (9th) bit to '1'. For data characters, the parity bit is set to '0'.

Each UART slave receiver can be assigned a unique address. The slave can be programmed to either manually or automatically reject data following an address which is not theirs.

RS-485/EIA-485 Normal Multidrop Mode (NMM)

Setting the RS485CTRL bit 0 enables this mode. In this mode, an address is detected when a received byte causes the UART to set the parity error and generate an interrupt.

If the receiver is disabled (RS485CTRL bit 1 = '1'), any received data bytes will be ignored and will not be stored in the RXFIFO. When an address byte is detected (parity bit = '1') it will be placed into the RXFIFO and an Rx Data Ready Interrupt will be generated. The processor can then read the address byte and decide whether or not to enable the receiver to accept the following data.

While the receiver is enabled (RS485CTRL bit 1 = '0'), all received bytes will be accepted and stored in the RXFIFO regardless of whether they are data or address. When an address character is received a parity error interrupt will be generated and the processor can decide whether or not to disable the receiver.

RS-485/EIA-485 Auto Address Detection (AAD) mode

When both RS485CTRL register bits 0 (9-bit mode enable) and 2 (AAD mode enable) are set, the UART is in auto address detect mode.

In this mode, the receiver will compare any address byte received (parity = '1') to the 8-bit value programmed into the RS485ADRMATCH register.

If the receiver is disabled (RS485CTRL bit 1 = '1'), any received byte will be discarded if it is either a data byte OR an address byte which fails to match the RS485ADRMATCH value.

When a matching address character is detected it will be pushed onto the RXFIFO along with the parity bit, and the receiver will be automatically enabled (RS485CTRL bit 1 will be cleared by hardware). The receiver will also generate an Rx Data Ready Interrupt.

While the receiver is enabled (RS485CTRL bit 1 = '0'), all bytes received will be accepted and stored in the RXFIFO until an address byte which does not match the RS485ADRMATCH value is received. When this occurs, the receiver will be automatically disabled in hardware (RS485CTRL bit 1 will be set), The received non-matching address character will not be stored in the RXFIFO.

RS-485/EIA-485 Auto Direction Control

RS485/EIA-485 mode includes the option of allowing the transmitter to automatically control the state of the DIR pin as a direction control output signal.

Setting RS485CTRL bit 4 = '1' enables this feature.

Direction control, if enabled, will use the $\overline{\text{RTS}}$ pin when RS485CTRL bit 3 = '0'. It will use the $\overline{\text{DTR}}$ pin when RS485CTRL bit 3 = '1'.

When Auto Direction Control is enabled, the selected pin will be asserted (driven LOW) when the CPU writes data into the TXFIFO. The pin will be de-asserted (driven HIGH) once the last bit of data has been transmitted. See bits 4 and 5 in the RS485CTRL register.

The RS485CTRL bit 4 takes precedence over all other mechanisms controlling the direction control pin with the exception of loopback mode.

RS485/EIA-485 driver delay time

The driver delay time is the delay between the last stop bit leaving the TXFIFO and the de-assertion of RTS. This delay time can be programmed in the 8-bit RS485DLY register. The delay time is in periods of the baud clock. Any delay time from 0 to 255 bit times may be used.

RS485/EIA-485 output inversion

The polarity of the direction control signal on the $\overline{\text{RTS}}$ (or $\overline{\text{DTR}}$) pins can be reversed by programming bit 5 in the U0RS485CTRL register. When this bit is set, the direction control pin will be driven to logic 1 when the transmitter has data waiting to be sent. The direction control pin will be driven to logic 0 after the last bit of data has been transmitted.

5.21 UART FIFO Level register (U0FIFOLVL - 0x4000 8058, Read Only)

U0FIFOLVL register is a Read Only register that allows software to read the current FIFO level status. Both the transmit and receive FIFO levels are present in this register.

Table 136. UART FIFO Level register (U0FIFOLVL - address 0x4000 8058, Read Only) bit description

Bit	Symbol	Description	Reset value
3:0	RXFIFILVL	Reflects the current level of the UART receiver FIFO. 0 = empty, 0xF = FIFO full.	0x00
7:4	-	Reserved. The value read from a reserved bit is not defined.	NA
11:8	TXFIFOLVL	Reflects the current level of the UART transmitter FIFO. 0 = empty, 0xF = FIFO full.	0x00
31:12	-	Reserved. The value read from a reserved bit is not defined.	NA

6. Architecture

The architecture of the UART is shown below in the block diagram.

The APB interface provides a communications link between the CPU or host and the UART.

The UART receiver block, U0RX, monitors the serial input line, RXD, for valid input. The UART RX Shift Register (U0RSR) accepts valid characters via RXD. After a valid character is assembled in the U0RSR, it is passed to the UART RX Buffer Register FIFO to await access by the CPU or host via the generic host interface.

The UART transmitter block, U0TX, accepts data written by the CPU or host and buffers the data in the UART TX Holding Register FIFO (U0THR). The UART TX Shift Register (U0TSR) reads the data stored in the U0THR and assembles the data to transmit via the serial output pin, TXD1.

The UART Baud Rate Generator block, U0BRG, generates the timing enables used by the UART TX block. The U0BRG clock input source is UART_PCLK. The main clock is divided down per the divisor specified in the U0DLL and U0DLM registers. This divided down clock is a 16x oversample clock, NBAUDOUT.

The interrupt interface contains registers U0IER and U0IIR. The interrupt interface receives several one clock wide enables from the U0TX and U0RX blocks.

Status information from the U0TX and U0RX is stored in the U0LSR. Control information for the U0TX and U0RX is stored in the U0LCR.

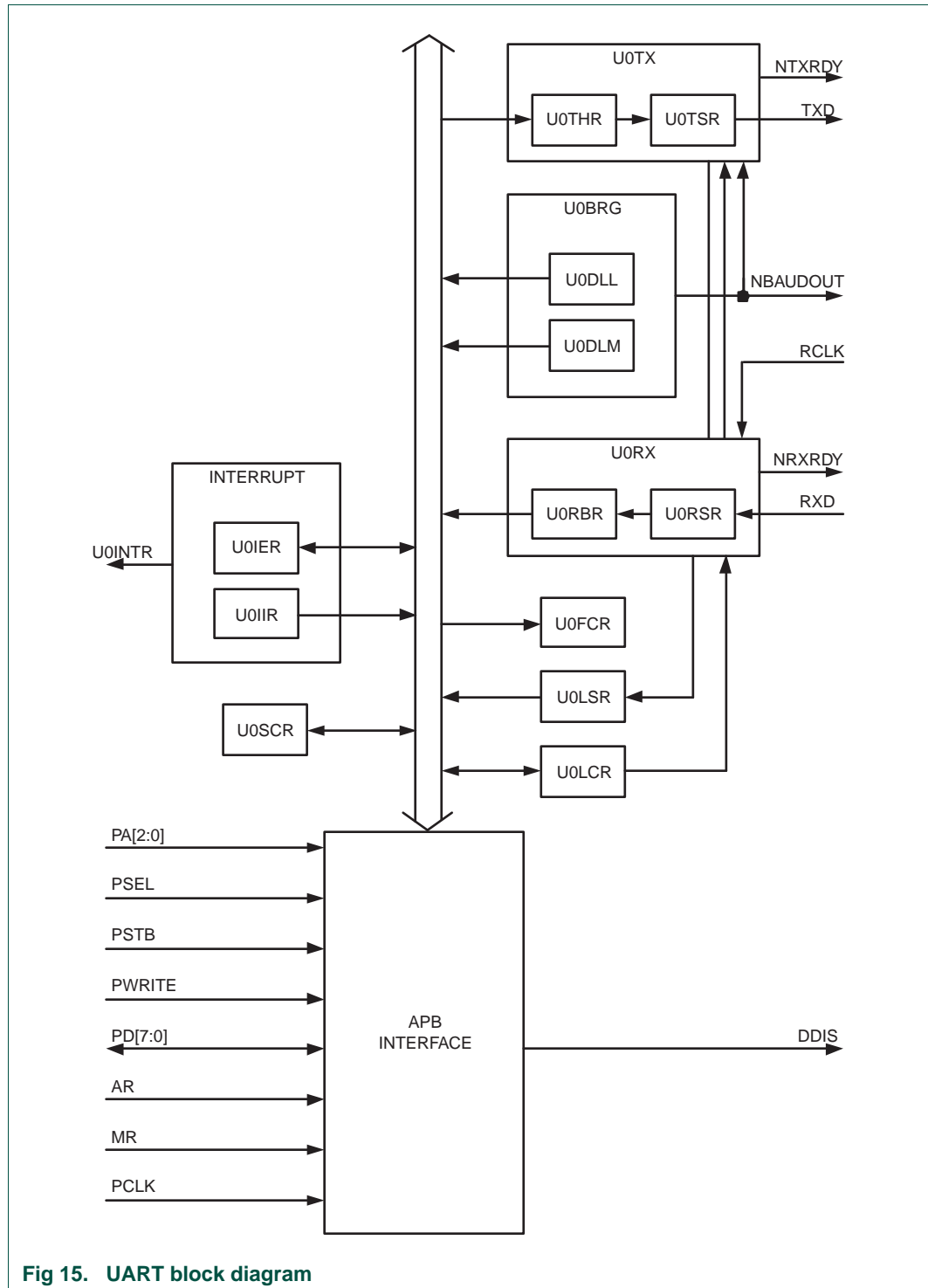


Fig 15. UART block diagram

1. How to read this chapter

The I²C-bus block is identical for all LPC111x parts.

2. Features

- Standard I²C-compliant bus interfaces may be configured as Master, Slave, or Master/Slave.
- Arbitration is handled between simultaneously transmitting masters without corruption of serial data on the bus.
- Programmable clock allows adjustment of I²C transfer rates.
- Data transfer is bidirectional between masters and slaves.
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus.
- Serial clock synchronization is used as a handshake mechanism to suspend and resume serial transfer.
- Supports Fast-mode Plus.
- Optional recognition of up to four distinct slave addresses.
- Monitor mode allows observing all I²C-bus traffic, regardless of slave address.
- I²C-bus can be used for test and diagnostic purposes.
- The I²C-bus contains a standard I²C-compliant bus interface with two pins.

3. Applications

Interfaces to external I²C standard parts, such as serial RAMs, LCDs, tone generators, other microcontrollers, etc.

4. General description

A typical I²C-bus configuration is shown in [Figure 10–16](#). Depending on the state of the direction bit (R/W), two types of data transfers are possible on the I²C-bus:

- Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte.
- Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows the data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a “not acknowledge” is returned. The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended

with a STOP condition or with a Repeated START condition. Since a Repeated START condition is also the beginning of the next serial transfer, the I²C bus will not be released.

The I²C interface is byte oriented and has four operating modes: master transmitter mode, master receiver mode, slave transmitter mode and slave receiver mode.

The I²C interface complies with the entire I²C specification, supporting the ability to turn power off to the ARM Cortex-M0 without interfering with other devices on the same I²C-bus.

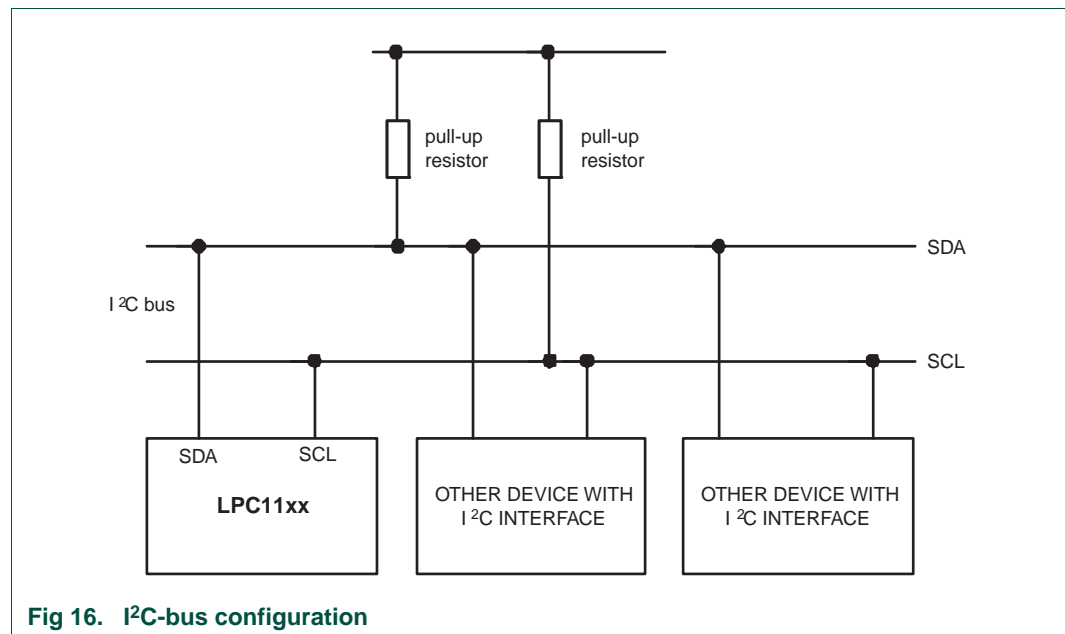


Fig 16. I²C-bus configuration

4.1 I²C Fast-mode Plus

Fast-Mode Plus supports a 1 Mbit/sec transfer rate to communicate with the I²C-bus products which NXP Semiconductors is now providing.

In order to use Fast-Mode Plus, the I²C pins must be properly configured in the IOCONFIG register block, see [Table 7-73](#) and [Table 7-74](#). In Fast-mode Plus, rates above 400 kHz and up to 1 MHz may be selected.

5. Pin description

Table 137. I²C-bus pin description

Pin	Type	Description
SDA	Input/Output	I ² C Serial Data
SCL	Input/Output	I ² C Serial Clock

The I²C-bus pins must be configured through the IOCON_PIO0_4 ([Table 7-73](#)) and IOCON_PIO0_5 ([Table 7-74](#)) registers for Standard/ Fast-mode or Fast-mode Plus. In these modes, the I²C-bus pins are open-drain outputs and fully compatible with the I²C-bus specification.

6. Clocking and power control

The clock to the I²C-bus interface (PCLK_I2C) is provided by the system clock (see [Figure 3–3](#)). This clock can be disabled through bit 5 in the AHBCLKCTRL register ([Section 3–4.14](#)) for power savings.

7. Register description

Table 138. Register overview: I²C (base address 0x4000 0000)

Name	Access	Address offset	Description	Reset value ^[1]
I2CONSET	R/W	0x000	I²C Control Set Register. When a one is written to a bit of this register, the corresponding bit in the I ² C control register is set. Writing a zero has no effect on the corresponding bit in the I ² C control register.	0x00
I2STAT	RO	0x004	I²C Status Register. During I ² C operation, this register provides detailed status codes that allow software to determine the next action needed.	0xF8
I2DAT	R/W	0x008	I²C Data Register. During master or slave transmit mode, data to be transmitted is written to this register. During master or slave receive mode, data that has been received may be read from this register.	0x00
I2ADR0	R/W	0x00C	I²C Slave Address Register 0. Contains the 7-bit slave address for operation of the I ² C interface in slave mode, and is not used in master mode. The least significant bit determines whether a slave responds to the General Call address.	0x00
I2SCLH	R/W	0x010	SCH Duty Cycle Register High Half Word. Determines the high time of the I ² C clock.	0x04
I2SCLL	R/W	0x014	SCL Duty Cycle Register Low Half Word. Determines the low time of the I ² C clock. I2nSCLL and I2nSCLH together determine the clock frequency generated by an I ² C master and certain times used in slave mode.	0x04
I2CONCLR	WO	0x018	I²C Control Clear Register. When a one is written to a bit of this register, the corresponding bit in the I ² C control register is cleared. Writing a zero has no effect on the corresponding bit in the I ² C control register.	NA
I2MMCTRL	R/W	0x01C	Monitor mode control register.	0x00
I2ADR1	R/W	0x020	I²C Slave Address Register 1. Contains the 7-bit slave address for operation of the I ² C interface in slave mode, and is not used in master mode. The least significant bit determines whether a slave responds to the General Call address.	0x00
I2ADR2	R/W	0x024	I²C Slave Address Register 2. Contains the 7-bit slave address for operation of the I ² C interface in slave mode, and is not used in master mode. The least significant bit determines whether a slave responds to the General Call address.	0x00
I2ADR3	R/W	0x028	I²C Slave Address Register 3. Contains the 7-bit slave address for operation of the I ² C interface in slave mode, and is not used in master mode. The least significant bit determines whether a slave responds to the General Call address.	0x00
I2DATA_BUFFER	RO	0x02C	Data buffer register. The contents of the 8 MSBs of the I2DAT shift register will be transferred to the DATA_BUFFER automatically after every nine bits (8 bits of data plus ACK or NACK) has been received on the bus.	0x00

Table 138. Register overview: I²C (base address 0x4000 0000) ...continued

Name	Access	Address offset	Description	Reset value ^[1]
I2MASK0	R/W	0x030	I²C Slave address mask register 0. This mask register is associated with I2ADR0 to determine an address match. The mask register has no effect when comparing to the General Call address ('0000000').	0x00
I2MASK1	R/W	0x034	I²C Slave address mask register 1. This mask register is associated with I2ADR0 to determine an address match. The mask register has no effect when comparing to the General Call address ('0000000').	0x00
I2MASK2	R/W	0x038	I²C Slave address mask register 2. This mask register is associated with I2ADR0 to determine an address match. The mask register has no effect when comparing to the General Call address ('0000000').	0x00
I2MASK3	R/W	0x03C	I²C Slave address mask register 3. This mask register is associated with I2ADR0 to determine an address match. The mask register has no effect when comparing to the General Call address ('0000000').	0x00

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

7.1 I²C Control Set register (I2CONSET - 0x4000 0000)

The I2CONSET registers control setting of bits in the I2CON register that controls operation of the I²C interface. Writing a one to a bit of this register causes the corresponding bit in the I²C control register to be set. Writing a zero has no effect.

Table 139. I²C Control Set register (I2CONSET - address 0x4000 0000) bit description

Bit	Symbol	Description	Reset value
1:0	-	Reserved. User software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
2	AA	Assert acknowledge flag.	
3	SI	I ² C interrupt flag.	0
4	STO	STOP flag.	0
5	STA	START flag.	0
6	I2EN	I ² C interface enable.	0
7	-	Reserved. User software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

I2EN I²C Interface Enable. When I2EN is 1, the I²C interface is enabled. I2EN can be cleared by writing 1 to the I2ENC bit in the I2CONCLR register. When I2EN is 0, the I²C interface is disabled.

When I2EN is "0", the SDA and SCL input signals are ignored, the I²C block is in the "not addressed" slave state, and the STO bit is forced to "0".

I2EN should not be used to temporarily release the I²C-bus since, when I2EN is reset, the I²C-bus status is lost. The AA flag should be used instead.

STA is the START flag. Setting this bit causes the I²C interface to enter master mode and transmit a START condition or transmit a Repeated START condition if it is already in master mode.

When STA is 1 and the I²C interface is not already in master mode, it enters master mode, checks the bus and generates a START condition if the bus is free. If the bus is not free, it waits for a STOP condition (which will free the bus) and generates a START condition after a delay of a half clock period of the internal clock generator. If the I²C interface is already in master mode and data has been transmitted or received, it transmits a Repeated START condition. STA may be set at any time, including when the I²C interface is in an addressed slave mode.

STA can be cleared by writing 1 to the STAC bit in the I2CONCLR register. When STA is 0, no START condition or Repeated START condition will be generated.

If STA and STO are both set, then a STOP condition is transmitted on the I²C-bus if it the interface is in master mode, and transmits a START condition thereafter. If the I²C interface is in slave mode, an internal STOP condition is generated, but is not transmitted on the bus.

STO is the STOP flag. Setting this bit causes the I²C interface to transmit a STOP condition in master mode, or recover from an error condition in slave mode. When STO is 1 in master mode, a STOP condition is transmitted on the I²C-bus. When the bus detects the STOP condition, STO is cleared automatically.

In slave mode, setting this bit can recover from an error condition. In this case, no STOP condition is transmitted to the bus. The hardware behaves as if a STOP condition has been received and it switches to “not addressed” slave receiver mode. The STO flag is cleared by hardware automatically.

SI is the I²C Interrupt Flag. This bit is set when the I²C state changes. However, entering state F8 does not set SI since there is nothing for an interrupt service routine to do in that case.

While SI is set, the low period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. When SCL is HIGH, it is unaffected by the state of the SI flag. SI must be reset by software, by writing a 1 to the SIC bit in I2CONCLR register.

AA is the Assert Acknowledge Flag. When set to 1, an acknowledge (low level to SDA) will be returned during the acknowledge clock pulse on the SCL line on the following situations:

1. The address in the Slave Address Register has been received.
2. The General Call address has been received while the General Call bit (GC) in I2ADR is set.
3. A data byte has been received while the I²C is in the master receiver mode.
4. A data byte has been received while the I²C is in the addressed slave receiver mode

The AA bit can be cleared by writing 1 to the AAC bit in the I2CONCLR register. When AA is 0, a not acknowledge (HIGH level to SDA) will be returned during the acknowledge clock pulse on the SCL line on the following situations:

1. A data byte has been received while the I²C is in the master receiver mode.
2. A data byte has been received while the I²C is in the addressed slave receiver mode.

7.2 I2C Status register (I2STAT - 0x4000 0004)

Each I2C Status register reflects the condition of the corresponding I2C interface. The I2C Status register is Read-Only.

Table 140. I2C Status register (I2STAT - 0x4000 0004) bit description

Bit	Symbol	Description	Reset value
2:0	-	These bits are unused and are always 0.	0
7:3	Status	These bits give the actual status information about the I2C interface.	0x1F

The three least significant bits are always 0. Taken as a byte, the status register contents represent a status code. There are 26 possible status codes. When the status code is 0xF8, there is no relevant information available and the SI bit is not set. All other 25 status codes correspond to defined I2C states. When any of these states entered, the SI bit will be set. For a complete list of status codes, refer to tables from [Table 10–157](#) to [Table 10–160](#).

7.3 I2C Data register (I2DAT - 0x4000 0008)

This register contains the data to be transmitted or the data just received. The CPU can read and write to this register only while it is not in the process of shifting a byte, when the SI bit is set. Data in I2DAT remains stable as long as the SI bit is set. Data in I2DAT is always shifted from right to left: the first bit to be transmitted is the MSB (bit 7), and after a byte has been received, the first bit of received data is located at the MSB of I2DAT.

Table 141. I2C Data register (I2DAT - 0x4000 0008) bit description

Bit	Symbol	Description	Reset value
7:0	Data	This register holds data values that have been received or are to be transmitted.	0

7.4 I2C Slave Address register (I2ADR0- 0x4000 000C)

These registers are readable and writable and are only used when an I2C interface is set to slave mode. In master mode, this register has no effect. The LSB of I2ADR is the General Call bit. When this bit is set, the General Call address (0x00) is recognized.

Any of these registers which contain the bit 00x will be disabled and will not match any address on the bus. All four registers will be cleared to this disabled state on reset.

Table 142. I2C Slave Address registers (I2ADR0 - 0x4000 000C) bit description

Bit	Symbol	Description	Reset value
0	GC	General Call enable bit.	0
7:1	Address	The I2C device address for slave mode.	0x00

7.5 I2C SCL HIGH duty cycle register and LOW duty cycle register (I2SCLH - 0x4000 0010 and I2SCLL - 0x4000 0014)

Table 143. I2C SCL HIGH duty cycle register (I2SCLH - address 0x4000 0010) bit description

Bit	Symbol	Description	Reset value
15:0	SCLH	Count for SCL HIGH time period selection.	0x0004

Table 144. I²C SCL Low duty cycle register (I2SCLL - 0x4000 0014) bit description

Bit	Symbol	Description	Reset value
15:0	SCLL	Count for SCL low time period selection.	0x0004

7.5.1 Selecting the appropriate I²C data rate and duty cycle

Software must set values for the registers I2SCLH and I2SCLL to select the appropriate data rate and duty cycle. I2SCLH defines the number of PCLK_I2C cycles for the SCL HIGH time, I2SCLL defines the number of PCLK_I2C cycles for the SCL low time. The frequency is determined by the following formula (PCLK_I2C is the frequency of the system clock):

(8)

$$I^2C_{bitfrequency} = \frac{PCLK_{I2C}}{I2SCLH + I2SCLL}$$

The values for I2SCLL and I2SCLH must ensure that the data rate is in the appropriate I²C data rate range. Each register value must be greater than or equal to 4. [Table 12–192](#) gives some examples of I²C-bus rates based on PCLK_I2C frequency and I2SCLL and I2SCLH values.

Table 145. I2SCLL + I2SCLH values for selected I²C clock values

I ² C mode	I ² C bit frequency	PCLK_I2C (MHz)								
		6	8	10	12	16	20	30	40	50
		I2SCLH + I2SCLL								
Standard mode	100 kHz	60	80	100	120	160	200	300	400	500
Fast-mode	400 kHz	15	20	25	30	40	50	75	100	125
Fast-mode Plus	1 MHz	-	8	10	12	16	20	30	40	50

I2SCLL and I2SCLH values should not necessarily be the same. Software can set different duty cycles on SCL by setting these two registers. For example, the I²C-bus specification defines the SCL low time and high time at different values for a Fast-mode and Fast-mode Plus I²C.

7.6 I²C Control Clear register (I2CONCLR - 0x4000 0018)

The I2CONCLR registers control clearing of bits in the I2CON register that controls operation of the I²C interface. Writing a one to a bit of this register causes the corresponding bit in the I²C control register to be cleared. Writing a zero has no effect.

Table 146. I²C Control Clear register (I2CONCLR - 0x4000 0018) bit description

Bit	Symbol	Description	Reset value
1:0	-	Reserved. User software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
2	AAC	Assert acknowledge Clear bit.	
3	SIC	I ² C interrupt Clear bit.	0

Table 146. I²C Control Clear register (I2CONCLR - 0x4000 0018) bit description

Bit	Symbol	Description	Reset value
4	-	Reserved. User software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
5	STAC	START flag Clear bit.	0
6	I2ENC	I ² C interface Disable bit.	0
7	-	Reserved. User software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

AAC is the Assert Acknowledge Clear bit. Writing a 1 to this bit clears the AA bit in the I2CONSET register. Writing 0 has no effect.

SIC is the I²C Interrupt Clear bit. Writing a 1 to this bit clears the SI bit in the I2CONSET register. Writing 0 has no effect.

STAC is the START flag Clear bit. Writing a 1 to this bit clears the STA bit in the I2CONSET register. Writing 0 has no effect.

I2ENC is the I²C Interface Disable bit. Writing a 1 to this bit clears the I2EN bit in the I2CONSET register. Writing 0 has no effect.

7.7 I²C Monitor mode control register (I2CMMCTRL0 - 0x4000 001C)

This register controls the Monitor mode which allows the I²C module to monitor traffic on the I²C bus without actually participating in traffic or interfering with the I²C bus.

Table 147. I²C Monitor mode control register (I2CMMCTRL0 - 0x4000 001C) bit description

Bit	Symbol	Value	Description	Reset value
0	MM_ENA		Monitor mode enable.	0
		0	Monitor mode disabled.	
		1	The I ² C module will enter monitor mode. In this mode the SDA output will be forced high. This will prevent the I ² C module from outputting data of any kind (including ACK) onto the I ² C data bus. Depending on the state of the ENA_SCL bit, the output may be also forced high, preventing the module from having control over the I ² C clock line.	
1	ENA_SCL		SCL output enable.	0
		0	When this bit is cleared to '0', the SCL output will be forced high when the module is in monitor mode. As described above, this will prevent the module from having any control over the I ² C clock line.	
		1	When this bit is set, the I ² C module may exercise the same control over the clock line that it would in normal operation. This means that, acting as a slave peripheral, the I ² C module can "stretch" the clock line (hold it low) until it has had time to respond to an I ² C interrupt. [1]	

Table 147. I²C Monitor mode control register (I2CMMCTRL0 - 0x4000 001C) bit description

Bit	Symbol	Value	Description	Reset value
3	MATCH_ALL		Select interrupt register match.	0
		0	When this bit is cleared, an interrupt will only be generated when a match occurs to one of the (up-to) four address registers described above. That is, the module will respond as a normal slave as far as address-recognition is concerned.	
		1	When this bit is set to '1' and the I ² C is in monitor mode, an interrupt will be generated on ANY address received. This will enable the part to monitor all traffic on the bus.	

[1] When the ENA_SCL bit is cleared and the I²C no longer has the ability to stall the bus, interrupt response time becomes important. To give the part more time to respond to an I²C interrupt under these conditions, a DATA_BUFFER register is used ([Section 10-7.9](#)) to hold received data for a full 9-bit word transmission time.

Remark: The ENA_SCL and MATCH_ALL bits have no effect if the MM_ENA is '0' (i.e. if the module is NOT in monitor mode).

7.7.1 Interrupt in Monitor mode

All interrupts will occur as normal when the module is in monitor mode. This means that the first interrupt will occur when an address-match is detected (any address received if the MATCH_ALL bit is set, otherwise an address matching one of the four address registers).

Subsequent to an address-match detection, interrupts will be generated after each data byte is received for a slave-write transfer, or after each byte that the module “thinks” it has transmitted for a slave-read transfer. In this second case, the data register will actually contain data transmitted by some other slave on the bus which was actually addressed by the master.

Following all of these interrupts, the processor may read the data register to see what was actually transmitted on the bus.

7.7.2 Loss of arbitration in Monitor mode

In monitor mode, the I²C module will not be able to respond to a request for information by the bus master or issue an ACK). Some other slave on the bus will respond instead. This will most probably result in a lost-arbitration state as far as our module is concerned.

Software should be aware of the fact that the module is in monitor mode and should not respond to any loss of arbitration state that is detected. In addition, hardware may be designed into the module to block some/all loss of arbitration states from occurring if those state would either prevent a desired interrupt from occurring or cause an unwanted interrupt to occur. Whether any such hardware will be added is still to be determined.

7.8 I²C Slave Address registers (I2ADR[0, 1, 2, 3]- 0x4000 00[0C, 20, 24, 28])

These registers are readable and writable and are only used when an I²C interface is set to slave mode. In master mode, this register has no effect. The LSB of I2ADR is the General Call bit. When this bit is set, the General Call address (0x00) is recognized.

Any of these registers which contain the bit 00x will be disabled and will not match any address on the bus. All four registers will be cleared to this disabled state on reset.

Table 148. I²C Slave Address registers (I2ADR[0, 1, 2, 3]- 0x4000 00[0C, 20, 24, 28]) bit description

Bit	Symbol	Description	Reset value
0	GC	General Call enable bit.	0
7:1	Address	The I ² C device address for slave mode.	0x00

7.9 I²C Data buffer register (I2CDATA_BUFFER - 0x4000 002C)

In monitor mode, the I²C module may lose the ability to stretch the clock (stall the bus) if the ENA_SCL bit is not set. This means that the processor will have a limited amount of time to read the contents of the data received on the bus. If the processor reads the I2DAT shift register, as it ordinarily would, it could have only one bit time to respond to the interrupt before the received data is overwritten by new data.

To give the processor more time to respond, a new 8-bit, read-only DATA_BUFFER register will be added. The contents of the 8 MSBs of the I2DAT shift register will be transferred to the DATA_BUFFER automatically after every nine bits (8 bits of data plus ACK or NACK) has been received on the bus. This means that the processor will have nine bit transmission times to respond to the interrupt and read the data before it is overwritten.

The processor will still have the ability to read I2DAT directly, as usual, and the behavior of I2DAT will not be altered in any way.

Although the DATA_BUFFER register is primarily intended for use in monitor mode with the ENA_SCL bit = '0', it will be available for reading at any time under any mode of operation.

Table 149. I²C Data buffer register (I2CDATA_BUFFER - 0x4000 002C) bit description

Bit	Symbol	Description	Reset value
7:0	Data	This register holds contents of the 8 MSBs of the I2DAT shift register.	0

7.10 I²C Mask registers (I2MASK[0, 1, 2, 3] - 0x4000 00[30, 34, 38, 3C])

The four mask registers each contain seven active bits (7:1). Any bit in these registers which is set to '1' will cause an automatic compare on the corresponding bit of the received address when it is compared to the I2ADDRn register associated with that mask register. In other words, bits in an I2ADDRn register which are masked are not taken into account in determining an address match.

On reset, all mask register bits are cleared to '0'.

The mask register has no effect on comparison to the General Call address ("0000000").

Bits(31:8) and bit(0) of the mask registers are unused and should not be written to. These bits will always read back as zeros.

When an address-match interrupt occurs, the processor will have to read the data register (I2DAT) to determine what the received address was that actually caused the match.

Table 150. I²C Mask registers (I2MASK[0, 1, 2, 3] - 0x4000 00[30, 34, 38, 3C]) bit description

Bit	Symbol	Description	Reset value
0	-	Reserved. User software should not write ones to reserved bits. This bit reads always back as 0.	0
7:1	MASK	Mask bits.	0x00
31:8	-	Reserved. User software should not write ones to reserved bits. These bits read always back as 0's.	0

8. I²C operating modes

In a given application, the I²C block may operate as a master, a slave, or both. In the slave mode, the I²C hardware looks for any one of its four slave addresses and the General Call address. If one of these addresses is detected, an interrupt is requested. If the processor wishes to become the bus master, the hardware waits until the bus is free before the master mode is entered so that a possible slave operation is not interrupted. If bus arbitration is lost in the master mode, the I²C block switches to the slave mode immediately and can detect its own slave address in the same serial transfer.

8.1 Master Transmitter mode

In this mode data is transmitted from master to slave. Before the master transmitter mode can be entered, the I2CONSET register must be initialized as shown in [Table 10–151](#). I2EN must be set to 1 to enable the I²C function. If the AA bit is 0, the I²C interface will not acknowledge any address when another device is master of the bus, so it can not enter slave mode. The STA, STO and SI bits must be 0. The SI Bit is cleared by writing 1 to the SIC bit in the I2CONCLR register. The STA bit should be cleared after writing the slave address.

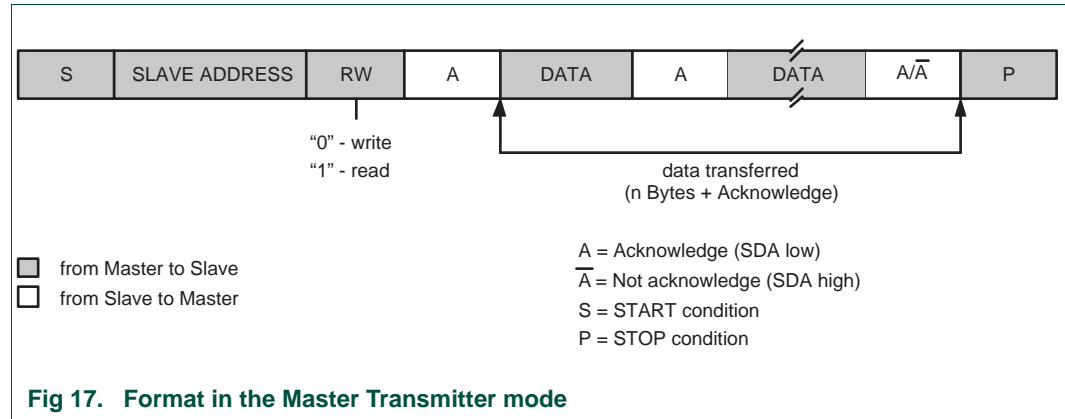
Table 151. I2CONSET used to configure Master mode

Bit	7	6	5	4	3	2	1	0
Symbol	-	I2EN	STA	STO	SI	AA	-	-
Value	-	1	0	0	0	0	-	-

The first byte transmitted contains the slave address of the receiving device (7 bits) and the data direction bit. In this mode the data direction bit (R/W) should be 0 which means Write. The first byte transmitted contains the slave address and Write bit. Data is transmitted 8 bits at a time. After each byte is transmitted, an acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

The I²C interface will enter master transmitter mode when software sets the STA bit. The I²C logic will send the START condition as soon as the bus is free. After the START condition is transmitted, the SI bit is set, and the status code in the I2STAT register is 0x08. This status code is used to vector to a state service routine which will load the slave address and Write bit to the I2DAT register, and then clear the SI bit. SI is cleared by writing a 1 to the SIC bit in the I2CONCLR register.

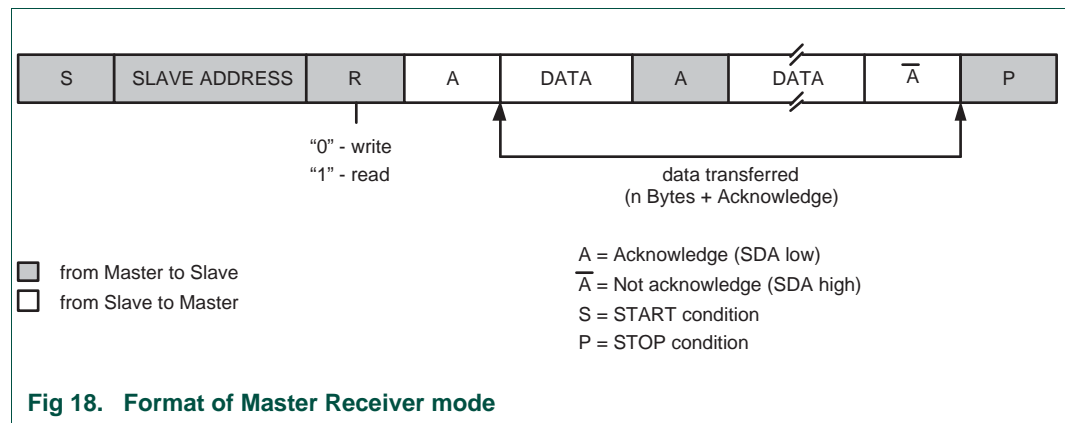
When the slave address and R/W bit have been transmitted and an acknowledgment bit has been received, the SI bit is set again, and the possible status codes now are 0x18, 0x20, or 0x38 for the master mode, or 0x68, 0x78, or 0xB0 if the slave mode was enabled (by setting AA to 1). The appropriate actions to be taken for each of these status codes are shown in [Table 10–157](#) to [Table 10–160](#).



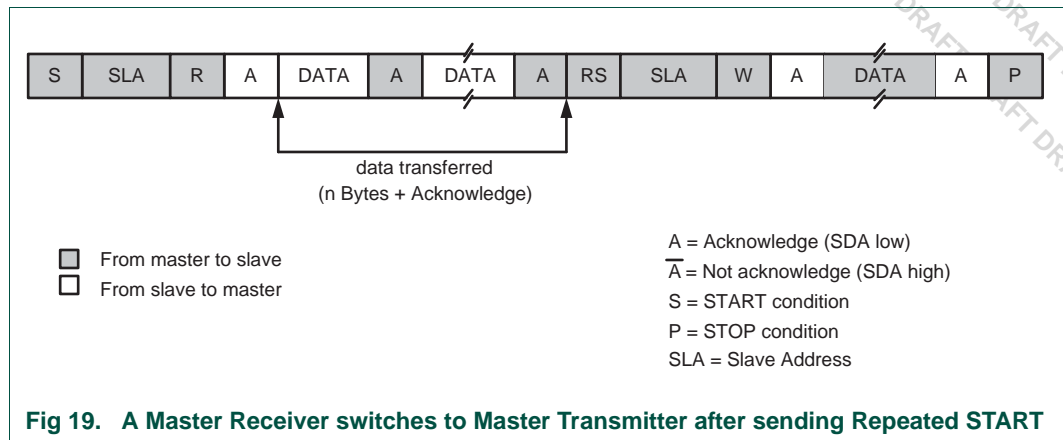
8.2 Master Receiver mode

In the master receiver mode, data is received from a slave transmitter. The transfer is initiated in the same way as in the master transmitter mode. When the START condition has been transmitted, the interrupt service routine must load the slave address and the data direction bit to the I²C Data register (I2DAT), and then clear the SI bit. In this case, the data direction bit (R/W) should be 1 to indicate a read.

When the slave address and data direction bit have been transmitted and an acknowledge bit has been received, the SI bit is set, and the Status Register will show the status code. For master mode, the possible status codes are 0x40, 0x48, or 0x38. For slave mode, the possible status codes are 0x68, 0x78, or 0xB0. For details, refer to [Table 10–158](#).



After a Repeated START condition, I²C may switch to the master transmitter mode.



8.3 Slave Receiver mode

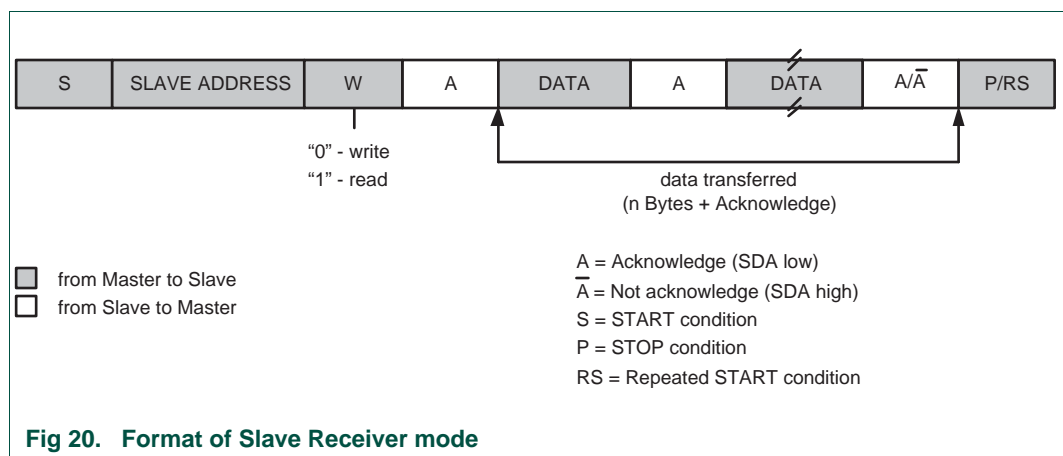
In the slave receiver mode, data bytes are received from a master transmitter. To initialize the slave receiver mode, write any of the Slave Address registers (I2ADR0-3) and write the I²C Control Set register (I2CONSET) as shown in [Table 10–152](#).

Table 152. I2CONSET used to configure Slave mode

Bit	7	6	5	4	3	2	1	0
Symbol	-	I2EN	STA	STO	SI	AA	-	-
Value	-	1	0	0	0	1	-	-

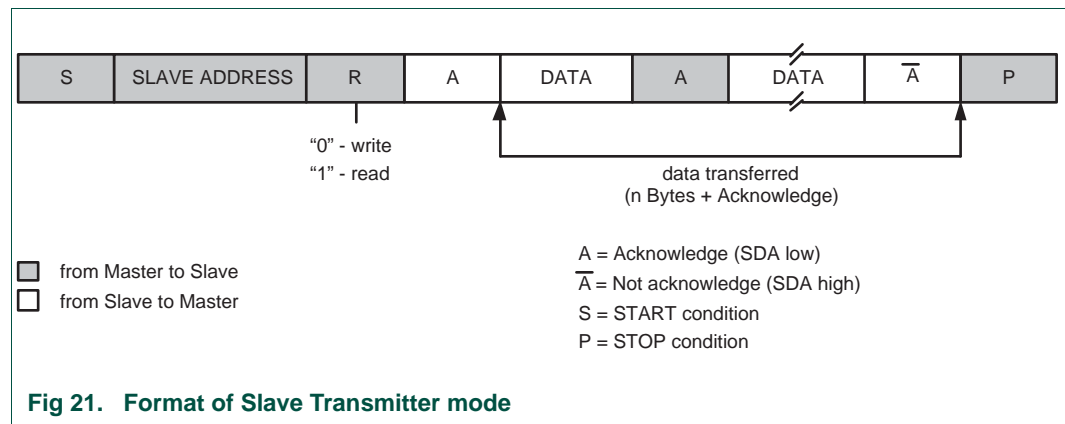
I2EN must be set to 1 to enable the I²C function. AA bit must be set to 1 to acknowledge its own slave address or the General Call address. The STA, STO and SI bits are set to 0.

After I2ADR and I2CONSET are initialized, the I²C interface waits until it is addressed by its own address or general address followed by the data direction bit. If the direction bit is 0 (W), it enters slave receiver mode. If the direction bit is 1 (R), it enters slave transmitter mode. After the address and direction bit have been received, the SI bit is set and a valid status code can be read from the Status register (I2STAT). Refer to [Table 10–159](#) for the status codes and actions.



8.4 Slave Transmitter mode

The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will be 1, indicating a read operation. Serial data is transmitted via SDA while the serial clock is input through SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer. In a given application, I²C may operate as a master and as a slave. In the slave mode, the I²C hardware looks for its own slave address and the General Call address. If one of these addresses is detected, an interrupt is requested. When the microcontroller wishes to become the bus master, the hardware waits until the bus is free before the master mode is entered so that a possible slave action is not interrupted. If bus arbitration is lost in the master mode, the I²C interface switches to the slave mode immediately and can detect its own slave address in the same serial transfer.



9. Functional description

[Figure 10–22](#) shows how the on-chip I²C-bus interface is implemented, and the following text describes the individual blocks.

9.1 Input filters and output stages

Input signals are synchronized with the internal clock, and spikes shorter than three clocks are filtered out.

The output for I²C-bus is a special pad designed to conform to the I²C-bus specification.

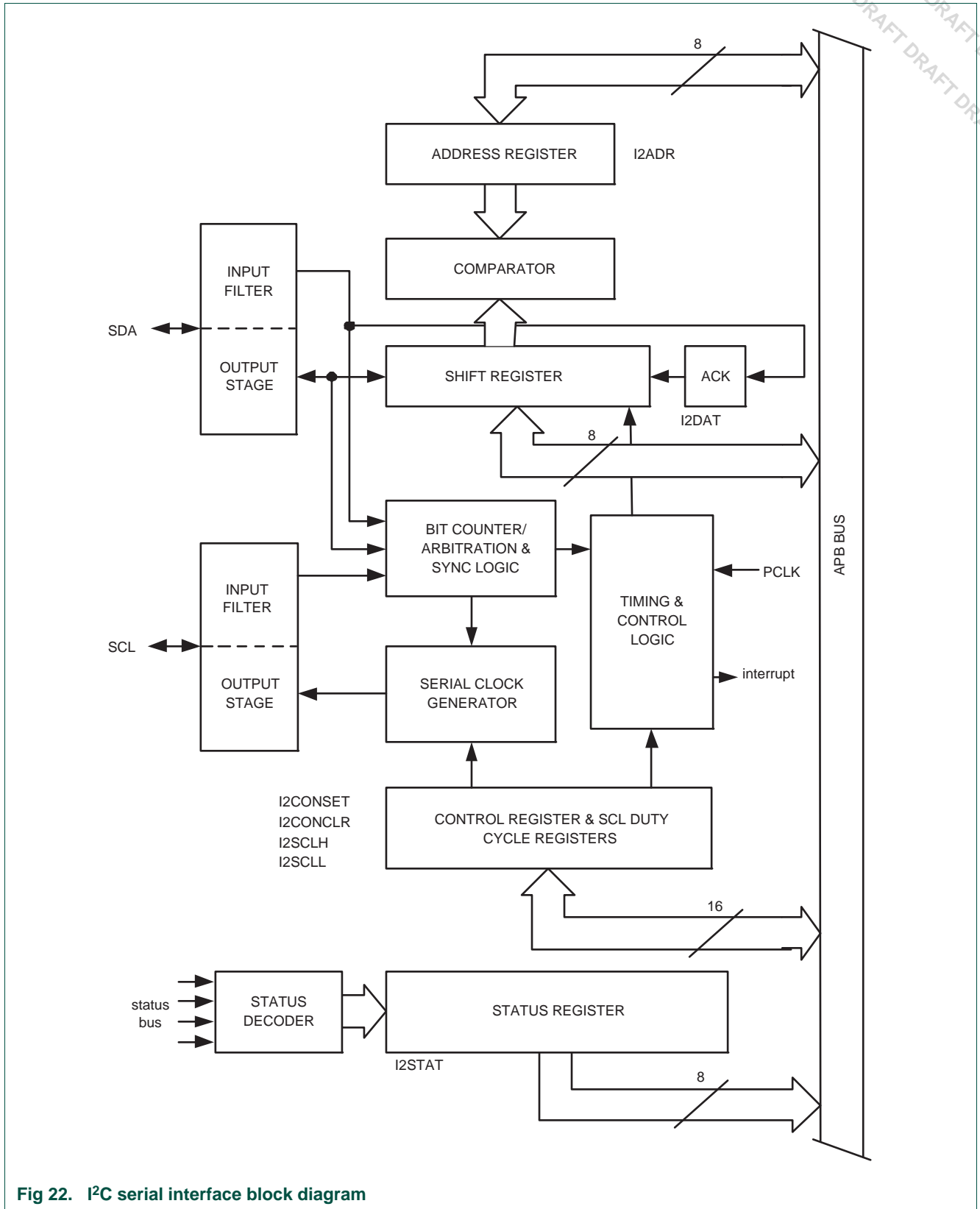


Fig 22. I2C serial interface block diagram

9.2 Address Registers, I2ADDR0 to I2ADDR3

These registers may be loaded with the 7-bit slave address (7 most significant bits) to which the I²C block will respond when programmed as a slave transmitter or receiver. The LSB (GC) is used to enable General Call address (0x00) recognition. When multiple slave addresses are enabled, the actual address received may be read from the I2DAT register at the state where the own slave address has been received.

9.3 Address mask registers, I2MASK0 to I2MASK3

The four mask registers each contain seven active bits (7:1). Any bit in these registers which is set to '1' will cause an automatic compare on the corresponding bit of the received address when it is compared to the I2ADDRn register associated with that mask register. In other words, bits in an I2ADDRn register which are masked are not taken into account in determining an address match.

If the I2ADDRn bit 0 (GC enable bit) is as set and bits(7:1) are all zeroes, then the part will respond to a received address = "000000" regardless of the state of the associated mask register.

When an address-match interrupt occurs, the processor will have to read the data register (I2DAT) to determine what the received address was that actually caused the match.

9.4 Comparator

The comparator compares the received 7-bit slave address with its own slave address (7 most significant bits in I2ADR). It also compares the first received 8-bit byte with the General Call address (0x00). If an equality is found, the appropriate status bits are set and an interrupt is requested.

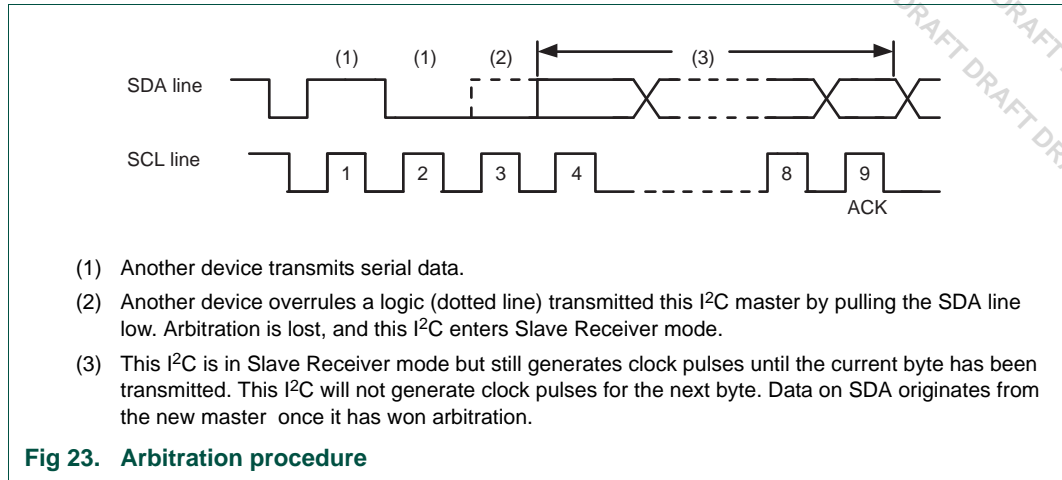
9.5 Shift register, I2DAT

This 8-bit register contains a byte of serial data to be transmitted or a byte which has just been received. Data in I2DAT is always shifted from right to left; the first bit to be transmitted is the MSB (bit 7) and, after a byte has been received, the first bit of received data is located at the MSB of I2DAT. While data is being shifted out, data on the bus is simultaneously being shifted in; I2DAT always contains the last byte present on the bus. Thus, in the event of lost arbitration, the transition from master transmitter to slave receiver is made with the correct data in I2DAT.

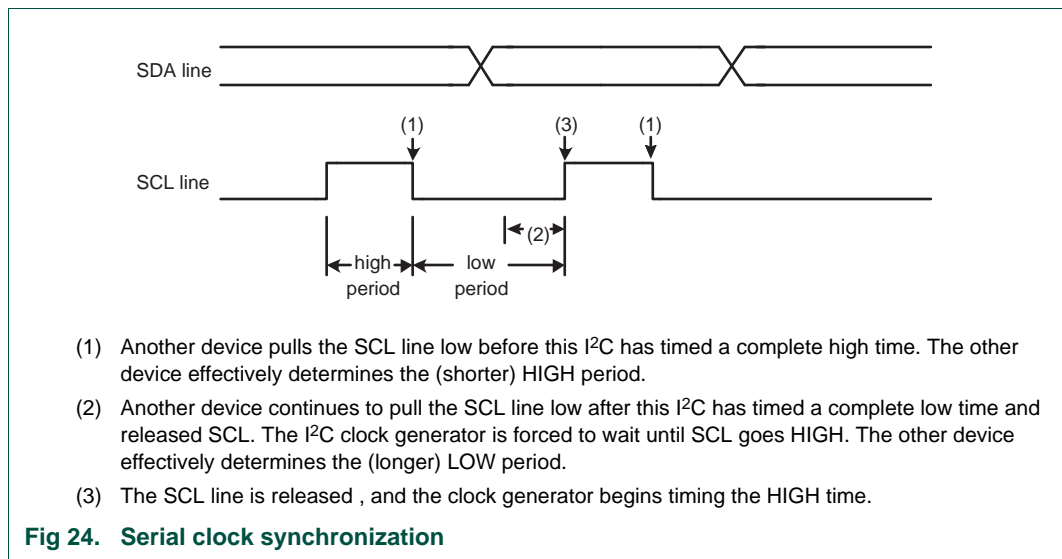
9.6 Arbitration and synchronization logic

In the master transmitter mode, the arbitration logic checks that every transmitted logic 1 actually appears as a logic 1 on the I²C-bus. If another device on the bus overrules a logic 1 and pulls the SDA line low, arbitration is lost, and the I²C block immediately changes from master transmitter to slave receiver. The I²C block will continue to output clock pulses (on SCL) until transmission of the current serial byte is complete.

Arbitration may also be lost in the master receiver mode. Loss of arbitration in this mode can only occur while the I²C block is returning a "not acknowledge: (logic 1) to the bus. Arbitration is lost when another device on the bus pulls this signal low. Since this can occur only at the end of a serial byte, the I²C block generates no further clock pulses. [Figure 10–23](#) shows the arbitration procedure.



The synchronization logic will synchronize the serial clock generator with the clock pulses on the SCL line from another device. If two or more master devices generate clock pulses, the “mark” duration is determined by the device that generates the shortest “marks,” and the “space” duration is determined by the device that generates the longest “spaces”. [Figure 10–24](#) shows the synchronization procedure.



A slave may stretch the space duration to slow down the bus master. The space duration may also be stretched for handshaking purposes. This can be done after each bit or after a complete byte transfer. the I²C block will stretch the SCL space duration after a byte has been transmitted or received and the acknowledge bit has been transferred. The serial interrupt flag (SI) is set, and the stretching continues until the serial interrupt flag is cleared.

9.7 Serial clock generator

This programmable clock pulse generator provides the SCL clock pulses when the I²C block is in the master transmitter or master receiver mode. It is switched off when the I²C block is in a slave mode. The I²C output clock frequency and duty cycle is programmable

via the I²C Clock Control Registers. See the description of the I2CSCLL and I2CSCLH registers for details. The output clock pulses have a duty cycle as programmed unless the bus is synchronizing with other SCL clock sources as described above.

9.8 Timing and control

The timing and control logic generates the timing and control signals for serial byte handling. This logic block provides the shift pulses for I2DAT, enables the comparator, generates and detects START and STOP conditions, receives and transmits acknowledge bits, controls the master and slave modes, contains interrupt request logic, and monitors the I²C-bus status.

9.9 Control register, I2CONSET and I2CONCLR

The I²C control register contains bits used to control the following I²C block functions: start and restart of a serial transfer, termination of a serial transfer, bit rate, address recognition, and acknowledgment.

The contents of the I²C control register may be read as I2CONSET. Writing to I2CONSET will set bits in the I²C control register that correspond to ones in the value written. Conversely, writing to I2CONCLR will clear bits in the I²C control register that correspond to ones in the value written.

9.10 Status decoder and status register

The status decoder takes all of the internal status bits and compresses them into a 5-bit code. This code is unique for each I²C-bus status. The 5-bit code may be used to generate vector addresses for fast processing of the various service routines. Each service routine processes a particular bus status. There are 26 possible bus states if all four modes of the I²C block are used. The 5-bit status code is latched into the five most significant bits of the status register when the serial interrupt flag is set (by hardware) and remains stable until the interrupt flag is cleared by software. The three least significant bits of the status register are always zero. If the status code is used as a vector to service routines, then the routines are displaced by eight address locations. Eight bytes of code is sufficient for most of the service routines (see the software example in this section).

10. Details of I²C operating modes

The four operating modes are:

- Master Transmitter
- Master Receiver
- Slave Receiver
- Slave Transmitter

Data transfers in each mode of operation are shown in [Figure 10–25](#), [Figure 10–26](#), [Figure 10–27](#), [Figure 10–28](#), and [Figure 10–29](#). [Table 10–153](#) lists abbreviations used in these figures when describing the I²C operating modes.

Table 153. Abbreviations used to describe an I²C operation

Abbreviation	Explanation
S	START Condition
SLA	7-bit slave address
R	Read bit (HIGH level at SDA)
W	Write bit (LOW level at SDA)
A	Acknowledge bit (LOW level at SDA)
\bar{A}	Not acknowledge bit (HIGH level at SDA)
Data	8-bit data byte
P	STOP condition

In [Figure 10–25](#) to [Figure 10–29](#), circles are used to indicate when the serial interrupt flag is set. The numbers in the circles show the status code held in the I2STAT register. At these points, a service routine must be executed to continue or complete the serial transfer. These service routines are not critical since the serial transfer is suspended until the serial interrupt flag is cleared by software.

When a serial interrupt routine is entered, the status code in I2STAT is used to branch to the appropriate service routine. For each status code, the required software action and details of the following serial transfer are given in tables from [Table 10–157](#) to [Table 10–161](#).

10.1 Master Transmitter mode

In the master transmitter mode, a number of data bytes are transmitted to a slave receiver (see [Figure 10–25](#)). Before the master transmitter mode can be entered, I2CON must be initialized as follows:

Table 154. I2CONSET used to initialize Master Transmitter mode

Bit	7	6	5	4	3	2	1	0
Symbol	-	I2EN	STA	STO	SI	AA	-	-
Value	-	1	0	0	0	x	-	-

The I²C rate must also be configured in the I2SCLL and I2SCLH registers. I2EN must be set to logic 1 to enable the I²C block. If the AA bit is reset, the I²C block will not acknowledge its own slave address or the General Call address in the event of another device becoming master of the bus. In other words, if AA is reset, the I²C interface cannot enter a slave mode. STA, STO, and SI must be reset.

The master transmitter mode may now be entered by setting the STA bit. The I²C logic will now test the I²C-bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the serial interrupt flag (SI) is set, and the status code in the status register (I2STAT) will be 0x08. This status code is used by the interrupt service routine to enter the appropriate state service routine that loads I2DAT with the slave address and the data direction bit (SLA+W). The SI bit in I2CON must then be reset before the serial transfer can continue.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in I2STAT are possible. There are 0x18, 0x20, or 0x38 for the master mode and also 0x68, 0x78, or 0xB0 if the slave mode was enabled (AA = logic 1).

The appropriate action to be taken for each of these status codes is detailed in [Table 10–157](#). After a Repeated START condition (state 0x10). The I2C block may switch to the master receiver mode by loading I2DAT with SLA+R).

10.2 Master Receiver mode

In the master receiver mode, a number of data bytes are received from a slave transmitter (see [Figure 10–26](#)). The transfer is initialized as in the master transmitter mode. When the START condition has been transmitted, the interrupt service routine must load I2DAT with the 7-bit slave address and the data direction bit (SLA+R). The SI bit in I2CON must then be cleared before the serial transfer can continue.

When the slave address and the data direction bit have been transmitted and an acknowledgment bit has been received, the serial interrupt flag (SI) is set again, and a number of status codes in I2STAT are possible. These are 0x40, 0x48, or 0x38 for the master mode and also 0x68, 0x78, or 0xB0 if the slave mode was enabled (AA = 1). The appropriate action to be taken for each of these status codes is detailed in [Table 10–158](#). After a Repeated START condition (state 0x10), the I2C block may switch to the master transmitter mode by loading I2DAT with SLA+W.

10.3 Slave Receiver mode

In the slave receiver mode, a number of data bytes are received from a master transmitter (see [Figure 10–27](#)). To initiate the slave receiver mode, I2ADR and I2CON must be loaded as follows:

Table 155. I2ADR usage in Slave Receiver mode

Bit	7	6	5	4	3	2	1	0
Symbol	own slave 7-bit address						GC	

The upper 7 bits are the address to which the I2C block will respond when addressed by a master. If the LSB (GC) is set, the I2C block will respond to the General Call address (0x00); otherwise it ignores the General Call address.

Table 156. I2CONSET used to initialize Slave Receiver mode

Bit	7	6	5	4	3	2	1	0
Symbol	-	I2EN	STA	STO	SI	AA	-	-
Value	-	1	0	0	0	1	-	-

The I2C-bus rate settings do not affect the I2C block in the slave mode. I2EN must be set to logic 1 to enable the I2C block. The AA bit must be set to enable the I2C block to acknowledge its own slave address or the General Call address. STA, STO, and SI must be reset.

When I2ADR and I2CON have been initialized, the I2C block waits until it is addressed by its own slave address followed by the data direction bit which must be “0” (W) for the I2C block to operate in the slave receiver mode. After its own slave address and the W bit have been received, the serial interrupt flag (SI) is set and a valid status code can be read from I2STAT. This status code is used to vector to a state service routine. The appropriate action to be taken for each of these status codes is detailed in [Table 10–159](#). The slave receiver mode may also be entered if arbitration is lost while the I2C block is in the master mode (see status 0x68 and 0x78).

If the AA bit is reset during a transfer, the I²C block will return a not acknowledge (logic 1) to SDA after the next received data byte. While AA is reset, the I²C block does not respond to its own slave address or a General Call address. However, the I²C-bus is still monitored and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate the I²C block from the I²C-bus.

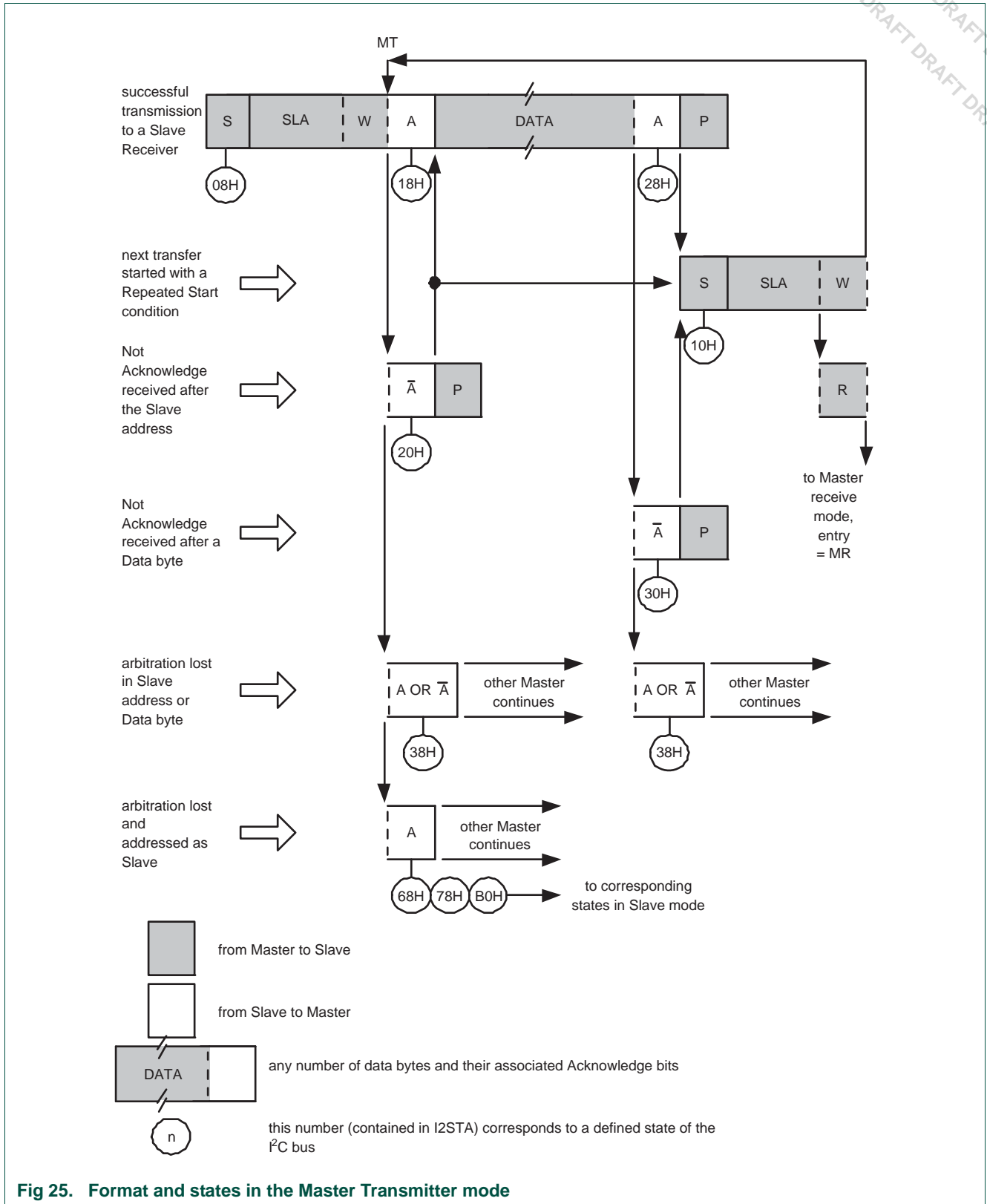


Fig 25. Format and states in the Master Transmitter mode

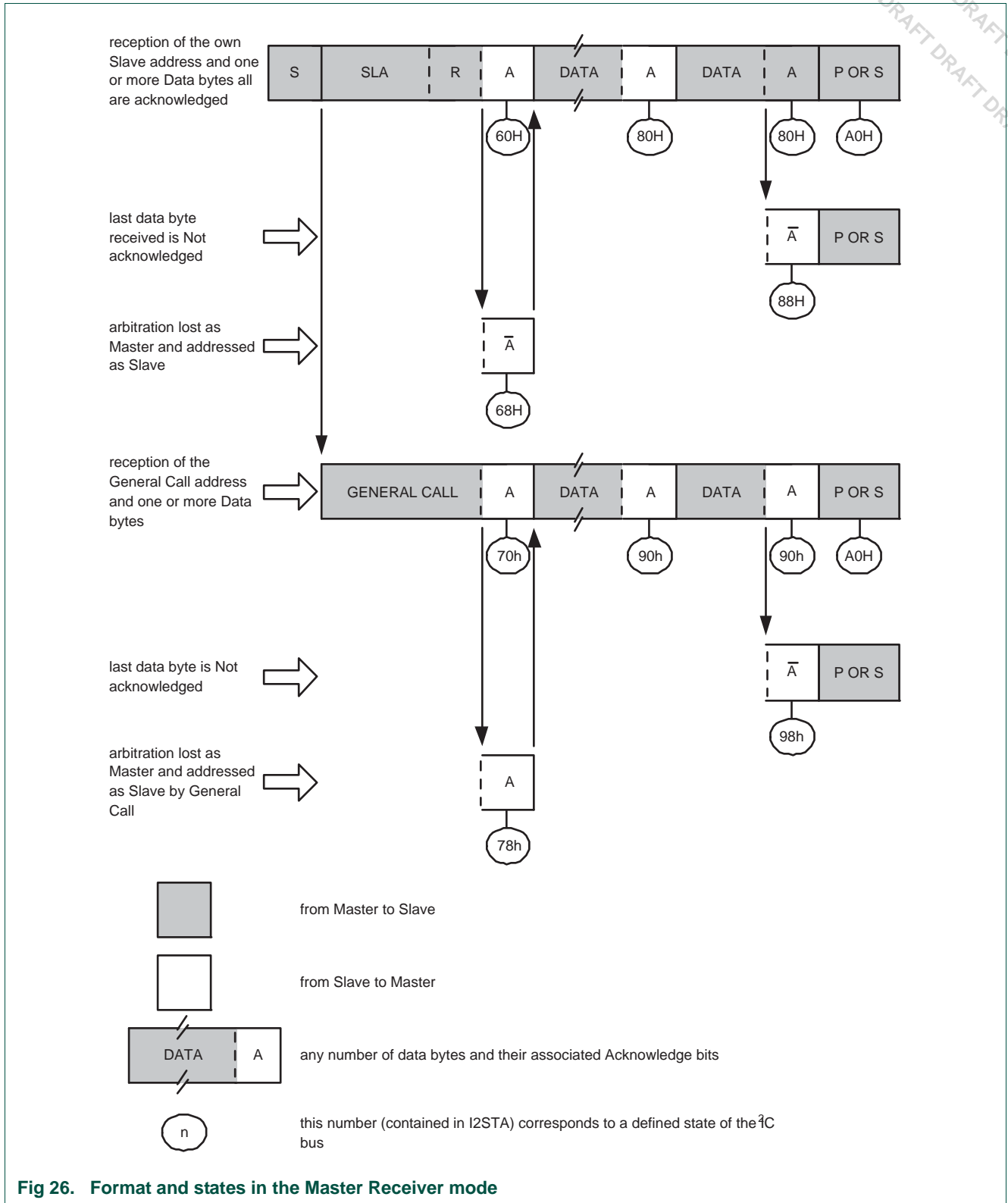


Fig 26. Format and states in the Master Receiver mode

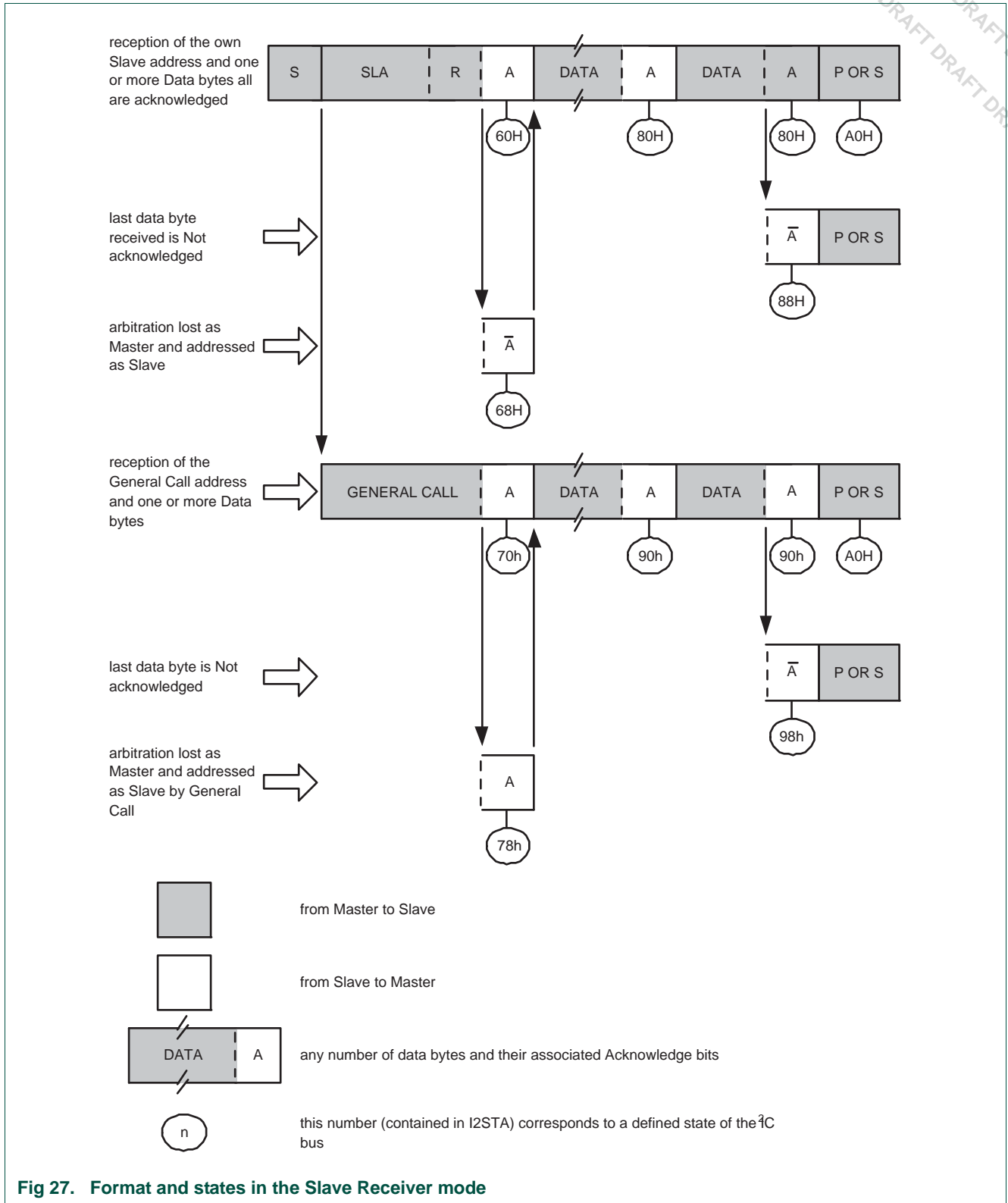
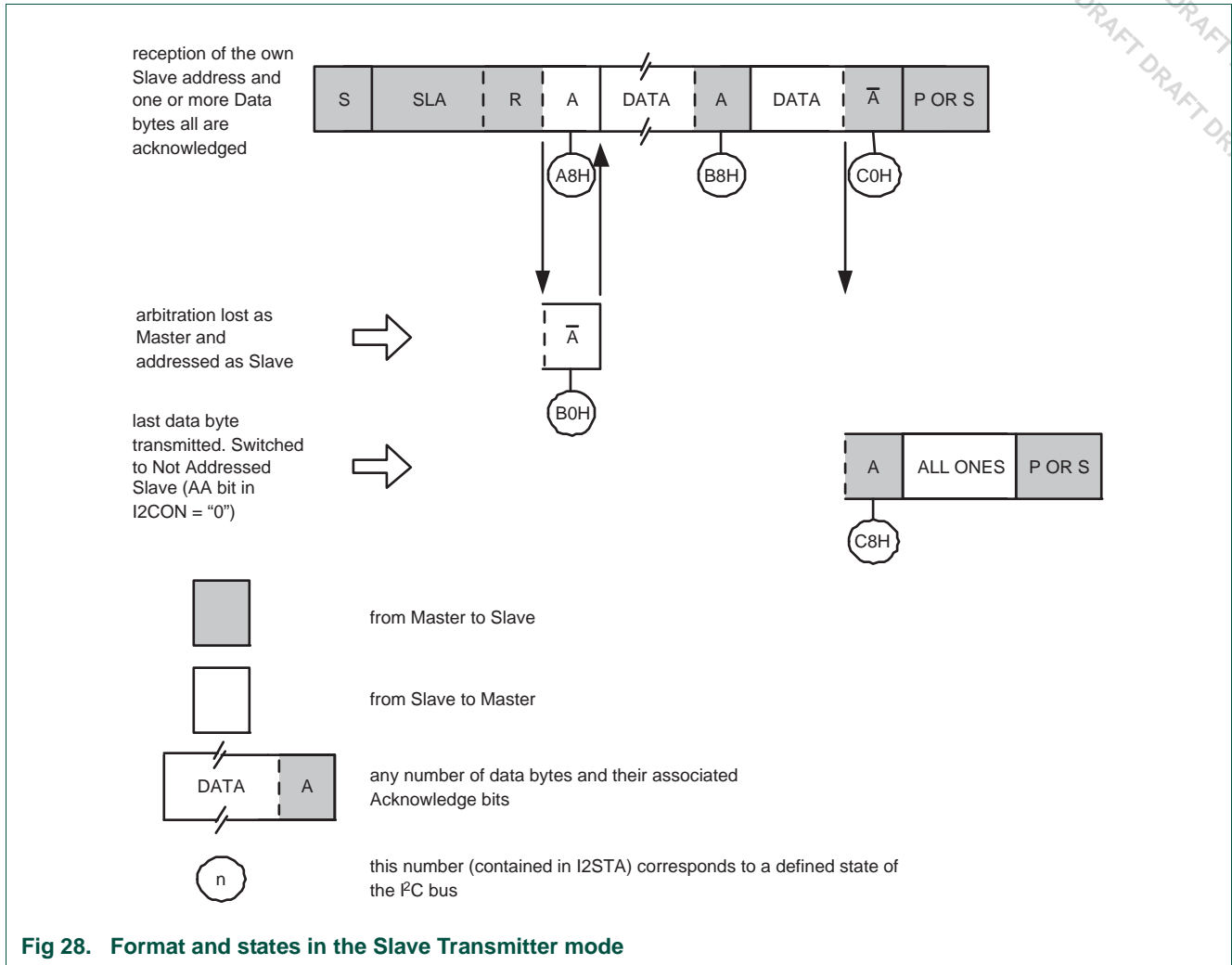


Fig 27. Format and states in the Slave Receiver mode



10.4 Slave Transmitter mode

In the slave transmitter mode, a number of data bytes are transmitted to a master receiver (see [Figure 10-28](#)). Data transfer is initialized as in the slave receiver mode. When I2ADR and I2CON have been initialized, the I²C block waits until it is addressed by its own slave address followed by the data direction bit which must be "1" (R) for the I²C block to operate in the slave transmitter mode. After its own slave address and the R bit have been received, the serial interrupt flag (SI) is set and a valid status code can be read from I2STAT. This status code is used to vector to a state service routine, and the appropriate action to be taken for each of these status codes is detailed in [Table 10-160](#). The slave transmitter mode may also be entered if arbitration is lost while the I²C block is in the master mode (see state 0xB0).

If the AA bit is reset during a transfer, the I²C block will transmit the last byte of the transfer and enter state 0xC0 or 0xC8. The I²C block is switched to the not addressed slave mode and will ignore the master receiver if it continues the transfer. Thus the master receiver receives all 1s as serial data. While AA is reset, the I²C block does not respond to its own slave address or a General Call address. However, the I²C-bus is still monitored, and address recognition may be resumed at any time by setting AA. This means that the AA bit may be used to temporarily isolate the I²C block from the I²C-bus.

Table 157. Master Transmitter mode

Status Code (I2STAT)	Status of the I ² C-bus and hardware	Application software response					Next action taken by I ² C hardware
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	
0x08	A START condition has been transmitted.	Load SLA+W; clear STA	X	0	0	X	SLA+W will be transmitted; ACK bit will be received.
0x10	A Repeated START condition has been transmitted.	Load SLA+W or	X	0	0	X	As above.
		Load SLA+R; Clear STA	X	0	0	X	SLA+W will be transmitted; the I ² C block will be switched to MST/REC mode.
0x18	SLA+W has been transmitted; ACK has been received.	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received.
		No I2DAT action or	1	0	0	X	Repeated START will be transmitted.
		No I2DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
		No I2DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
0x20	SLA+W has been transmitted; NOT ACK has been received.	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received.
		No I2DAT action or	1	0	0	X	Repeated START will be transmitted.
		No I2DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
		No I2DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
0x28	Data byte in I2DAT has been transmitted; ACK has been received.	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received.
		No I2DAT action or	1	0	0	X	Repeated START will be transmitted.
		No I2DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
		No I2DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
0x30	Data byte in I2DAT has been transmitted; NOT ACK has been received.	Load data byte or	0	0	0	X	Data byte will be transmitted; ACK bit will be received.
		No I2DAT action or	1	0	0	X	Repeated START will be transmitted.
		No I2DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
		No I2DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
0x38	Arbitration lost in SLA+R/W or Data bytes.	No I2DAT action or	0	0	0	X	I ² C-bus will be released; not addressed slave will be entered.
		No I2DAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free.

Table 158. Master Receiver mode

Status Code (I2CSTAT)	Status of the I ² C-bus and hardware	Application software response To/From I2DAT	To I2CON				Next action taken by I ² C hardware
			STA	STO	SI	AA	
0x08	A START condition has been transmitted.	Load SLA+R	X	0	0	X	SLA+R will be transmitted; ACK bit will be received.
0x10	A Repeated START condition has been transmitted.	Load SLA+R or	X	0	0	X	As above.
		Load SLA+W	X	0	0	X	SLA+W will be transmitted; the I ² C block will be switched to MST/TRX mode.
0x38	Arbitration lost in NOT ACK bit.	No I2DAT action or	0	0	0	X	I ² C-bus will be released; the I ² C block will enter a slave mode.
		No I2DAT action	1	0	0	X	A START condition will be transmitted when the bus becomes free.
0x40	SLA+R has been transmitted; ACK has been received.	No I2DAT action or	0	0	0	0	Data byte will be received; NOT ACK bit will be returned.
		No I2DAT action	0	0	0	1	Data byte will be received; ACK bit will be returned.
0x48	SLA+R has been transmitted; NOT ACK has been received.	No I2DAT action or	1	0	0	X	Repeated START condition will be transmitted.
		No I2DAT action or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
		No I2DAT action	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.
0x50	Data byte has been received; ACK has been returned.	Read data byte or	0	0	0	0	Data byte will be received; NOT ACK bit will be returned.
		Read data byte	0	0	0	1	Data byte will be received; ACK bit will be returned.
0x58	Data byte has been received; NOT ACK has been returned.	Read data byte or	1	0	0	X	Repeated START condition will be transmitted.
		Read data byte or	0	1	0	X	STOP condition will be transmitted; STO flag will be reset.
		Read data byte	1	1	0	X	STOP condition followed by a START condition will be transmitted; STO flag will be reset.

Table 159. Slave Receiver mode

Status Code (I2CSTAT)	Status of the I2C-bus and hardware	Application software response					Next action taken by I2C hardware
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	
0x60	Own SLA+W has been received; ACK has been returned.	No I2DAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned.
		No I2DAT action	X	0	0	1	Data byte will be received and ACK will be returned.
0x68	Arbitration lost in SLA+R/W as master; Own SLA+W has been received, ACK returned.	No I2DAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned.
		No I2DAT action	X	0	0	1	Data byte will be received and ACK will be returned.
0x70	General call address (0x00) has been received; ACK has been returned.	No I2DAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned.
		No I2DAT action	X	0	0	1	Data byte will be received and ACK will be returned.
0x78	Arbitration lost in SLA+R/W as master; General call address has been received, ACK has been returned.	No I2DAT action or	X	0	0	0	Data byte will be received and NOT ACK will be returned.
		No I2DAT action	X	0	0	1	Data byte will be received and ACK will be returned.
0x80	Previously addressed with own SLV address; DATA has been received; ACK has been returned.	Read data byte or	X	0	0	0	Data byte will be received and NOT ACK will be returned.
		Read data byte	X	0	0	1	Data byte will be received and ACK will be returned.
0x88	Previously addressed with own SLA; DATA byte has been received; NOT ACK has been returned.	Read data byte or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address.
		Read data byte or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2ADR[0] = logic 1.
		Read data byte or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free.
		Read data byte	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2ADR[0] = logic 1. A START condition will be transmitted when the bus becomes free.
0x90	Previously addressed with General Call; DATA byte has been received; ACK has been returned.	Read data byte or	X	0	0	0	Data byte will be received and NOT ACK will be returned.
		Read data byte	X	0	0	1	Data byte will be received and ACK will be returned.

Table 159. Slave Receiver mode

Status Code (I2CSTAT)	Status of the I ² C-bus and hardware	Application software response To/From I2DAT	To I2CON				Next action taken by I ² C hardware
			STA	STO	SI	AA	
0x98	Previously addressed with General Call; DATA byte has been received; NOT ACK has been returned.	Read data byte or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address.
		Read data byte or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2ADR[0] = logic 1.
		Read data byte or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free.
		Read data byte	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2ADR[0] = logic 1. A START condition will be transmitted when the bus becomes free.
0xA0	A STOP condition or Repeated START condition has been received while still addressed as SLV/REC or SLV/TRX.	No STDAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address.
		No STDAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2ADR[0] = logic 1.
		No STDAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free.
		No STDAT action	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2ADR[0] = logic 1. A START condition will be transmitted when the bus becomes free.

Table 160. Slave Transmitter mode

Status Code (I2CSTAT)	Status of the I2C-bus and hardware	Application software response				Next action taken by I2C hardware	
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	
0xA8	Own SLA+R has been received; ACK has been returned.	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received.
		Load data byte	X	0	0	1	Data byte will be transmitted; ACK will be received.
0xB0	Arbitration lost in SLA+R/W as master; Own SLA+R has been received, ACK has been returned.	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received.
		Load data byte	X	0	0	1	Data byte will be transmitted; ACK bit will be received.
0xB8	Data byte in I2DAT has been transmitted; ACK has been received.	Load data byte or	X	0	0	0	Last data byte will be transmitted and ACK bit will be received.
		Load data byte	X	0	0	1	Data byte will be transmitted; ACK bit will be received.
0xC0	Data byte in I2DAT has been transmitted; NOT ACK has been received.	No I2DAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address.
		No I2DAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2ADR[0] = logic 1.
		No I2DAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free.
		No I2DAT action	1	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2ADR[0] = logic 1. A START condition will be transmitted when the bus becomes free.
0xC8	Last data byte in I2DAT has been transmitted (AA = 0); ACK has been received.	No I2DAT action or	0	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address.
		No I2DAT action or	0	0	0	1	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2ADR[0] = logic 1.
		No I2DAT action or	1	0	0	0	Switched to not addressed SLV mode; no recognition of own SLA or General call address. A START condition will be transmitted when the bus becomes free.
		No I2DAT action	1	0	0	01	Switched to not addressed SLV mode; Own SLA will be recognized; General call address will be recognized if I2ADR.0 = logic 1. A START condition will be transmitted when the bus becomes free.

10.5 Miscellaneous states

There are two I2STAT codes that do not correspond to a defined I²C hardware state (see [Table 10–161](#)). These are discussed below.

10.6 I2STAT = 0xF8

This status code indicates that no relevant information is available because the serial interrupt flag, SI, is not yet set. This occurs between other states and when the I²C block is not involved in a serial transfer.

10.7 I2STAT = 0x00

This status code indicates that a bus error has occurred during an I²C serial transfer. A bus error is caused when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. A bus error may also be caused when external interference disturbs the internal I²C block signals. When a bus error occurs, SI is set. To recover from a bus error, the STO flag must be set and SI must be cleared. This causes the I²C block to enter the “not addressed” slave mode (a defined state) and to clear the STO flag (no other bits in I2CON are affected). The SDA and SCL lines are released (a STOP condition is not transmitted).

Table 161. Miscellaneous States

Status Code (I2CSTAT)	Status of the I ² C-bus and hardware	Application software response				Next action taken by I ² C hardware	
		To/From I2DAT	To I2CON				
			STA	STO	SI	AA	
0xF8	No relevant state information available; SI = 0.	No I2DAT action	No I2CON action				Wait or proceed current transfer.
0x00	Bus error during MST or selected slave modes, due to an illegal START or STOP condition. State 0x00 can also occur when interference causes the I ² C block to enter an undefined state.	No I2DAT action	0	1	0	X	Only the internal hardware is affected in the MST or addressed SLV modes. In all cases, the bus is released and the I ² C block is switched to the not addressed SLV mode. STO is reset.

10.8 Some special cases

The I²C hardware has facilities to handle the following special cases that may occur during a serial transfer:

10.9 Simultaneous Repeated START conditions from two masters

A Repeated START condition may be generated in the master transmitter or master receiver modes. A special case occurs if another master simultaneously generates a Repeated START condition (see [Figure 10–29](#)). Until this occurs, arbitration is not lost by either master since they were both transmitting the same data.

If the I²C hardware detects a Repeated START condition on the I²C-bus before generating a Repeated START condition itself, it will release the bus, and no interrupt request is generated. If another master frees the bus by generating a STOP condition, the I²C block will transmit a normal START condition (state 0x08), and a retry of the total serial data transfer can commence.

10.10 Data transfer after loss of arbitration

Arbitration may be lost in the master transmitter and master receiver modes (see [Figure 10–23](#)). Loss of arbitration is indicated by the following states in I2STAT; 0x38, 0x68, 0x78, and 0xB0 (see [Figure 10–25](#) and [Figure 10–26](#)).

If the STA flag in I2CON is set by the routines which service these states, then, if the bus is free again, a START condition (state 0x08) is transmitted without intervention by the CPU, and a retry of the total serial transfer can commence.

10.11 Forced access to the I²C-bus

In some applications, it may be possible for an uncontrolled source to cause a bus hang-up. In such situations, the problem may be caused by interference, temporary interruption of the bus or a temporary short-circuit between SDA and SCL.

If an uncontrolled source generates a superfluous START or masks a STOP condition, then the I²C-bus stays busy indefinitely. If the STA flag is set and bus access is not obtained within a reasonable amount of time, then a forced access to the I²C-bus is possible. This is achieved by setting the STO flag while the STA flag is still set. No STOP condition is transmitted. The I²C hardware behaves as if a STOP condition was received and is able to transmit a START condition. The STO flag is cleared by hardware (see [Figure 10–30](#)).

10.12 I²C-bus obstructed by a LOW level on SCL or SDA

An I²C-bus hang-up can occur if either the SDA or SCL line is held LOW by any device on the bus. If the SCL line is obstructed (pulled LOW) by a device on the bus, no further serial transfer is possible, and the problem must be resolved by the device that is pulling the SCL bus line LOW.

Typically, the SDA line may be obstructed by another device on the bus that has become out of synchronization with the current bus master by either missing a clock, or by sensing a noise pulse as a clock. In this case, the problem can be solved by transmitting additional clock pulses on the SCL line (see [Figure 10–31](#)). The I²C interface does not include a dedicated time-out timer to detect an obstructed bus, but this can be implemented using another timer in the system. When detected, software can force clocks (up to 9 may be required) on SCL until SDA is released by the offending device. At that point, the slave may still be out of synchronization, so a START should be generated to insure that all I²C peripherals are synchronized.

10.13 Bus error

A bus error occurs when a START or STOP condition is detected at an illegal position in the format frame. Examples of illegal positions are during the serial transfer of an address byte, a data bit, or an acknowledge bit.

The I²C hardware only reacts to a bus error when it is involved in a serial transfer either as a master or an addressed slave. When a bus error is detected, the I²C block immediately switches to the not addressed slave mode, releases the SDA and SCL lines, sets the interrupt flag, and loads the status register with 0x00. This status code may be used to vector to a state service routine which either attempts the aborted serial transfer again or simply recovers from the error condition as shown in [Table 10–161](#).

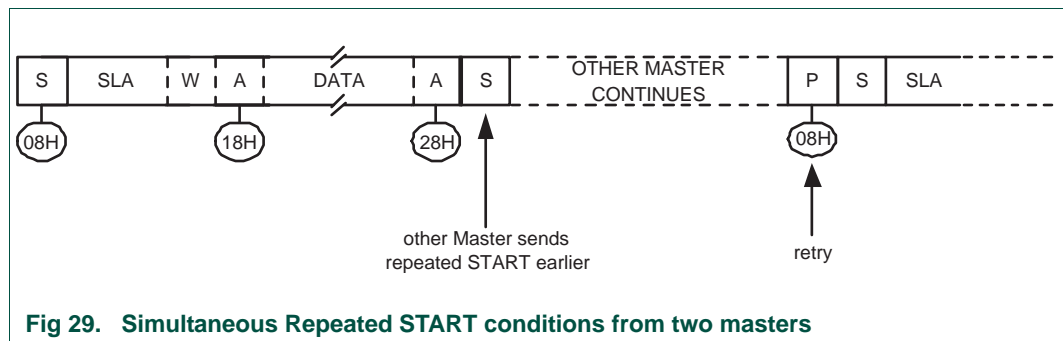


Fig 29. Simultaneous Repeated START conditions from two masters

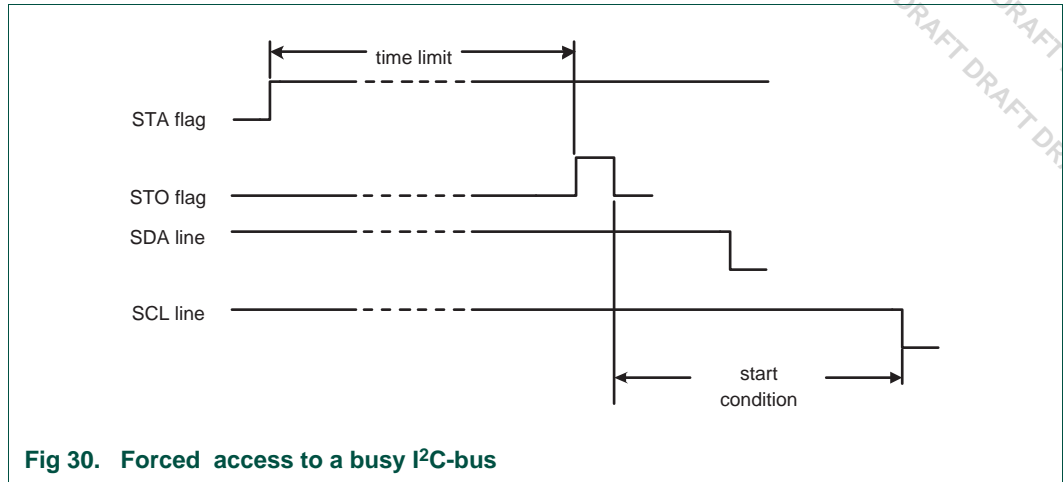


Fig 30. Forced access to a busy I2C-bus

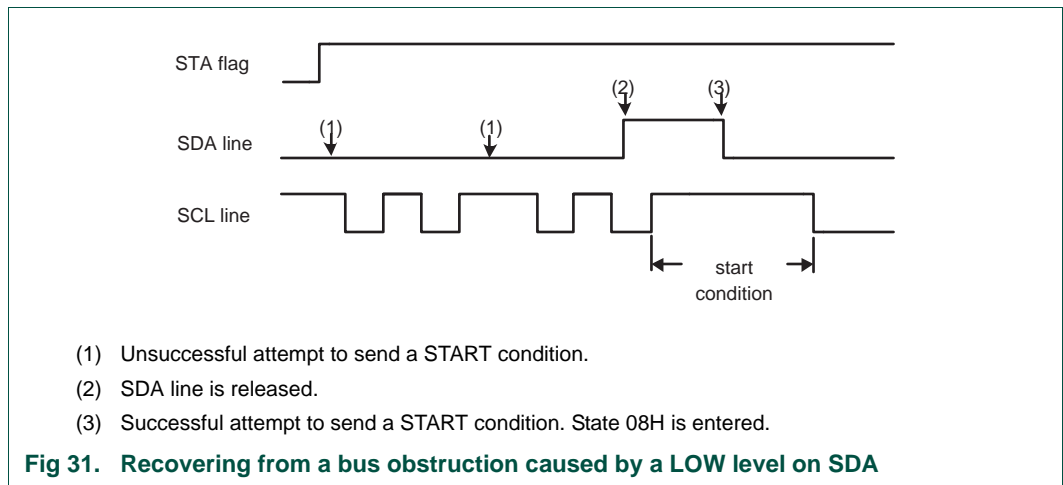


Fig 31. Recovering from a bus obstruction caused by a LOW level on SDA

10.14 I2C state service routines

This section provides examples of operations that must be performed by various I2C state service routines. This includes:

- Initialization of the I2C block after a Reset.
- I2C Interrupt Service
- The 26 state service routines providing support for all four I2C operating modes.

10.15 Initialization

In the initialization example, the I2C block is enabled for both master and slave modes. For each mode, a buffer is used for transmission and reception. The initialization routine performs the following functions:

- I2ADR is loaded with the part's own slave address and the General Call bit (GC)
- The I2C interrupt enable and interrupt priority bits are set

- The slave mode is enabled by simultaneously setting the I2EN and AA bits in I2CON and the serial clock frequency (for master modes) is defined by loading the I2SCLH and I2SCLL registers. The master routines must be started in the main program.

The I²C hardware now begins checking the I²C-bus for its own slave address and General Call. If the General Call or the own slave address is detected, an interrupt is requested and I2STAT is loaded with the appropriate state information.

10.16 I²C interrupt service

When the I²C interrupt is entered, I2STAT contains a status code which identifies one of the 26 state services to be executed.

10.17 The state service routines

Each state routine is part of the I²C interrupt routine and handles one of the 26 states.

10.18 Adapting state services to an application

The state service examples show the typical actions that must be performed in response to the 26 I²C state codes. If one or more of the four I²C operating modes are not used, the associated state services can be omitted, as long as care is taken that those states can never occur.

In an application, it may be desirable to implement some kind of time-out during I²C operations, in order to trap an inoperative bus or a lost service routine.

11. Software example

11.1 Initialization routine

Example to initialize I²C Interface as a Slave and/or Master.

1. Load I2ADR with own Slave Address, enable General Call recognition if needed.
2. Enable I²C interrupt.
3. Write 0x44 to I2CONSET to set the I2EN and AA bits, enabling Slave functions. For Master only functions, write 0x40 to I2CONSET.

11.2 Start Master Transmit function

Begin a Master Transmit operation by setting up the buffer, pointer, and data count, then initiating a START.

1. Initialize Master data counter.
2. Set up the Slave Address to which data will be transmitted, and add the Write bit.
3. Write 0x20 to I2CONSET to set the STA bit.
4. Set up data to be transmitted in Master Transmit buffer.
5. Initialize the Master data counter to match the length of the message being sent.
6. Exit

11.3 Start Master Receive function

Begin a Master Receive operation by setting up the buffer, pointer, and data count, then initiating a START.

1. Initialize Master data counter.
2. Set up the Slave Address to which data will be transmitted, and add the Read bit.
3. Write 0x20 to I2CONSET to set the STA bit.
4. Set up the Master Receive buffer.
5. Initialize the Master data counter to match the length of the message to be received.
6. Exit

11.4 I²C interrupt routine

Determine the I²C state and which state routine will be used to handle it.

1. Read the I²C status from I2STA.
2. Use the status value to branch to one of 26 possible state routines.

11.5 Non mode specific states

11.5.1 State: 0x00

Bus Error. Enter not addressed Slave mode and release bus.

1. Write 0x14 to I2CONSET to set the STO and AA bits.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Exit

11.5.2 Master States

State 08 and State 10 are for both Master Transmit and Master Receive modes. The R/W bit decides whether the next state is within Master Transmit mode or Master Receive mode.

11.5.3 State: 0x08

A START condition has been transmitted. The Slave Address + R/W bit will be transmitted, an ACK bit will be received.

1. Write Slave Address with R/W bit to I2DAT.
2. Write 0x04 to I2CONSET to set the AA bit.
3. Write 0x08 to I2CONCLR to clear the SI flag.
4. Set up Master Transmit mode data buffer.
5. Set up Master Receive mode data buffer.
6. Initialize Master data counter.
7. Exit

11.5.4 State: 0x10

A Repeated START condition has been transmitted. The Slave Address + R/W bit will be transmitted, an ACK bit will be received.

1. Write Slave Address with R/W bit to I2DAT.
2. Write 0x04 to I2CONSET to set the AA bit.
3. Write 0x08 to I2CONCLR to clear the SI flag.
4. Set up Master Transmit mode data buffer.
5. Set up Master Receive mode data buffer.
6. Initialize Master data counter.
7. Exit

11.6 Master Transmitter states

11.6.1 State: 0x18

Previous state was State 8 or State 10, Slave Address + Write has been transmitted, ACK has been received. The first data byte will be transmitted, an ACK bit will be received.

1. Load I2DAT with first data byte from Master Transmit buffer.
2. Write 0x04 to I2CONSET to set the AA bit.
3. Write 0x08 to I2CONCLR to clear the SI flag.
4. Increment Master Transmit buffer pointer.
5. Exit

11.6.2 State: 0x20

Slave Address + Write has been transmitted, NOT ACK has been received. A STOP condition will be transmitted.

1. Write 0x14 to I2CONSET to set the STO and AA bits.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Exit

11.6.3 State: 0x28

Data has been transmitted, ACK has been received. If the transmitted data was the last data byte then transmit a STOP condition, otherwise transmit the next data byte.

1. Decrement the Master data counter, skip to step 5 if not the last data byte.
2. Write 0x14 to I2CONSET to set the STO and AA bits.
3. Write 0x08 to I2CONCLR to clear the SI flag.
4. Exit
5. Load I2DAT with next data byte from Master Transmit buffer.
6. Write 0x04 to I2CONSET to set the AA bit.
7. Write 0x08 to I2CONCLR to clear the SI flag.
8. Increment Master Transmit buffer pointer

9. Exit

11.6.4 State: 0x30

Data has been transmitted, NOT ACK received. A STOP condition will be transmitted.

1. Write 0x14 to I2CONSET to set the STO and AA bits.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Exit

11.6.5 State: 0x38

Arbitration has been lost during Slave Address + Write or data. The bus has been released and not addressed Slave mode is entered. A new START condition will be transmitted when the bus is free again.

1. Write 0x24 to I2CONSET to set the STA and AA bits.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Exit

11.7 Master Receive states

11.7.1 State: 0x40

Previous state was State 08 or State 10. Slave Address + Read has been transmitted, ACK has been received. Data will be received and ACK returned.

1. Write 0x04 to I2CONSET to set the AA bit.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Exit

11.7.2 State: 0x48

Slave Address + Read has been transmitted, NOT ACK has been received. A STOP condition will be transmitted.

1. Write 0x14 to I2CONSET to set the STO and AA bits.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Exit

11.7.3 State: 0x50

Data has been received, ACK has been returned. Data will be read from I2DAT. Additional data will be received. If this is the last data byte then NOT ACK will be returned, otherwise ACK will be returned.

1. Read data byte from I2DAT into Master Receive buffer.
2. Decrement the Master data counter, skip to step 5 if not the last data byte.
3. Write 0x0C to I2CONCLR to clear the SI flag and the AA bit.
4. Exit
5. Write 0x04 to I2CONSET to set the AA bit.
6. Write 0x08 to I2CONCLR to clear the SI flag.

7. Increment Master Receive buffer pointer
8. Exit

11.7.4 State: 0x58

Data has been received, NOT ACK has been returned. Data will be read from I2DAT. A STOP condition will be transmitted.

1. Read data byte from I2DAT into Master Receive buffer.
2. Write 0x14 to I2CONSET to set the STO and AA bits.
3. Write 0x08 to I2CONCLR to clear the SI flag.
4. Exit

11.8 Slave Receiver states

11.8.1 State: 0x60

Own Slave Address + Write has been received, ACK has been returned. Data will be received and ACK returned.

1. Write 0x04 to I2CONSET to set the AA bit.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Set up Slave Receive mode data buffer.
4. Initialize Slave data counter.
5. Exit

11.8.2 State: 0x68

Arbitration has been lost in Slave Address and R/W bit as bus Master. Own Slave Address + Write has been received, ACK has been returned. Data will be received and ACK will be returned. STA is set to restart Master mode after the bus is free again.

1. Write 0x24 to I2CONSET to set the STA and AA bits.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Set up Slave Receive mode data buffer.
4. Initialize Slave data counter.
5. Exit.

11.8.3 State: 0x70

General call has been received, ACK has been returned. Data will be received and ACK returned.

1. Write 0x04 to I2CONSET to set the AA bit.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Set up Slave Receive mode data buffer.
4. Initialize Slave data counter.
5. Exit

11.8.4 State: 0x78

Arbitration has been lost in Slave Address + R/W bit as bus Master. General call has been received and ACK has been returned. Data will be received and ACK returned. STA is set to restart Master mode after the bus is free again.

1. Write 0x24 to I2CONSET to set the STA and AA bits.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Set up Slave Receive mode data buffer.
4. Initialize Slave data counter.
5. Exit

11.8.5 State: 0x80

Previously addressed with own Slave Address. Data has been received and ACK has been returned. Additional data will be read.

1. Read data byte from I2DAT into the Slave Receive buffer.
2. Decrement the Slave data counter, skip to step 5 if not the last data byte.
3. Write 0x0C to I2CONCLR to clear the SI flag and the AA bit.
4. Exit.
5. Write 0x04 to I2CONSET to set the AA bit.
6. Write 0x08 to I2CONCLR to clear the SI flag.
7. Increment Slave Receive buffer pointer.
8. Exit

11.8.6 State: 0x88

Previously addressed with own Slave Address. Data has been received and NOT ACK has been returned. Received data will not be saved. Not addressed Slave mode is entered.

1. Write 0x04 to I2CONSET to set the AA bit.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Exit

11.8.7 State: 0x90

Previously addressed with General Call. Data has been received, ACK has been returned. Received data will be saved. Only the first data byte will be received with ACK. Additional data will be received with NOT ACK.

1. Read data byte from I2DAT into the Slave Receive buffer.
2. Write 0x0C to I2CONCLR to clear the SI flag and the AA bit.
3. Exit

11.8.8 State: 0x98

Previously addressed with General Call. Data has been received, NOT ACK has been returned. Received data will not be saved. Not addressed Slave mode is entered.

1. Write 0x04 to I2CONSET to set the AA bit.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Exit

11.8.9 State: 0xA0

A STOP condition or Repeated START has been received, while still addressed as a Slave. Data will not be saved. Not addressed Slave mode is entered.

1. Write 0x04 to I2CONSET to set the AA bit.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Exit

11.9 Slave Transmitter states

11.9.1 State: 0xA8

Own Slave Address + Read has been received, ACK has been returned. Data will be transmitted, ACK bit will be received.

1. Load I2DAT from Slave Transmit buffer with first data byte.
2. Write 0x04 to I2CONSET to set the AA bit.
3. Write 0x08 to I2CONCLR to clear the SI flag.
4. Set up Slave Transmit mode data buffer.
5. Increment Slave Transmit buffer pointer.
6. Exit

11.9.2 State: 0xB0

Arbitration lost in Slave Address and R/W bit as bus Master. Own Slave Address + Read has been received, ACK has been returned. Data will be transmitted, ACK bit will be received. STA is set to restart Master mode after the bus is free again.

1. Load I2DAT from Slave Transmit buffer with first data byte.
2. Write 0x24 to I2CONSET to set the STA and AA bits.
3. Write 0x08 to I2CONCLR to clear the SI flag.
4. Set up Slave Transmit mode data buffer.
5. Increment Slave Transmit buffer pointer.
6. Exit

11.9.3 State: 0xB8

Data has been transmitted, ACK has been received. Data will be transmitted, ACK bit will be received.

1. Load I2DAT from Slave Transmit buffer with data byte.
2. Write 0x04 to I2CONSET to set the AA bit.
3. Write 0x08 to I2CONCLR to clear the SI flag.
4. Increment Slave Transmit buffer pointer.

5. Exit

11.9.4 State: 0xC0

Data has been transmitted, NOT ACK has been received. Not addressed Slave mode is entered.

1. Write 0x04 to I2CONSET to set the AA bit.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Exit.

11.9.5 State: 0xC8

The last data byte has been transmitted, ACK has been received. Not addressed Slave mode is entered.

1. Write 0x04 to I2CONSET to set the AA bit.
2. Write 0x08 to I2CONCLR to clear the SI flag.
3. Exit

4. Pin description

Table 162. SSP pin descriptions

Pin name	Type	Interface pin name/function			Pin description
		SPI	SSI	Microwire	
SCK0/1	I/O	SCK	CLK	SK	Serial Clock. SCK/CLK/SK is a clock signal used to synchronize the transfer of data. It is driven by the master and received by the slave. When SPI interface is used, the clock is programmable to be active-high or active-low, otherwise it is always active-high. SCK only switches during a data transfer. Any other time, the SSP interface either holds it in its inactive state or does not drive it (leaves it in high-impedance state).
SSEL0/1	I/O	SSEL	FS	CS	Frame Sync/Slave Select. When the SSP interface is a bus master, it drives this signal to an active state before the start of serial data and then releases it to an inactive state after the data has been sent. The active state of this signal can be high or low depending upon the selected bus and mode. When the SSP interface is a bus slave, this signal qualifies the presence of data from the Master according to the protocol in use. When there is just one bus master and one bus slave, the Frame Sync or Slave Select signal from the Master can be connected directly to the slave's corresponding input. When there is more than one slave on the bus, further qualification of their Frame Select/Slave Select inputs will typically be necessary to prevent more than one slave from responding to a transfer.
MISO0/1	I/O	MISO	DR(M) DX(S)	SI(M) SO(S)	Master In Slave Out. The MISO signal transfers serial data from the slave to the master. When the SSP is a slave, serial data is output on this signal. When the SSP0 is a master, it clocks in serial data from this signal. When the SSP is a slave and is not selected by FS/SSEL, it does not drive this signal (leaves it in high-impedance state).
MOSI0/1	I/O	MOSI	DX(M) DR(S)	SO(M) SI(S)	Master Out Slave In. The MOSI signal transfers serial data from the master to the slave. When the SSP is a master, it outputs serial data on this signal. When the SSP is a slave, it clocks in serial data from this signal.

Remark: The SCK0 function is multiplexed to three different pin locations (two locations on the HVQFN package). Use the IOCON_LOC register (see [Section 7-4.2](#)) to select a physical location for the SCK0 function in addition to selecting the function in the IOCON registers. The SCK1 pin is not multiplexed.

5. Clocking and power control

The SSP block is gated by the AHBCLKCTRL register (see [Table 3–19](#)). The peripheral SSP clock, which is used by the SSP clock divider and prescaler, is controlled by the SSP0/1CLKDIV registers (see [Section 3–4.15](#)).

The SSP0/1_PCLK clocks can be disabled in SSP0/1CLKDIV registers (see [Section 3–4.15](#)), and the SSP blocks can be disabled in the AHBCLKCTRL register ([Table 3–19](#)) for power savings.

6. Register description

The register addresses of the SSP controllers are shown in [Table 11–163](#) and [Table 11–164](#).

Table 163. Register overview: SSP0 (base address 0x4004 0000)

Name	Access	Address offset	Description	Reset Value ^[1]
SSP0CR0	R/W	0x000	Control Register 0. Selects the serial clock rate, bus type, and data size.	0
SSP0CR1	R/W	0x004	Control Register 1. Selects master/slave and other modes.	0
SSP0DR	R/W	0x008	Data Register. Writes fill the transmit FIFO, and reads empty the receive FIFO.	0
SSP0SR	RO	0x00C	Status Register	-
SSP0CPSR	R/W	0x010	Clock Prescale Register	0
SSP0IMSC	R/W	0x014	Interrupt Mask Set and Clear Register	0
SSP0RIS	R/W	0x018	Raw Interrupt Status Register	-
SSP0MIS	R/W	0x01C	Masked Interrupt Status Register	0
SSP0ICR	R/W	0x020	SSPICR Interrupt Clear Register	NA

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

Table 164. Register overview: SSP1 (base address 0x4005 8000)

Name	Access	Address offset	Description	Reset Value ^[1]
SSP1CR0	R/W	0x000	Control Register 0. Selects the serial clock rate, bus type, and data size.	0
SSP1CR1	R/W	0x004	Control Register 1. Selects master/slave and other modes.	0
SSP1DR	R/W	0x008	Data Register. Writes fill the transmit FIFO, and reads empty the receive FIFO.	0
SSP1SR	RO	0x00C	Status Register	-
SSP1CPSR	R/W	0x010	Clock Prescale Register	0
SSP1IMSC	R/W	0x014	Interrupt Mask Set and Clear Register	0
SSP1RIS	R/W	0x018	Raw Interrupt Status Register	-
SSP1MIS	R/W	0x01C	Masked Interrupt Status Register	0
SSP1ICR	R/W	0x020	SSPICR Interrupt Clear Register	NA

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

6.1 SSP Control Register 0

This register controls the basic operation of the SSP controller.

Table 165: SSP Control Register 0 (SSP0CR0 - address 0x4004 0000, SSP1CR0 - address 0x4005 8000) bit description

Bit	Symbol	Value	Description	Reset Value
3:0	DSS		Data Size Select. This field controls the number of bits transferred in each frame. Values 0000-0010 are not supported and should not be used.	0000
		0011	4-bit transfer	
		0100	5-bit transfer	
		0101	6-bit transfer	
		0110	7-bit transfer	
		0111	8-bit transfer	
		1000	9-bit transfer	
		1001	10-bit transfer	
		1010	11-bit transfer	
		1011	12-bit transfer	
		1100	13-bit transfer	
		1101	14-bit transfer	
		1110	15-bit transfer	
		1111	16-bit transfer	
		5:4	FRF	
00	SPI			
01	TI			
10	Microwire			
11	This combination is not supported and should not be used.			
6	CPOL		Clock Out Polarity. This bit is only used in SPI mode.	0
		0	SSP controller maintains the bus clock low between frames.	
	1	SSP controller maintains the bus clock high between frames.		
7	CPHA		Clock Out Phase. This bit is only used in SPI mode.	0
		0	SSP controller captures serial data on the first clock transition of the frame, that is, the transition away from the inter-frame state of the clock line.	
	1	SSP controller captures serial data on the second clock transition of the frame, that is, the transition back to the inter-frame state of the clock line.		
15:8	SCR		Serial Clock Rate. The number of prescaler-output clocks per bit on the bus, minus one. Given that CPSDVSR is the prescale divider, and the APB clock PCLK clocks the prescaler, the bit frequency is $PCLK / (CPSDVSR \times [SCR+1])$.	0x00

6.2 SSP0 Control Register 1

This register controls certain aspects of the operation of the SSP controller.

Table 166: SSP Control Register 1 (SSP0CR1 - address 0x4004 0004, SSP1CR1 - address 0x4005 8004) bit description

Bit	Symbol	Value	Description	Reset Value
0	LBM		Loop Back Mode.	0
		0	During normal operation.	
		1	Serial input is taken from the serial output (MOSI or MISO) rather than the serial input pin (MISO or MOSI respectively).	
1	SSE		SSP Enable.	0
		0	The SSP controller is disabled.	
		1	The SSP controller will interact with other devices on the serial bus. Software should write the appropriate control information to the other SSP registers and interrupt controller registers, before setting this bit.	
2	MS		Master/Slave Mode. This bit can only be written when the SSE bit is 0.	0
		0	The SSP controller acts as a master on the bus, driving the SCLK, MOSI, and SSEL lines and receiving the MISO line.	
		1	The SSP controller acts as a slave on the bus, driving MISO line and receiving SCLK, MOSI, and SSEL lines.	
3	SOD		Slave Output Disable. This bit is relevant only in slave mode (MS = 1). If it is 1, this blocks this SSP controller from driving the transmit data line (MISO).	0
7:4	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

6.3 SSP Data Register

Software can write data to be transmitted to this register and read data that has been received.

Table 167: SSP Data Register (SSP0DR - address 0x4004 0008, SSP1DR - address 0x4005 8008) bit description

Bit	Symbol	Description	Reset Value
15:0	DATA	<p>Write: software can write data to be sent in a future frame to this register whenever the TNF bit in the Status register is 1, indicating that the Tx FIFO is not full. If the Tx FIFO was previously empty and the SSP controller is not busy on the bus, transmission of the data will begin immediately. Otherwise the data written to this register will be sent as soon as all previous data has been sent (and received). If the data length is less than 16 bit, software must right-justify the data written to this register.</p> <p>Read: software can read data from this register whenever the RNE bit in the Status register is 1, indicating that the Rx FIFO is not empty. When software reads this register, the SSP controller returns data from the least recent frame in the Rx FIFO. If the data length is less than 16 bit, the data is right-justified in this field with higher order bits filled with 0s.</p>	0x0000

6.4 SSP Status Register

This read-only register reflects the current status of the SSP controller.

Table 168: SSP Status Register (SSP0SR - address 0x4004 000C, SSP1SR - address 0x4005 800C) bit description

Bit	Symbol	Description	Reset Value
0	TFE	Transmit FIFO Empty. This bit is 1 if the Transmit FIFO is empty, 0 if not.	1
1	TNF	Transmit FIFO Not Full. This bit is 0 if the Tx FIFO is full, 1 if not.	1
2	RNE	Receive FIFO Not Empty. This bit is 0 if the Receive FIFO is empty, 1 if not.	0
3	RFF	Receive FIFO Full. This bit is 1 if the Receive FIFO is full, 0 if not.	0
4	BSY	Busy. This bit is 0 if the SSP controller is idle, 1 if it is currently sending/receiving a frame and/or the Tx FIFO is not empty.	0
7:5	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

6.5 SSP Clock Prescale Register

This register controls the factor by which the Prescaler divides the SSP peripheral clock SSP_PCLK to yield the prescaler clock that is, in turn, divided by the SCR factor in the SSPCR0 registers, to determine the bit clock.

Table 169: SSP0 Clock Prescale Register (SSP0CPSR - address 0x4004 0010, SSP1CPSR - address 0x4005 8010) bit description

Bit	Symbol	Description	Reset Value
7:0	CPSDVSR	This even value between 2 and 254, by which SSP_PCLK is divided to yield the prescaler output clock. Bit 0 always reads as 0.	0

Important: the SSPnCPSR value must be properly initialized, or the SSP controller will not be able to transmit data correctly.

In Slave mode, the SSP clock rate provided by the master must not exceed 1/12 of the SSP peripheral clock selected in [Section 3–4.15](#). The content of the SSPnCPSR register is not relevant.

In master mode, $CPSDVSR_{min} = 2$ or larger (even numbers only).

6.6 SSP Interrupt Mask Set/Clear Register

This register controls whether each of the four possible interrupt conditions in the SSP controller are enabled. Note that ARM uses the word “masked” in the opposite sense from classic computer terminology, in which “masked” meant “disabled”. ARM uses the word “masked” to mean “enabled”. To avoid confusion we will not use the word “masked”.

Table 170: SSP Interrupt Mask Set/Clear register (SSP0IMSC - address 0x4004 0014, SSP1IMSC - address 0x4005 8014) bit description

Bit	Symbol	Description	Reset Value
0	RORIM	Software should set this bit to enable interrupt when a Receive Overrun occurs, that is, when the Rx FIFO is full and another frame is completely received. The ARM spec implies that the preceding frame data is overwritten by the new frame data when this occurs.	0
1	RTIM	Software should set this bit to enable interrupt when a Receive Timeout condition occurs. A Receive Timeout occurs when the Rx FIFO is not empty, and no has not been read for a "timeout period".	0
2	RXIM	Software should set this bit to enable interrupt when the Rx FIFO is at least half full.	0
3	TXIM	Software should set this bit to enable interrupt when the Tx FIFO is at least half empty.	0
7:4	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

6.7 SSP Raw Interrupt Status Register

This read-only register contains a 1 for each interrupt condition that is asserted, regardless of whether or not the interrupt is enabled in the SSPIMSC registers.

Table 171: SSP Raw Interrupt Status register (SSP0RIS - address 0x4004 0018, SSP1RIS - address 0x4005 8018) bit description

Bit	Symbol	Description	Reset Value
0	RORRIS	This bit is 1 if another frame was completely received while the RxFIFO was full. The ARM spec implies that the preceding frame data is overwritten by the new frame data when this occurs.	0
1	RTRIS	This bit is 1 if the Rx FIFO is not empty, and has not been read for a "timeout period".	0
2	RXRIS	This bit is 1 if the Rx FIFO is at least half full.	0
3	TXRIS	This bit is 1 if the Tx FIFO is at least half empty.	1
7:4	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

6.8 SSP Masked Interrupt Status Register

This read-only register contains a 1 for each interrupt condition that is asserted and enabled in the SSPIMSC registers. When an SSP interrupt occurs, the interrupt service routine should read this register to determine the cause(s) of the interrupt.

Table 172: SSP Masked Interrupt Status register (SSP0MIS - address 0x4004 001C, SSP1MIS - address 0x4005 801C) bit description

Bit	Symbol	Description	Reset Value
0	RORMIS	This bit is 1 if another frame was completely received while the RxFIFO was full, and this interrupt is enabled.	0
1	RTMIS	This bit is 1 if the Rx FIFO is not empty, has not been read for a "timeout period", and this interrupt is enabled.	0
2	RXMIS	This bit is 1 if the Rx FIFO is at least half full, and this interrupt is enabled.	0
3	TXMIS	This bit is 1 if the Tx FIFO is at least half empty, and this interrupt is enabled.	0
7:4	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

6.9 SSP Interrupt Clear Register

Software can write one or more one(s) to this write-only register, to clear the corresponding interrupt condition(s) in the SSP controller. Note that the other two interrupt conditions can be cleared by writing or reading the appropriate FIFO or disabled by clearing the corresponding bit in SSPIMSC registers.

Table 173: SSP interrupt Clear Register (SSP0ICR - address 0x4004 0020, SSP1ICR - address 0x4005 8020) bit description

Bit	Symbol	Description	Reset Value
0	RORIC	Writing a 1 to this bit clears the "frame was received when RxFIFO was full" interrupt.	NA
1	RTIC	Writing a 1 to this bit clears the "Rx FIFO was not empty and has not been read for a timeout period" interrupt.	NA
7:2	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

7. Functional description

7.1 Texas Instruments synchronous serial frame format

[Figure 11–32](#) shows the 4-wire Texas Instruments synchronous serial frame format supported by the SSP module.

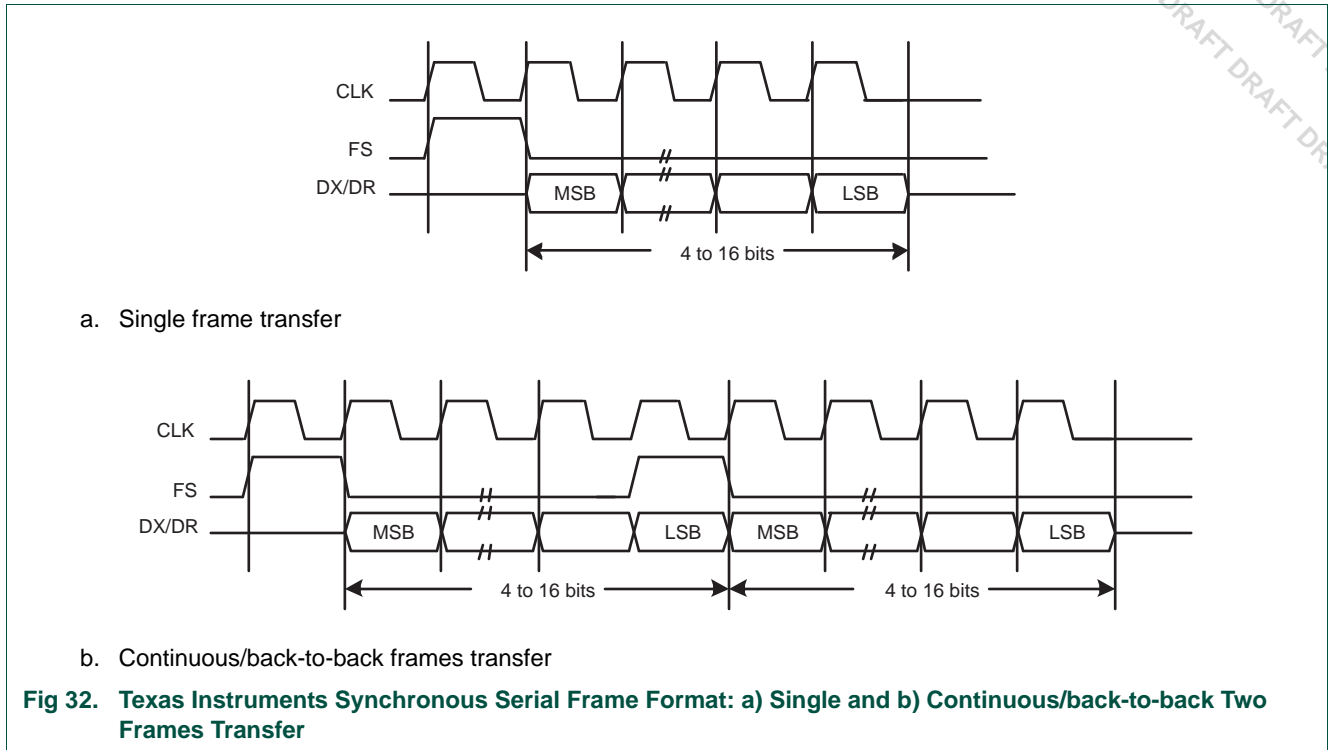


Fig 32. Texas Instruments Synchronous Serial Frame Format: a) Single and b) Continuous/back-to-back Two Frames Transfer

For device configured as a master in this mode, CLK and FS are forced LOW, and the transmit data line DX is in 3-state mode whenever the SSP is idle. Once the bottom entry of the transmit FIFO contains data, FS is pulsed HIGH for one CLK period. The value to be transmitted is also transferred from the transmit FIFO to the serial shift register of the transmit logic. On the next rising edge of CLK, the MSB of the 4-bit to 16-bit data frame is shifted out on the DX pin. Likewise, the MSB of the received data is shifted onto the DR pin by the off-chip serial slave device.

Both the SSP and the off-chip serial slave device then clock each data bit into their serial shifter on the falling edge of each CLK. The received data is transferred from the serial shifter to the receive FIFO on the first rising edge of CLK after the LSB has been latched.

7.2 SPI frame format

The SPI interface is a four-wire interface where the SSEL signal behaves as a slave select. The main feature of the SPI format is that the inactive state and phase of the SCK signal are programmable through the CPOL and CPHA bits within the SSPCR0 control register.

7.2.1 Clock Polarity (CPOL) and Phase (CPHA) control

When the CPOL clock polarity control bit is LOW, it produces a steady state low value on the SCK pin. If the CPOL clock polarity control bit is HIGH, a steady state high value is placed on the CLK pin when data is not being transferred.

The CPHA control bit selects the clock edge that captures data and allows it to change state. It has the most impact on the first bit transmitted by either allowing or not allowing a clock transition before the first data capture edge. When the CPHA phase control bit is LOW, data is captured on the first clock edge transition. If the CPHA clock phase control bit is HIGH, data is captured on the second clock edge transition.

7.2.2 SPI format with CPOL=0,CPHA=0

Single and continuous transmission signal sequences for SPI format with CPOL = 0, CPHA = 0 are shown in [Figure 11–33](#).

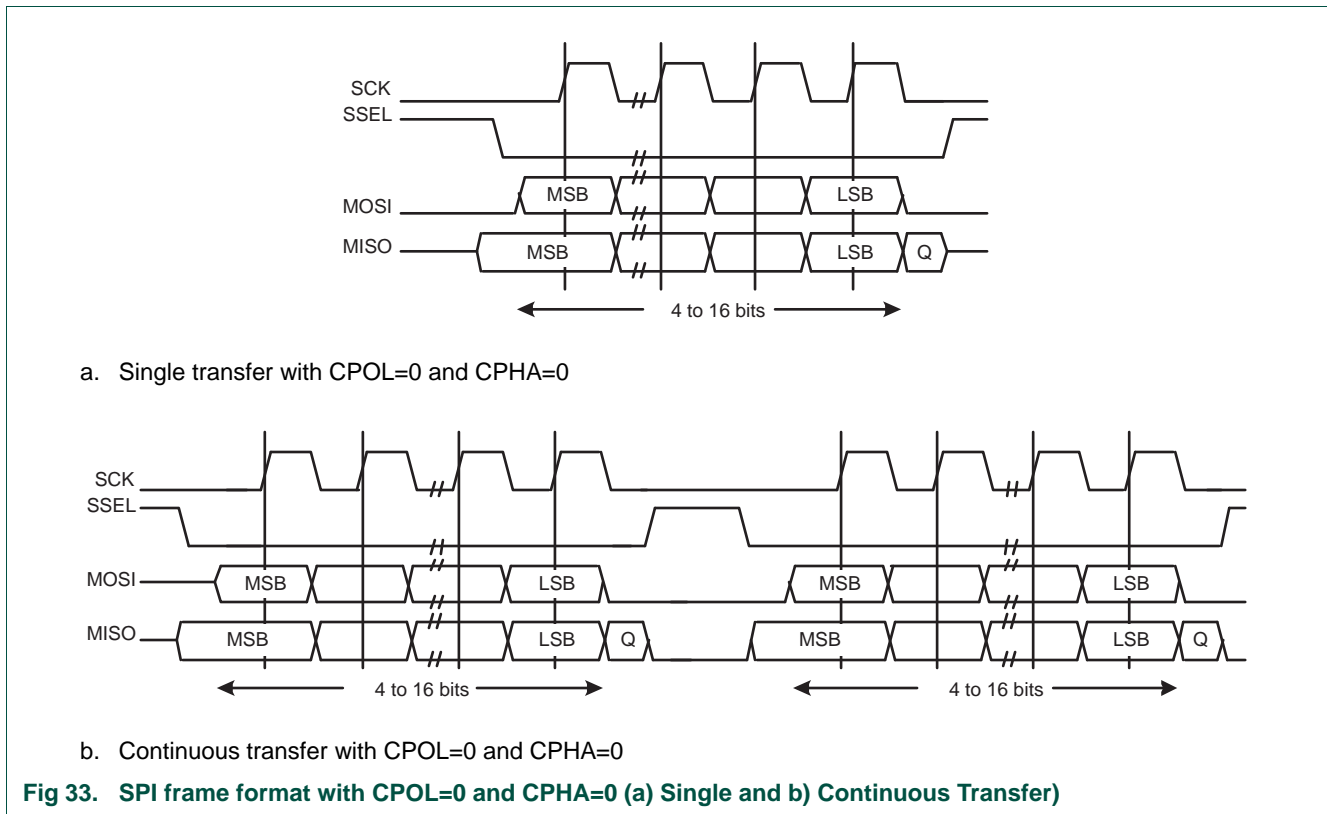


Fig 33. SPI frame format with CPOL=0 and CPHA=0 (a) Single and b) Continuous Transfer)

In this configuration, during idle periods:

- The CLK signal is forced LOW.
- SSEL is forced HIGH.
- The transmit MOSI/MISO pad is in high impedance.

If the SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSEL master signal being driven LOW. This causes slave data to be enabled onto the MISO input line of the master. Master's MOSI is enabled.

One half SCK period later, valid master data is transferred to the MOSI pin. Now that both the master and slave data have been set, the SCK master clock pin goes HIGH after one further half SCK period.

The data is captured on the rising and propagated on the falling edges of the SCK signal.

In the case of a single word transmission, after all bits of the data word have been transferred, the SSEL line is returned to its idle HIGH state one SCK period after the last bit has been captured.

However, in the case of continuous back-to-back transmissions, the SSEL signal must be pulsed HIGH between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not allow it to be altered if the CPHA bit is logic zero. Therefore the master device must raise the SSEL pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous transfer, the SSEL pin is returned to its idle state one SCK period after the last bit has been captured.

7.2.3 SPI format with CPOL=0,CPHA=1

The transfer signal sequence for SPI format with CPOL = 0, CPHA = 1 is shown in [Figure 11–34](#), which covers both single and continuous transfers.

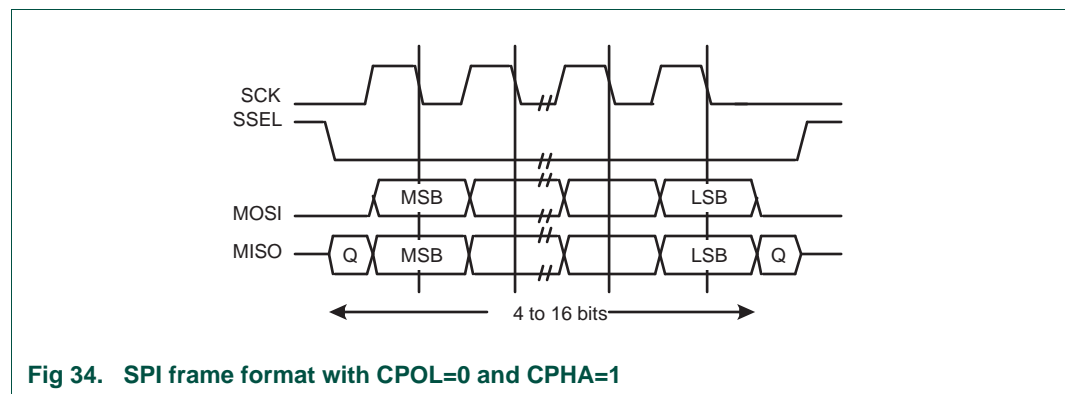


Fig 34. SPI frame format with CPOL=0 and CPHA=1

In this configuration, during idle periods:

- The CLK signal is forced LOW.
- SSEL is forced HIGH.
- The transmit MOSI/MISO pad is in high impedance.

If the SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSEL master signal being driven LOW. Master’s MOSI pin is enabled. After a further one half SCK period, both master and slave valid data is enabled onto their respective transmission lines. At the same time, the SCK is enabled with a rising edge transition.

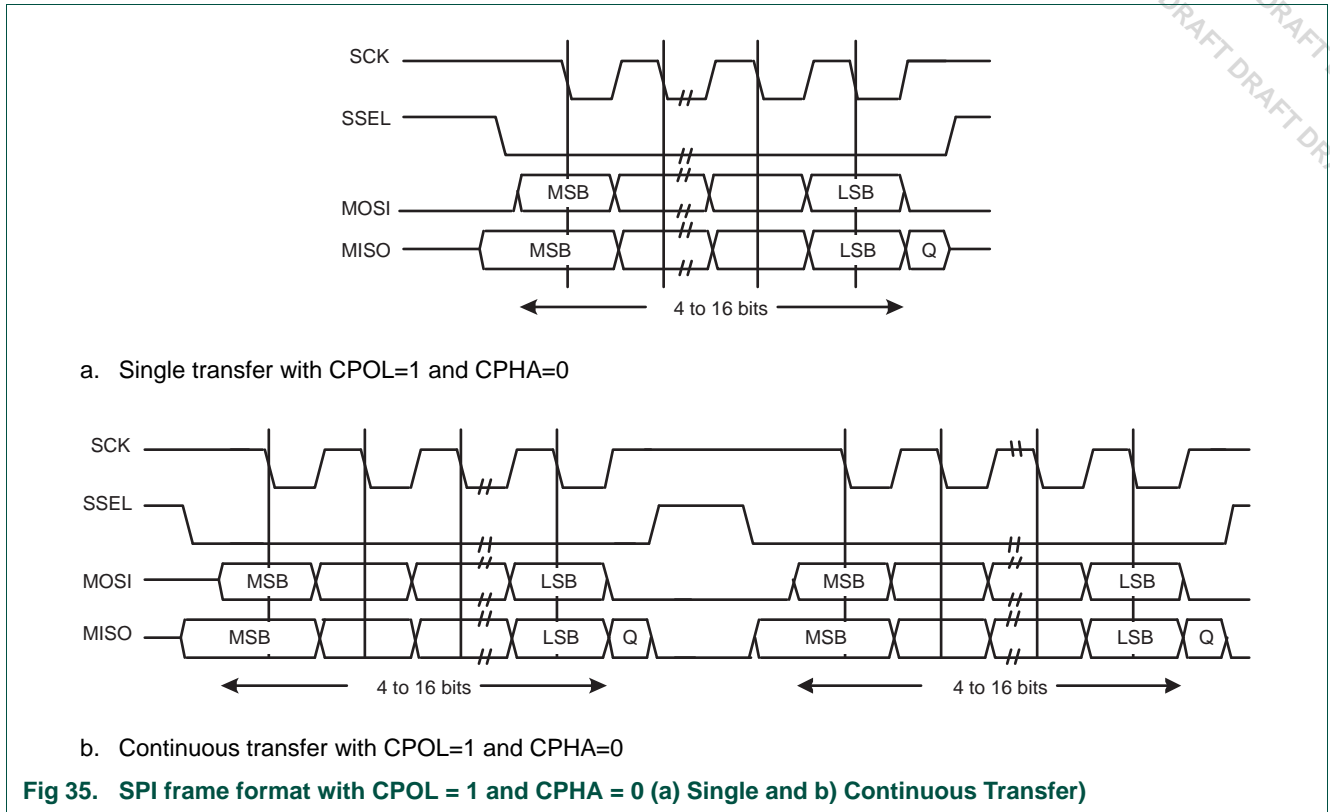
Data is then captured on the falling edges and propagated on the rising edges of the SCK signal.

In the case of a single word transfer, after all bits have been transferred, the SSEL line is returned to its idle HIGH state one SCK period after the last bit has been captured.

For continuous back-to-back transfers, the SSEL pin is held LOW between successive data words and termination is the same as that of the single word transfer.

7.2.4 SPI format with CPOL = 1,CPHA = 0

Single and continuous transmission signal sequences for SPI format with CPOL=1, CPHA=0 are shown in [Figure 11–35](#).



In this configuration, during idle periods:

- The CLK signal is forced HIGH.
- SSEL is forced HIGH.
- The transmit MOSI/MISO pad is in high impedance.

If the SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSEL master signal being driven LOW, which causes slave data to be immediately transferred onto the MISO line of the master. Master's MOSI pin is enabled.

One half period later, valid master data is transferred to the MOSI line. Now that both the master and slave data have been set, the SCK master clock pin becomes LOW after one further half SCK period. This means that data is captured on the falling edges and be propagated on the rising edges of the SCK signal.

In the case of a single word transmission, after all bits of the data word are transferred, the SSEL line is returned to its idle HIGH state one SCK period after the last bit has been captured.

However, in the case of continuous back-to-back transmissions, the SSEL signal must be pulsed HIGH between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not allow it to be altered if the CPHA bit is logic zero. Therefore the master device must raise the SSEL pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous transfer, the SSEL pin is returned to its idle state one SCK period after the last bit has been captured.

7.2.5 SPI format with CPOL = 1, CPHA = 1

The transfer signal sequence for SPI format with CPOL = 1, CPHA = 1 is shown in [Figure 11–36](#), which covers both single and continuous transfers.

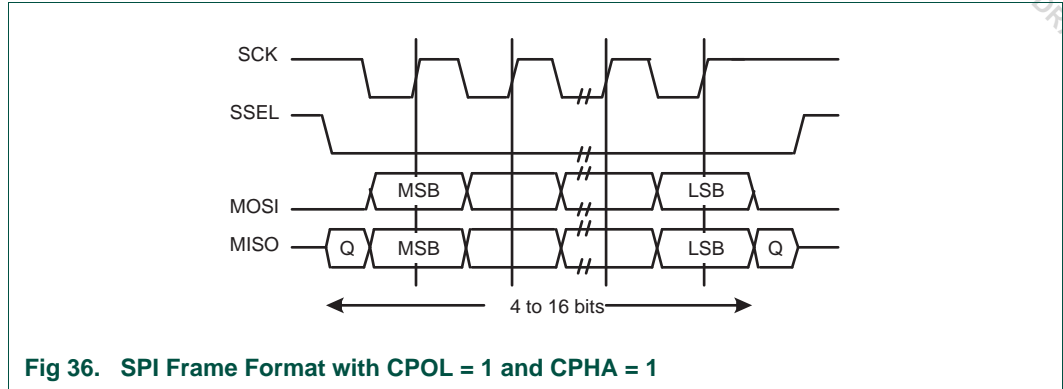


Fig 36. SPI Frame Format with CPOL = 1 and CPHA = 1

In this configuration, during idle periods:

- The CLK signal is forced HIGH.
- SSEL is forced HIGH.
- The transmit MOSI/MISO pad is in high impedance.

If the SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSEL master signal being driven LOW. Master’s MOSI is enabled. After a further one half SCK period, both master and slave data are enabled onto their respective transmission lines. At the same time, the SCK is enabled with a falling edge transition. Data is then captured on the rising edges and propagated on the falling edges of the SCK signal.

After all bits have been transferred, in the case of a single word transmission, the SSEL line is returned to its idle HIGH state one SCK period after the last bit has been captured. For continuous back-to-back transmissions, the SSEL pins remains in its active LOW state, until the final bit of the last word has been captured, and then returns to its idle state as described above. In general, for continuous back-to-back transfers the SSEL pin is held LOW between successive data words and termination is the same as that of the single word transfer.

7.3 Semiconductor Microwire frame format

[Figure 11–37](#) shows the Microwire frame format for a single frame. [Figure 11–38](#) shows the same format when back-to-back frames are transmitted.

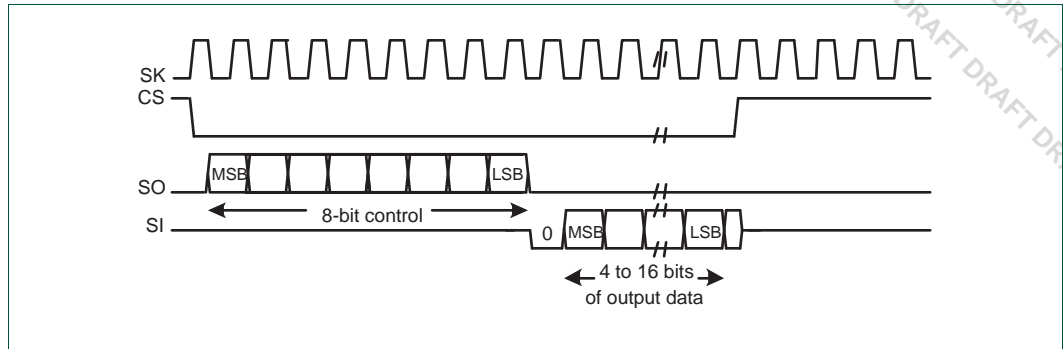


Fig 37. Microwire frame format (single transfer)

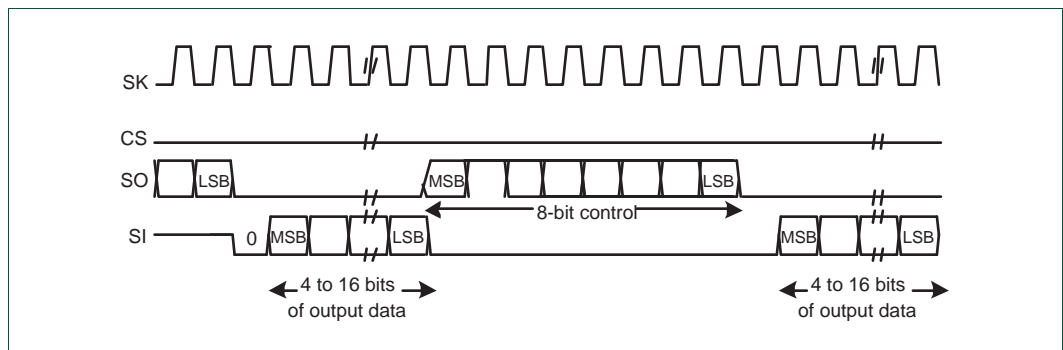


Fig 38. Microwire frame format (continuous transfers)

Microwire format is very similar to SPI format, except that transmission is half-duplex instead of full-duplex, using a master-slave message passing technique. Each serial transmission begins with an 8-bit control word that is transmitted from the SSP to the off-chip slave device. During this transmission, no incoming data is received by the SSP. After the message has been sent, the off-chip slave decodes it and, after waiting one serial clock after the last bit of the 8-bit control message has been sent, responds with the required data. The returned data is 4 to 16 bit in length, making the total frame length anywhere from 13 to 25 bits.

In this configuration, during idle periods:

- The SK signal is forced LOW.
- CS is forced HIGH.
- The transmit data line SO is arbitrarily forced LOW.

A transmission is triggered by writing a control byte to the transmit FIFO. The falling edge of CS causes the value contained in the bottom entry of the transmit FIFO to be transferred to the serial shift register of the transmit logic, and the MSB of the 8-bit control frame to be shifted out onto the SO pin. CS remains LOW for the duration of the frame transmission. The SI pin remains tristated during this transmission.

The off-chip serial slave device latches each control bit into its serial shifter on the rising edge of each SK. After the last bit is latched by the slave device, the control byte is decoded during a one clock wait-state, and the slave responds by transmitting data back to the SSP. Each bit is driven onto SI line on the falling edge of SK. The SSP in turn

latches each bit on the rising edge of SK. At the end of the frame, for single transfers, the CS signal is pulled HIGH one clock period after the last bit has been latched in the receive serial shifter, that causes the data to be transferred to the receive FIFO.

Note: The off-chip slave device can tristate the receive line either on the falling edge of SK after the LSB has been latched by the receive shifter, or when the CS pin goes HIGH.

For continuous transfers, data transmission begins and ends in the same manner as a single transfer. However, the CS line is continuously asserted (held LOW) and transmission of data occurs back to back. The control byte of the next frame follows directly after the LSB of the received data from the current frame. Each of the received values is transferred from the receive shifter on the falling edge SK, after the LSB of the frame has been latched into the SSP.

7.3.1 Setup and hold time requirements on CS with respect to SK in Microwire mode

In the Microwire mode, the SSP slave samples the first bit of receive data on the rising edge of SK after CS has gone LOW. Masters that drive a free-running SK must ensure that the CS signal has sufficient setup and hold margins with respect to the rising edge of SK.

Figure 11–39 illustrates these setup and hold time requirements. With respect to the SK rising edge on which the first bit of receive data is to be sampled by the SSP slave, CS must have a setup of at least two times the period of SK on which the SSP operates. With respect to the SK rising edge previous to this edge, CS must have a hold of at least one SK period.

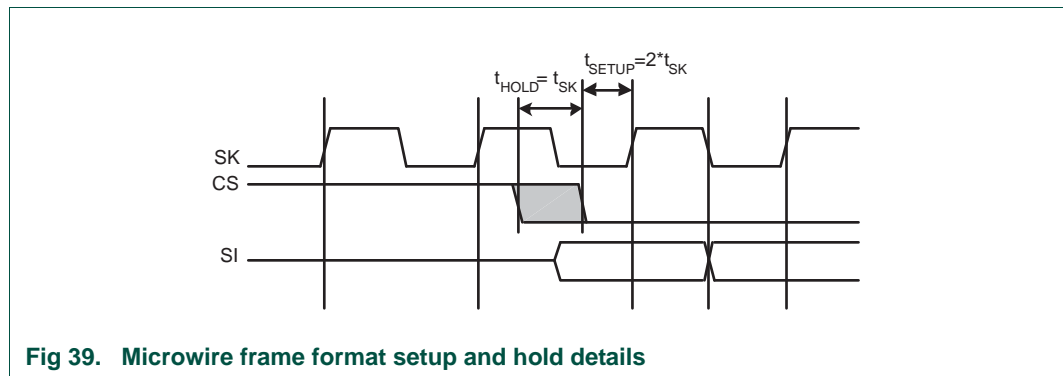


Fig 39. Microwire frame format setup and hold details

1. How to read this chapter

The 16-bit timer blocks are identical for all LPC111x parts.

2. Features

- Two 16-bit counter/timers with a programmable 16-bit prescaler.
- Counter or timer operation.
- One 16-bit capture channel that can take a snapshot of the timer value when an input signal transitions. A capture event may also optionally generate an interrupt.
- Four 16-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 three (CT16B0) or two (CT16B1) 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.
- For each timer, up to four match registers can be configured as PWM allowing to use up to three match outputs as single edge controlled PWM outputs.

3. Applications

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

4. Description

Each Counter/timer 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. Each counter/timer also includes one capture input to trap the timer value when an input signal transitions, optionally generating an interrupt.

In PWM mode, three match registers on CT16B0 and two match registers on CT16B1 can be used to provide a single-edge controlled PWM output on the match output pins. It is recommended to use the match registers that are not pinned out to control the PWM cycle length.

Remark: The 16-bit counter/timer0 (CT16B0) and the 16-bit counter/timer1 (CT16B1) are functionally identical except for the peripheral base address.

5. Pin description

[Table 12–174](#) gives a brief summary of each of the counter/timer related pins.

Table 174. Counter/timer pin description

Pin	Type	Description
CT16B0_CAP0 CT16B1_CAP0	Input	Capture Signal: A transition on a capture pin can be configured to load the Capture Register with the value in the counter/timer and optionally generate an interrupt. Counter/Timer block can select a capture signal as a clock source instead of the PCLK derived clock. For more details see Section 12–7.11 .
CT16B0_MAT[2:0] CT16B1_MAT[1:0]	Output	External Match Outputs of CT16B0/1: When a match register of CT16B0/1 (MR3:0) equals the timer counter (TC), this output can either toggle, go LOW, go HIGH, or do nothing. The External Match Register (EMR) and the PWM Control Register (PWMCON) control the functionality of this output.

6. Clocking and power control

The peripheral clocks (PCLK) to the 16-bit timers are provided by the system clock (see [Figure 3–3](#)). These clocks can be disabled through bit 7 and 8 in the AHBCLKCTRL register ([Section 3–4.14](#)) for power savings.

7. Register description

The 16-bit counter/timer0 contains the registers shown in [Table 12–175](#) and the 16-bit counter/timer1 contains the registers shown in [Table 12–176](#). More detailed descriptions follow.

Table 175. Register overview: 16-bit counter/timer 0 CT16B0 (base address 0x4000 C000)

Name	Access	Address offset	Description	Reset value ^[1]
TMR16B0IR	R/W	0x000	Interrupt Register (IR). The IR can be written to clear interrupts. The IR can be read to identify which of five possible interrupt sources are pending.	0
TMR16B0TCR	R/W	0x004	Timer Control Register (TCR). The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR.	0
TMR16B0TC	R/W	0x008	Timer Counter (TC). The 16-bit TC is incremented every PR+1 cycles of PCLK. The TC is controlled through the TCR.	0
TMR16B0PR	R/W	0x00C	Prescale Register (PR). When the Prescale Counter (below) is equal to this value, the next clock increments the TC and clears the PC.	0

Table 175. Register overview: 16-bit counter/timer 0 CT16B0 (base address 0x4000 C000) ...continued

Name	Access	Address offset	Description	Reset value ^[1]
TMR16B0PC	R/W	0x010	Prescale Counter (PC). The 16-bit PC is a counter which is incremented to the value stored in PR. When the value in PR is reached, the TC is incremented and the PC is cleared. The PC is observable and controllable through the bus interface.	0
TMR16B0MCR	R/W	0x014	Match Control Register (MCR). The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs.	0
TMR16B0MR0	R/W	0x018	Match Register 0 (MR0). MR0 can be enabled through the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt every time MR0 matches the TC.	0
TMR16B0MR1	R/W	0x01C	Match Register 1 (MR1). See MR0 description.	0
TMR16B0MR2	R/W	0x020	Match Register 2 (MR2). See MR0 description.	0
TMR16B0MR3	R/W	0x024	Match Register 3 (MR3). See MR0 description.	0
TMR16B0CCR	R/W	0x028	Capture Control Register (CCR). The CCR controls which edges of the capture inputs are used to load the Capture Registers and whether or not an interrupt is generated when a capture takes place.	0
TMR16B0CR0	RO	0x02C	Capture Register 0 (CR0). CR0 is loaded with the value of TC when there is an event on the CT16B0_CAP0 input.	0
TMR16B0EMR	R/W	0x03C	External Match Register (EMR). The EMR controls the match function and the external match pins CT16B0_MAT[2:0].	0
-	-	0x040 - 0x06C	reserved	-
TMR16B0CTCR	R/W	0x070	Count Control Register (CTCR). The CTCR selects between Timer and Counter mode, and in Counter mode selects the signal and edge(s) for counting.	0
TMR16B0PWMC	R/W	0x074	PWM Control Register (PWMCON). The PWMCON enables PWM mode for the external match pins CT16B0_MAT[2:0].	0

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

Table 176. Register overview: 16-bit counter/timer 1 CT16B1 (base address 0x4001 0000)

Name	Access	Address offset	Description	Reset value ^[1]
TMR16B1IR	R/W	0x000	Interrupt Register (IR). The IR can be written to clear interrupts. The IR can be read to identify which of five possible interrupt sources are pending.	0
TMR16B1TCR	R/W	0x004	Timer Control Register (TCR). The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR.	0
TMR16B1TC	R/W	0x008	Timer Counter (TC). The 16-bit TC is incremented every PR+1 cycles of PCLK. The TC is controlled through the TCR.	0
TMR16B1PR	R/W	0x00C	Prescale Register (PR). When the Prescale Counter (below) is equal to this value, the next clock increments the TC and clears the PC.	0
TMR16B1PC	R/W	0x010	Prescale Counter (PC). The 16-bit PC is a counter which is incremented to the value stored in PR. When the value in PR is reached, the TC is incremented and the PC is cleared. The PC is observable and controllable through the bus interface.	0

Table 176. Register overview: 16-bit counter/timer 1 CT16B1 (base address 0x4001 0000) ...continued

Name	Access	Address offset	Description	Reset value ^[1]
TMR16B1MCR	R/W	0x014	Match Control Register (MCR). The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs.	0
TMR16B1MR0	R/W	0x018	Match Register 0 (MR0). MR0 can be enabled through the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt every time MR0 matches the TC.	0
TMR16B1MR1	R/W	0x01C	Match Register 1 (MR1). See MR0 description.	0
TMR16B1MR2	R/W	0x020	Match Register 2 (MR2). See MR0 description.	0
TMR16B1MR3	R/W	0x024	Match Register 3 (MR3). See MR0 description.	0
TMR16B1CCR	R/W	0x028	Capture Control Register (CCR). The CCR controls which edges of the capture inputs are used to load the Capture Registers and whether or not an interrupt is generated when a capture takes place.	0
TMR16B1CR0	RO	0x02C	Capture Register 0 (CR0). CR0 is loaded with the value of TC when there is an event on the CT16B1_CAP0 input.	0
TMR16B1EMR	R/W	0x03C	External Match Register (EMR). The EMR controls the match function and the external match pins CT16B1_MAT[1:0].	0
-	-	0x040 - 0x06C	reserved	-
TMR16B1CTCR	R/W	0x070	Count Control Register (CTCR). The CTCR selects between Timer and Counter mode, and in Counter mode selects the signal and edge(s) for counting.	0
TMR16B1PWMC	R/W	0x074	PWM Control Register (PWMCN). The PWMCN enables PWM mode for the external match pins CT16B1_MAT[1:0].	0

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

7.1 Interrupt Register (TMR16B0IR and TMR16B1IR)

The Interrupt Register (IR) consists of four bits for the match interrupts and one bit for the capture interrupt. If an interrupt is generated then the corresponding bit in the IR will be HIGH. Otherwise, the bit will be LOW. Writing a logic one to the corresponding IR bit will reset the interrupt. Writing a zero has no effect.

Table 177. Interrupt Register (TMR16B0IR - address 0x4000 C000 and TMR16B1IR - address 0x4001 0000) 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
31:5	-	Reserved	-

7.2 Timer Control Register (TMR16B0TCR and TMR16B1TCR)

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

Table 178. Timer Control Register (TMR16B0TCR - address 0x4000 C004 and TMR16B1TCR - address 0x4001 0004) 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

7.3 Timer Counter (TMR16B0TC - address 0x4000 C008 and TMR16B1TC - address 0x4001 0008)

The 16-bit Timer Counter is incremented when the Prescale Counter reaches its terminal count. Unless it is reset before reaching its upper limit, the TC will count up through the value 0x0000 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.

7.4 Prescale Register (TMR16B0PR - address 0x4000 C00C and TMR16B1PR - address 0x4001 000C)

The 16-bit Prescale Register specifies the maximum value for the Prescale Counter.

7.5 Prescale Counter register (TMR16B0PC - address 0x4000 C010 and TMR16B1PC - address 0x4001 0010)

The 16-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 between the resolution of the timer and 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 TC to increment on every PCLK when PR = 0, every 2 PCLKs when PR = 1, etc.

7.6 Match Control Register (TMR16B0MCR and TMR16B1MCR)

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 12-179](#).

Table 179. Match Control Register (TMR16B0MCR - address 0x4000 C014 and TMR16B1MCR - address 0x4001 0014) 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	

Table 179. Match Control Register (TMR16B0MCR - address 0x4000 C014 and TMR16B1MCR - address 0x4001 0014) bit description ...continued

Bit	Symbol	Value	Description	Reset value
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

7.7 Match Registers (TMR16B0MR0/1/2/3 - addresses 0x4000 C018/1C/20/24 and TMR16B1MR0/1/2/3 - addresses 0x4001 0018/1C/20/24)

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.

7.8 Capture Control Register (TMR16B0CCR and TMR16B1CCR)

The Capture Control Register is used to control whether the Capture Register is loaded with the value in the Counter/timer 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.

Table 180. Capture Control Register (TMR16B0CCR - address 0x4000 C028 and TMR16B1CCR - address 0x4001 0028) bit description

Bit	Symbol	Value	Description	Reset value
0	CAP0RE	1	Capture on CT16Bn_CAP0 rising edge: a sequence of 0 then 1 on CT16Bn_CAP0 will cause CR0 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
1	CAP0FE	1	Capture on CT16Bn_CAP0 falling edge: a sequence of 1 then 0 on CT16Bn_CAP0 will cause CR0 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
2	CAP0I	1	Interrupt on CT16Bn_CAP0 event: a CR0 load due to a CT16Bn_CAP0 event will generate an interrupt.	0
		0	This feature is disabled.	
31:3	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

7.9 Capture Register (CT16B0CR0 - address 0x4000 C02C and CT16B1CR0 - address 0x4001 002C)

Each Capture register is associated with a device pin and may be loaded with the counter/timer 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.

7.10 External Match Register (TMR16B0EMR and TMR16B1EMR)

The External Match Register provides both control and status of the external match channels and external match pins CT16B0_MAT[2:0] and CT16B1_MAT[1:0].

If the match outputs are configured as PWM output in the PWMCON registers ([Section 12–7.12](#)), the function of the external match registers is determined by the PWM rules ([Section 12–7.13 “Rules for single edge controlled PWM outputs” on page 185](#)).

Table 181. External Match Register (TMR16B0EMR - address 0x4000 C03C and TMR16B1EMR - address 0x4001 003C) bit description

Bit	Symbol	Description	Reset value
0	EM0	External Match 0. This bit reflects the state of output CT16B0_MAT0/CT16B1_MAT0, whether or not this output is connected to its pin. When a match occurs between the TC and MR0, this bit can either toggle, go LOW, go HIGH, or do nothing. Bits EMR[5:4] control the functionality of this output. This bit is driven to the CT16B0_MAT0/CT16B1_MAT0 pins if the match function is selected in the IOCON registers (0 = LOW, 1 = HIGH).	0
1	EM1	External Match 1. This bit reflects the state of output CT16B0_MAT1/CT16B1_MAT1, whether or not this output is connected to its pin. When a match occurs between the TC and MR1, this bit can either toggle, go LOW, go HIGH, or do nothing. Bits EMR[7:6] control the functionality of this output. This bit is driven to the CT16B0_MAT1/CT16B1_MAT1 pins if the match function is selected in the IOCON registers (0 = LOW, 1 = HIGH).	0
2	EM2	External Match 2. This bit reflects the state of output match channel 2, whether or not this output is connected to its pin. When a match occurs between the TC and MR2, this bit can either toggle, go LOW, go HIGH, or do nothing. Bits EMR[9:8] control the functionality of this output. Note that on counter/timer 0 this match channel is not pinned out. This bit is driven to the CT16B1_MAT2 pin if the match function is selected in the IOCON registers (0 = LOW, 1 = HIGH).	0
3	EM3	External Match 3. This bit reflects the state of output of match channel 3. When a match occurs between the TC and MR3, this bit can either toggle, go LOW, go HIGH, or do nothing. Bits EMR[11:10] control the functionality of this output. There is no output pin available for this channel on either of the 16-bit timers.	0
5:4	EMC0	External Match Control 0. Determines the functionality of External Match 0. Table 12–182 shows the encoding of these bits.	00
7:6	EMC1	External Match Control 1. Determines the functionality of External Match 1. Table 12–182 shows the encoding of these bits.	00
9:8	EMC2	External Match Control 2. Determines the functionality of External Match 2. Table 12–182 shows the encoding of these bits.	00
11:10	EMC3	External Match Control 3. Determines the functionality of External Match 3. Table 12–182 shows the encoding of these bits.	00
31:12	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Table 182. 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 (CT16Bn_MATm pin is LOW if pinned out).
10	Set the corresponding External Match bit/output to 1 (CT16Bn_MATm pin is HIGH if pinned out).
11	Toggle the corresponding External Match bit/output.

7.11 Count Control Register (TMR16B0CTCR and TMR16B1CTCR)

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 half of the PCLK clock. Consequently, duration of the HIGH/LOW levels on the same CAP input in this case can not be shorter than $1/(2 \times \text{PCLK})$.

Table 183. Count Control Register (TMR16B0CTCR - address 0x4000 C070 and TMR16B1CTCR - address 0x4001 0070) bit description

Bit	Symbol	Value	Description	Reset value
1:0	Counter/ Timer Mode		This field selects which rising PCLK edges can increment Timer's Prescale Counter (PC), or clear PC and increment Timer Counter (TC).	00
		00	Timer Mode: 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.	
3:2	Count Input Select		In counter mode (when bits 1:0 in this register are not 00), these bits select which CAP pin is sampled for clocking:	00
		00	CT16Bn_CAP0	
		01	Reserved.	
		10	Reserved.	
		11	Reserved.	
31:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

7.12 PWM Control register (TMR16B0PWMC and TMR16B1PWMC)

The PWM Control Register is used to configure the match outputs as PWM outputs. Each match output can be independently set to perform either as PWM output or as match output whose function is controlled by the External Match Register (EMR).

For timer 0, three single-edge controlled PWM outputs can be selected on the CT16B0_MAT[2:0] outputs. For timer 1, two single-edged PWM outputs can be selected on the CT16B1_Mat[1:0] outputs. One additional match register determines the PWM cycle length. When a match occurs in any of the other match registers, the PWM output is

set to HIGH. The timer is reset by the match register that is configured to set the PWM cycle length. When the timer is reset to zero, all currently HIGH match outputs configured as PWM outputs are cleared.

Table 184. PWM Control Register (TMR16B0PWMC - address 0x4000 C074 and TMR16B1PWMC- address 0x4001 0074) bit description

Bit	Symbol	Description	Reset value
0	PWM enable	When one, PWM mode is enabled for CT16Bn_MAT0. When zero, CT16Bn_MAT0 is controlled by EM0.	0
1	PWM enable	When one, PWM mode is enabled for CT16Bn_MAT1. When zero, CT16Bn_MAT1 is controlled by EM1.	0
2	PWM enable	When one, PWM mode is enabled for match channel 2 or pin CT16B0_MAT2. When zero, match channel 2 or pin CT16B0_MAT2 is controlled by EM2. Match channel 2 is not pinned out on timer 1.	0
3	PWM enable	When one, PWM mode is enabled for match channel 3 match channel 3. When zero, match channel 3 match channel 3 is controlled by EM3. Note: It is recommended to use match channel 3 to set the PWM cycle because it is not pinned out.	0
4:32	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

7.13 Rules for single edge controlled PWM outputs

1. All single edge controlled PWM outputs go LOW at the beginning of each PWM cycle (timer is set to zero) unless their match value is equal to zero.
2. Each PWM output will go HIGH when its match value is reached. If no match occurs (i.e. the match value is greater than the PWM cycle length), the PWM output remains continuously LOW.
3. If a match value larger than the PWM cycle length is written to the match register, and the PWM signal is HIGH already, then the PWM signal will be cleared on the next start of the next PWM cycle.
4. If a match register contains the same value as the timer reset value (the PWM cycle length), then the PWM output will be reset to LOW on the next clock tick. Therefore, the PWM output will always consist of a one clock tick wide positive pulse with a period determined by the PWM cycle length (i.e. the timer reload value).
5. If a match register is set to zero, then the PWM output will go to HIGH the first time the timer goes back to zero and will stay HIGH continuously.

Note: When the match outputs are selected to serve as PWM outputs, the timer reset (MRnR) and timer stop (MRnS) bits in the Match Control Register MCR must be set to 0 except for the match register setting the PWM cycle length. For this register, set the MRnR bit to 1 to enable the timer reset when the timer value matches the value of the corresponding match register.

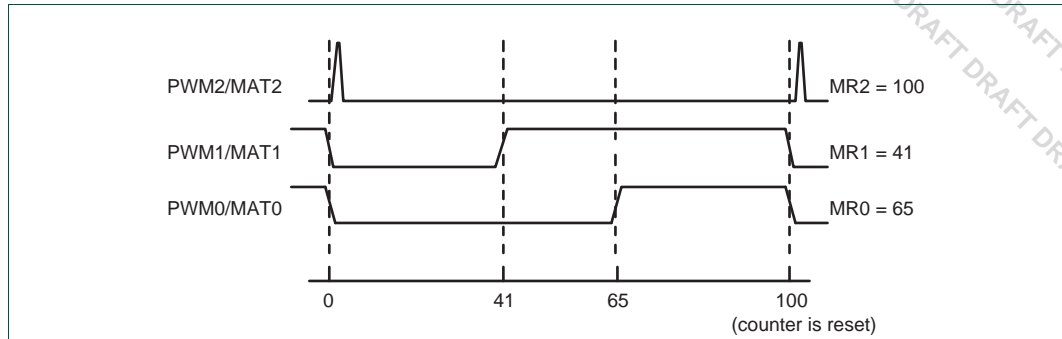


Fig 40. Sample PWM waveforms with a PWM cycle length of 100 (selected by MR3) and MAT3:0 enabled as PWM outputs by the PWCON register.

8. Example timer operation

Figure 12–41 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 12–42 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.

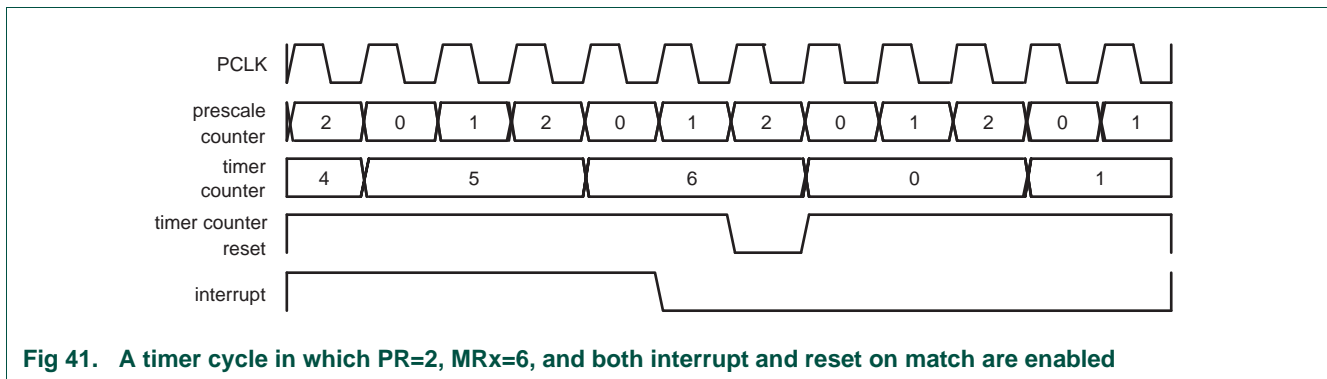


Fig 41. A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled

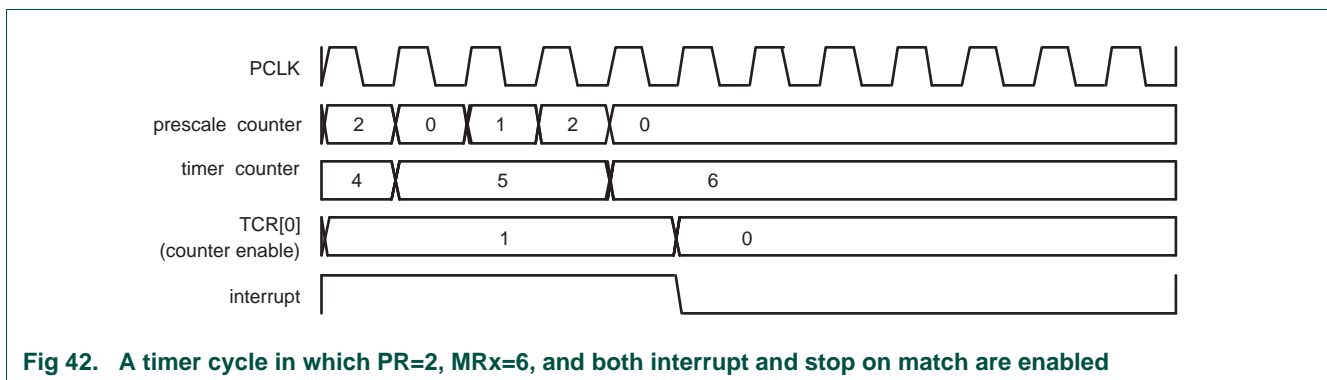


Fig 42. A timer cycle in which PR=2, MRx=6, and both interrupt and stop on match are enabled

9. Architecture

The block diagram for counter/timer0 and counter/timer1 is shown in [Figure 12-43](#).

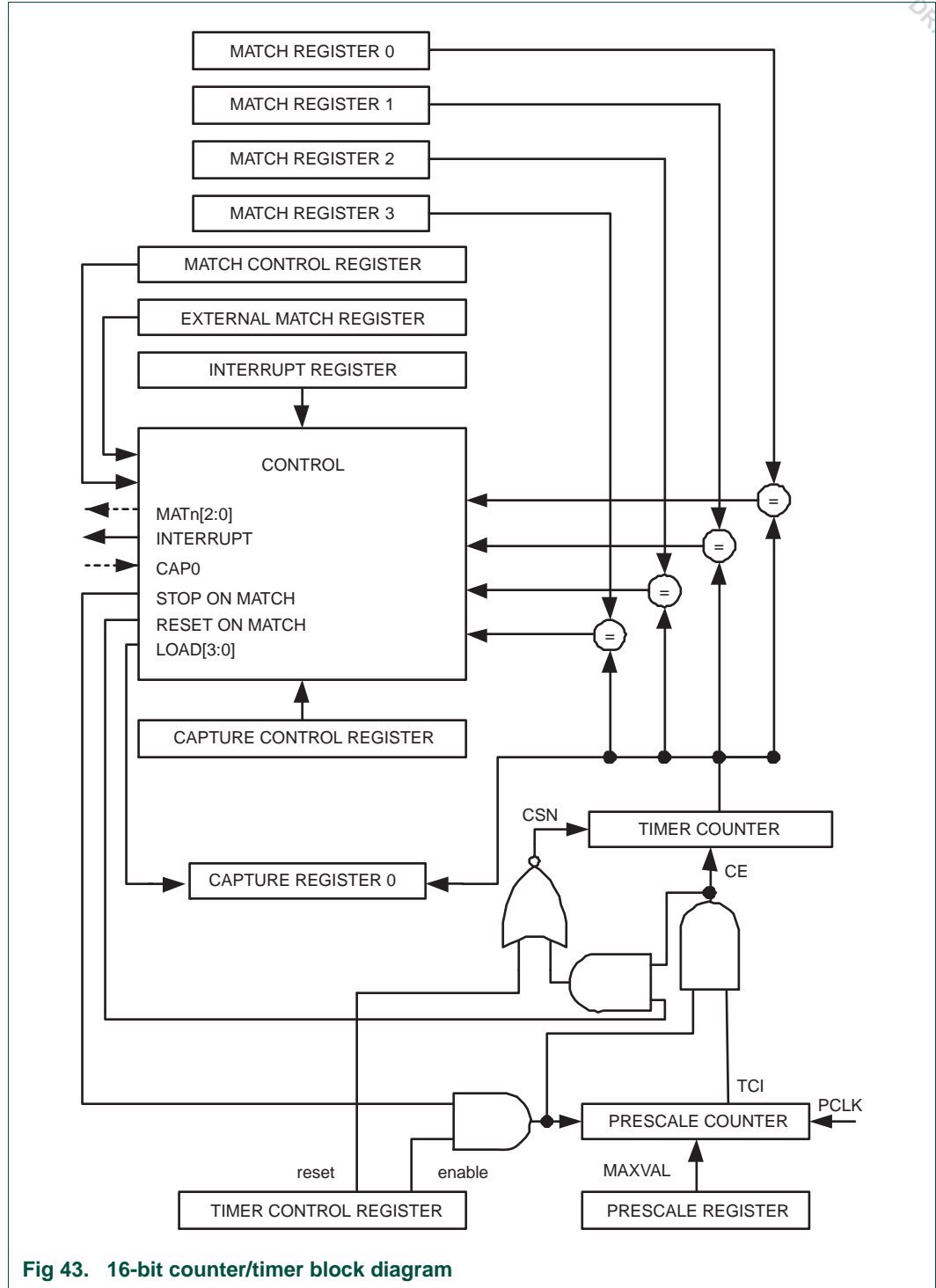


Fig 43. 16-bit counter/timer block diagram

1. How to read this chapter

The 32-bit timer blocks are identical for all LPC111x parts.

2. Features

- Two 32-bit counter/timers with a programmable 32-bit prescaler.
- Counter or Timer operation.
- One 32-bit capture channel 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.
- 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.
- For each timer, up to four match registers can be configured as PWM allowing to use up to three match outputs as single edge controlled PWM outputs.

3. Applications

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

4. Description

Each Counter/timer 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. Each counter/timer also includes one capture input to trap the timer value when an input signal transitions, optionally generating an interrupt.

In PWM mode, three match registers can be used to provide a single-edge controlled PWM output on the match output pins. One match register is used to control the PWM cycle length.

Remark: 32-bit counter/timer0 (CT32B0) and 32-bit counter/timer1 (CT32B1) are functionally identical except for the peripheral base address.

5. Pin description

[Table 13–185](#) gives a brief summary of each of the counter/timer related pins.

Table 185. Counter/timer pin description

Pin	Type	Description
CT32B0_CAP0 CT32B1_CAP0	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. The counter/timer block can select a capture signal as a clock source instead of the PCLK derived clock. For more details see Section 13–7.11 “Count Control Register (TMR32B0CTCR and TMR32B1TCR)” on page 195 .
CT32B0_MAT[3:0] CT32B1_MAT[3:0]	Output	External Match Output of CT32B0/1: When a match register TMR32B0/1MR3:0 equals the timer counter (TC), this output can either toggle, go LOW, go HIGH, or do nothing. The External Match Register (EMR) and the PWM Control register (PWMCON) control the functionality of this output.

6. Clocking and power control

The peripheral clocks (PCLK) to the 32-bit timers are provided by the system clock (see [Figure 3–3](#)). These clocks can be disabled through bits 9 and 10 in the AHBCLKCTRL register ([Section 3–4.14](#)) for power savings.

7. Register description

32-bit counter/timer0 contains the registers shown in [Table 13–186](#) and 32-bit counter/timer1 contains the registers shown in [Table 13–187](#). More detailed descriptions follow.

Table 186. Register overview: 32-bit counter/timer 0 CT32B0 (base address 0x4001 4000)

Name	Access	Address offset	Description	Reset value ^[1]
TMR32B0IR	R/W	0x000	Interrupt Register (IR). The IR can be written to clear interrupts. The IR can be read to identify which of five possible interrupt sources are pending.	0
TMR32B0TCR	R/W	0x004	Timer Control Register (TCR). The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR.	0
TMR32B0TC	R/W	0x008	Timer Counter (TC). The 32-bit TC is incremented every PR+1 cycles of PCLK. The TC is controlled through the TCR.	0
TMR32B0PR	R/W	0x00C	Prescale Register (PR). When the Prescale Counter (below) is equal to this value, the next clock increments the TC and clears the PC.	0
TMR32B0PC	R/W	0x010	Prescale Counter (PC). The 32-bit PC is a counter which is incremented to the value stored in PR. When the value in PR is reached, the TC is incremented and the PC is cleared. The PC is observable and controllable through the bus interface.	0

Table 186. Register overview: 32-bit counter/timer 0 CT32B0 (base address 0x4001 4000) ...continued

Name	Access	Address offset	Description	Reset value ^[1]
TMR32B0MCR	R/W	0x014	Match Control Register (MCR). The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs.	0
TMR32B0MR0	R/W	0x018	Match Register 0 (MR0). MR0 can be enabled through the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt every time MR0 matches the TC.	0
TMR32B0MR1	R/W	0x01C	Match Register 1 (MR1). See MR0 description.	0
TMR32B0MR2	R/W	0x020	Match Register 2 (MR2). See MR0 description.	0
TMR32B0MR3	R/W	0x024	Match Register 3 (MR3). See MR0 description.	0
TMR32B0CCR	R/W	0x028	Capture Control Register (CCR). The CCR controls which edges of the capture inputs are used to load the Capture Registers and whether or not an interrupt is generated when a capture takes place.	0
TMR32B0CR0	RO	0x02C	Capture Register 0 (CR0). CR0 is loaded with the value of TC when there is an event on the CT32B0_CAP0 input.	0
TMR32B0EMR	R/W	0x03C	External Match Register (EMR). The EMR controls the match function and the external match pins CT32B0_MAT[3:0].	0
-	-	0x040 - 0x06C	reserved	-
TMR32B0CTCR	R/W	0x070	Count Control Register (CTCR). The CTCR selects between Timer and Counter mode, and in Counter mode selects the signal and edge(s) for counting.	0
TMR32B0PWMC	R/W	0x074	PWM Control Register (PWMCON). The PWMCON enables PWM mode for the external match pins CT32B0_MAT[3:0].	0

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

Table 187. Register overview: 32-bit counter/timer 1 CT32B1 (base address 0x4001 8000)

Name	Access	Address offset	Description	Reset value ^[1]
TMR32B1IR	R/W	0x000	Interrupt Register (IR). The IR can be written to clear interrupts. The IR can be read to identify which of five possible interrupt sources are pending.	0
TMR32B1TCR	R/W	0x004	Timer Control Register (TCR). The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR.	0
TMR32B1TC	R/W	0x008	Timer Counter (TC). The 32-bit TC is incremented every PR+1 cycles of PCLK. The TC is controlled through the TCR.	0
TMR32B1PR	R/W	0x00C	Prescale Register (PR). When the Prescale Counter (below) is equal to this value, the next clock increments the TC and clears the PC.	0
TMR32B1PC	R/W	0x010	Prescale Counter (PC). The 32-bit PC is a counter which is incremented to the value stored in PR. When the value in PR is reached, the TC is incremented and the PC is cleared. The PC is observable and controllable through the bus interface.	0
TMR32B1MCR	R/W	0x014	Match Control Register (MCR). The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs.	0
TMR32B1MR0	R/W	0x018	Match Register 0 (MR0). MR0 can be enabled through the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt every time MR0 matches the TC.	0

Table 187. Register overview: 32-bit counter/timer 1 CT32B1 (base address 0x4001 8000) ...continued

Name	Access	Address offset	Description	Reset value ^[1]
TMR32B1MR1	R/W	0x01C	Match Register 1 (MR1). See MR0 description.	0
TMR32B1MR2	R/W	0x020	Match Register 2 (MR2). See MR0 description.	0
TMR32B1MR3	R/W	0x024	Match Register 3 (MR3). See MR0 description.	0
TMR32B1CCR	R/W	0x028	Capture Control Register (CCR). The CCR controls which edges of the capture inputs are used to load the Capture Registers and whether or not an interrupt is generated when a capture takes place.	0
TMR32B1CR0	RO	0x02C	Capture Register 0 (CR0). CR0 is loaded with the value of TC when there is an event on the CT32B1_CAP0 input.	0
TMR32B1EMR	R/W	0x03C	External Match Register (EMR). The EMR controls the match function and the external match pins CT32B1_MAT[3:0].	0
-	-	0x040 - 0x06C	reserved	-
TMR32B1CTCR	R/W	0x070	Count Control Register (CTCR). The CTCR selects between Timer and Counter mode, and in Counter mode selects the signal and edge(s) for counting.	0
TMR32B1PWMC	R/W	0x074	PWM Control Register (PWMCON). The PWMCON enables PWM mode for the external match pins CT32B1_MAT[3:0].	0

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

7.1 Interrupt Register (TMR32B0IR and TMR32B1IR)

The Interrupt Register consists of four bits for the match interrupts and one bit for the capture interrupts. If an interrupt is generated then the corresponding bit in the IR will be HIGH. Otherwise, the bit will be LOW. Writing a logic one to the corresponding IR bit will reset the interrupt. Writing a zero has no effect.

Table 188: Interrupt Register (TMR32B0IR - address 0x4001 4000 and TMR32B1IR - address 0x4001 8000) 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
31:5	-	Reserved	-

7.2 Timer Control Register (TMR32B0TCR and TMR32B1TCR)

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

Table 189: Timer Control Register (TMR32B0TCR - address 0x4001 4004 and TMR32B1TCR - address 0x4001 8004) 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

7.3 Timer Counter (TMR32B0TC - address 0x4001 4008 and TMR32B1TC - address 0x4001 8008)

The 32-bit Timer Counter is incremented when the Prescale Counter reaches its terminal count. Unless it is reset before reaching its upper limit, the TC 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.

7.4 Prescale Register (TMR32B0PR - address 0x4001 400C and TMR32B1PR - address 0x4001 800C)

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

7.5 Prescale Counter Register (TMR32B0PC - address 0x4001 4010 and TMR32B1PC - address 0x4001 8010)

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 between the resolution of the timer and 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 TC to increment on every PCLK when PR = 0, every 2 PCLKs when PR = 1, etc.

7.6 Match Control Register (TMR32B0MCR and TMR32B1MCR)

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 13–190](#).

Table 190: Match Control Register (TMR32B0MCR - address 0x4001 4014 and TMR32B1MCR - address 0x4001 8014) 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	

Table 190: Match Control Register (TMR32B0MCR - address 0x4001 4014 and TMR32B1MCR - address 0x4001 8014) bit description

Bit	Symbol	Value	Description	Reset value
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

7.7 Match Registers (TMR32B0MR0/1/2/3 - addresses 0x4001 4018/1C/20/24 and TMR32B1MR0/1/2/3 addresses 0x4001 8018/1C/20/24)

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.

7.8 Capture Control Register (TMR32B0CCR and TMR32B1CCR)

The Capture Control Register is used to control whether the Capture Register 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.

Table 191: Capture Control Register (TMR32B0CCR - address 0x4001 4028 and TMR32B1CCR - address 0x4001 8028) bit description

Bit	Symbol	Value	Description	Reset value
0	CAP0RE	1	Capture on CT32Bn_CAP0 rising edge: a sequence of 0 then 1 on CT32Bn_CAP0 will cause CR0 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
1	CAP0FE	1	Capture on CT32Bn_CAP0 falling edge: a sequence of 1 then 0 on CT32Bn_CAP0 will cause CR0 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
2	CAP0I	1	Interrupt on CT32Bn_CAP0 event: a CR0 load due to a CT32Bn_CAP0 event will generate an interrupt.	0
		0	This feature is disabled.	
31:3	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

7.9 Capture Register (TMR32B0CR0 - address 0x4001 402C and TMR32B1CR0 - address 0x4001 802C)

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.

7.10 External Match Register (TMR32B0EMR and TMR32B1EMR)

The External Match Register provides both control and status of the external match pins CAP32Bn_MAT[3:0].

If the match outputs are configured as PWM output, the function of the external match registers is determined by the PWM rules ([Section 13–7.13 “Rules for single edge controlled PWM outputs” on page 197](#)).

Table 192: External Match Register (TMR32B0EMR - address 0x4001 403C and TMR32B1EMR - address 0x4001 803C) bit description

Bit	Symbol	Description	Reset value
0	EM0	External Match 0. This bit reflects the state of output CT32Bn_MAT0, whether or not this output is connected to its pin. When a match occurs between the TC and MR0, this bit can either toggle, go LOW, go HIGH, or do nothing. Bits EMR[5:4] control the functionality of this output. This bit is driven to the CT32B0_MAT0/CT16B1_MAT0 pins if the match function is selected in the IOCON registers (0 = LOW, 1 = HIGH).	0
1	EM1	External Match 1. This bit reflects the state of output CT32Bn_MAT1, whether or not this output is connected to its pin. When a match occurs between the TC and MR1, this bit can either toggle, go LOW, go HIGH, or do nothing. Bits EMR[7:6] control the functionality of this output. This bit is driven to the CT32B0_MAT1/CT16B1_MAT1 pins if the match function is selected in the IOCON registers (0 = LOW, 1 = HIGH).	0
2	EM2	External Match 2. This bit reflects the state of output CT32Bn_MAT2, whether or not this output is connected to its pin. When a match occurs between the TC and MR2, this bit can either toggle, go LOW, go HIGH, or do nothing. Bits EMR[9:8] control the functionality of this output. This bit is driven to the CT32B0_MAT2/CT16B1_MAT2 pins if the match function is selected in the IOCON registers (0 = LOW, 1 = HIGH).	0
3	EM3	External Match 3. This bit reflects the state of output CT32Bn_MAT3, whether or not this output is connected to its pin. When a match occurs between the TC and MR3, this bit can either toggle, go LOW, go HIGH, or do nothing. Bits EMR[11:10] control the functionality of this output. This bit is driven to the CT32B0_MAT3/CT16B1_MAT3 pins if the match function is selected in the IOCON registers (0 = LOW, 1 = HIGH).	0
5:4	EMC0	External Match Control 0. Determines the functionality of External Match 0. Table 13–193 shows the encoding of these bits.	00
7:6	EMC1	External Match Control 1. Determines the functionality of External Match 1. Table 13–193 shows the encoding of these bits.	00
9:8	EMC2	External Match Control 2. Determines the functionality of External Match 2. Table 13–193 shows the encoding of these bits.	00
11:10	EMC3	External Match Control 3. Determines the functionality of External Match 3. Table 13–193 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 193. 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 (CT32Bn_MATm pin is LOW if pinned out).
10	Set the corresponding External Match bit/output to 1 (CT32Bn_MATm pin is HIGH if pinned out).
11	Toggle the corresponding External Match bit/output.

7.11 Count Control Register (TMR32B0CTCR and TMR32B1TCR)

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 half of the PCLK clock. Consequently, duration of the HIGH/LOW levels on the same CAP input in this case can not be shorter than $1/(2 \times \text{PCLK})$.

Table 194: Count Control Register (TMR32B0CTCR - address 0x4001 4070 and TMR32B1TCR - address 0x4001 8070) bit description

Bit	Symbol	Value	Description	Reset value
1:0	Counter/ Timer Mode		This field selects which rising PCLK edges can increment Timer's Prescale Counter (PC), or clear PC and increment Timer Counter (TC).	00
		00	Timer Mode: 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.	
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: CT32Bn_	00
		00	CAP0	
		01	Reserved	
		10	Reserved	
		11	Reserved	
			Note: If Counter mode is selected in the TnCTCR, the 3 bits for that input in the Capture Control Register (TnCCR) must be programmed as 000.	
31:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

7.12 PWM Control Register (TMR32B0PWMC and TMR32B1PWMC)

The PWM Control Register is used to configure the match outputs as PWM outputs. Each match output can be independently set to perform either as PWM output or as match output whose function is controlled by the External Match Register (EMR).

For each timer, a maximum of three-single edge controlled PWM outputs can be selected on the MATn[2:0] outputs. One additional match register determines the PWM cycle length. When a match occurs in any of the other match registers, the PWM output is set to

HIGH. The timer is reset by the match register that is configured to set the PWM cycle length. When the timer is reset to zero, all currently HIGH match outputs configured as PWM outputs are cleared.

Table 195: PWM Control Register (TMR32B0PWMC - 0x4001 4074 and TMR32B1PWMC - 0x4001 8074) bit description

Bit	Symbol	Description	Reset value
0	PWM enable	When one, PWM mode is enabled for CT32Bn_MAT0. When zero, CT32Bn_MAT0 is controlled by EM0.	0
1	PWM enable	When one, PWM mode is enabled for CT32Bn_MAT1. When zero, CT32Bn_MAT1 is controlled by EM1.	0
2	PWM enable	When one, PWM mode is enabled for CT32Bn_MAT2. When zero, CT32Bn_MAT2 is controlled by EM2.	0
3	PWM enable	When one, PWM mode is enabled for CT32Bn_MAT3. When zero, CT32Bn_MAT3 is controlled by EM3. Note: It is recommended to use match channel 3 to set the PWM cycle.	0
4:32	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

7.13 Rules for single edge controlled PWM outputs

1. All single edge controlled PWM outputs go LOW at the beginning of each PWM cycle (timer is set to zero) unless their match value is equal to zero.
2. Each PWM output will go HIGH when its match value is reached. If no match occurs (i.e. the match value is greater than the PWM cycle length), the PWM output remains continuously LOW.
3. If a match value larger than the PWM cycle length is written to the match register, and the PWM signal is HIGH already, then the PWM signal will be cleared with the start of the next PWM cycle.
4. If a match register contains the same value as the timer reset value (the PWM cycle length), then the PWM output will be reset to LOW on the next clock tick after the timer reaches the match value. Therefore, the PWM output will always consist of a one clock tick wide positive pulse with a period determined by the PWM cycle length (i.e. the timer reload value).
5. If a match register is set to zero, then the PWM output will go to HIGH the first time the timer goes back to zero and will stay HIGH continuously.

Note: When the match outputs are selected to function as PWM outputs, the timer reset (MRnR) and timer stop (MRnS) bits in the Match Control Register MCR must be set to 0 except for the match register setting the PWM cycle length. For this register, set the MRnR bit to 1 to enable the timer reset when the timer value matches the value of the corresponding match register.

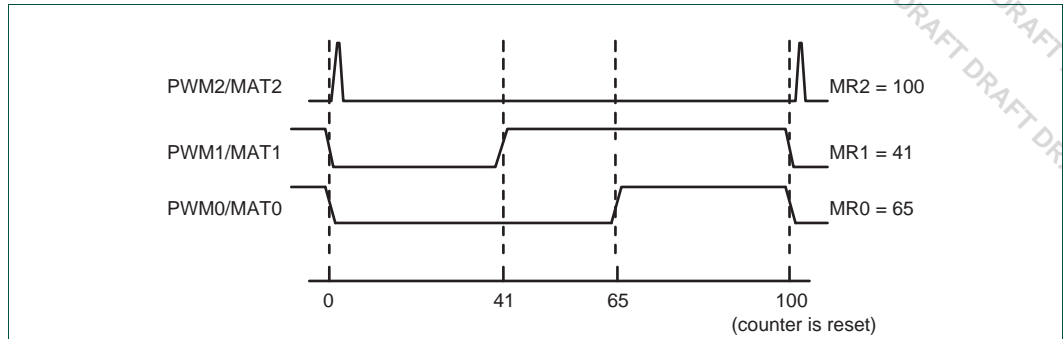


Fig 44. Sample PWM waveforms with a PWM cycle length of 100 (selected by MR3) and MAT3:0 enabled as PWM outputs by the PWCON register.

8. Example timer operation

Figure 13–45 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 13–46 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.

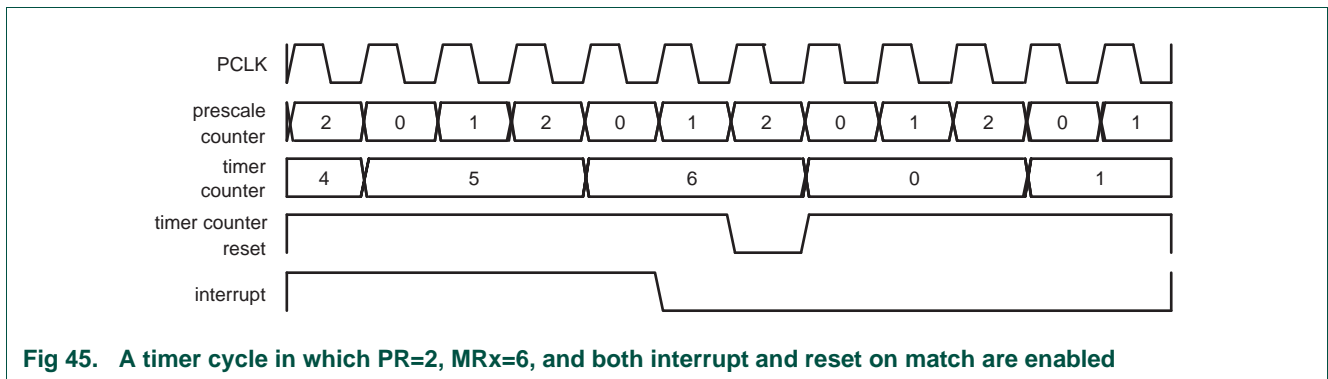


Fig 45. A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled

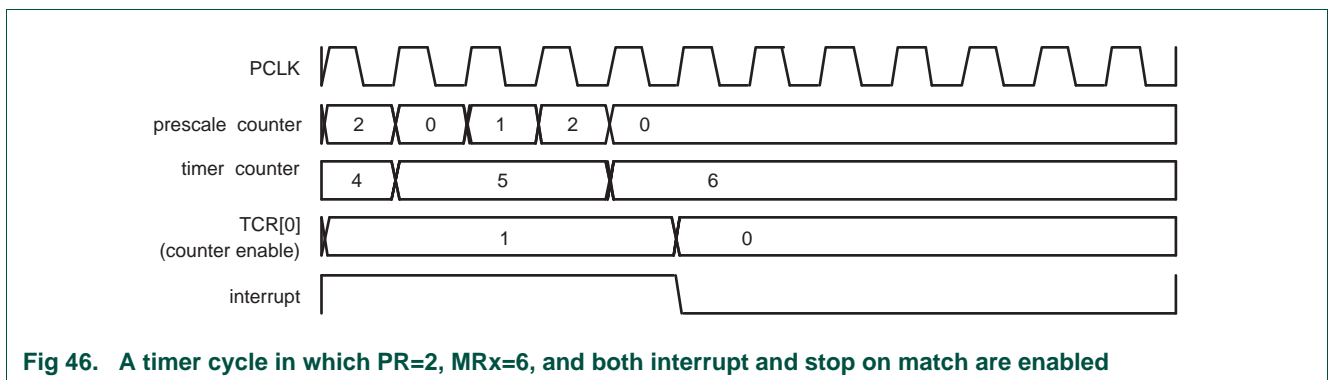


Fig 46. A timer cycle in which PR=2, MRx=6, and both interrupt and stop on match are enabled

9. Architecture

The block diagram for 32-bit counter/timer0 and 32-bit counter/timer1 is shown in [Figure 13-47](#).

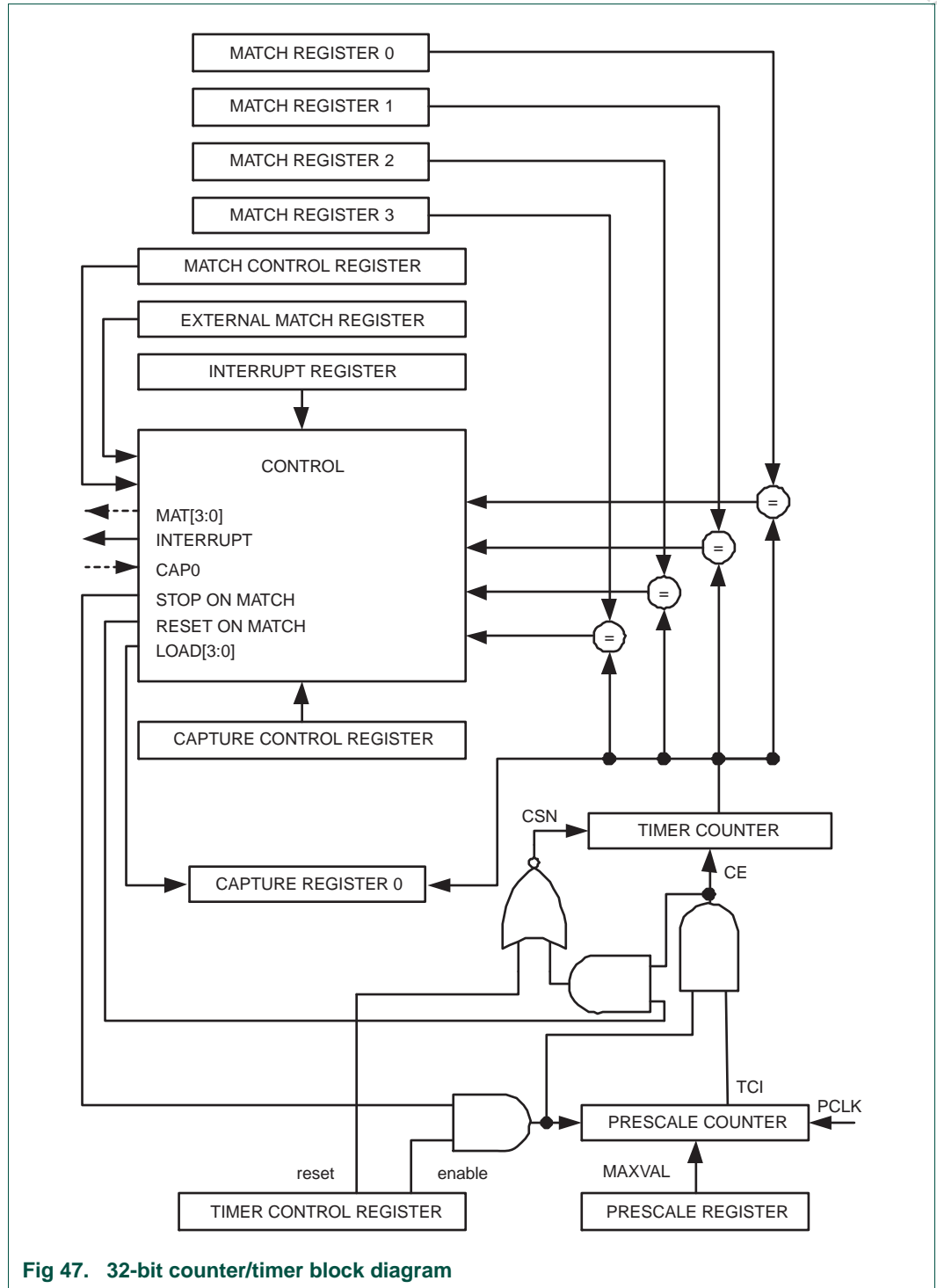


Fig 47. 32-bit counter/timer block diagram

1. How to read this chapter

The WDT block is identical for all LPC111x parts.

2. Features

- Internally resets chip if not periodically reloaded.
- Debug mode.
- Enabled by software but requires a hardware reset or a Watchdog reset/interrupt to be disabled.
- Incorrect/Incomplete feed sequence causes reset/interrupt if enabled.
- Flag to indicate Watchdog reset.
- Programmable 32 bit timer with internal pre-scaler.
- Selectable time period from $(T_{WDCLK} \times 256 \times 4)$ to $(T_{WDCLK} \times 2^{32} \times 4)$ in multiples of $T_{WDCLK} \times 4$.
- The Watchdog clock (WDCLK) source is selected in the syscon block from the Internal RC oscillator (IRC), the main clock, or the Watchdog oscillator, see [Table 3-24](#). This gives a wide range of potential timing choices for Watchdog operation under different power reduction conditions. For increased reliability, it also provides the ability to run the Watchdog timer from an entirely internal source that is not dependent on an external crystal and its associated components and wiring.
- The Watchdog timer can be configured to run in Deep-sleep mode.

3. Applications

The purpose of the Watchdog is to reset the microcontroller within a reasonable amount of time if it enters an erroneous state. When enabled, the Watchdog will generate a system reset if the user program fails to "feed" (or reload) the Watchdog within a predetermined amount of time.

4. Description

The Watchdog consists of a divide by 4 fixed pre-scaler and a 32-bit counter. The clock is fed to the timer via a pre-scaler. The timer decrements when clocked. The minimum value from which the counter decrements is 0xFF. Setting a value lower than 0xFF causes 0xFF to be loaded in the counter. Hence the minimum Watchdog interval is $(T_{WDCLK} \times 256 \times 4)$ and the maximum Watchdog interval is $(T_{WDCLK} \times 2^{32} \times 4)$ in multiples of $(T_{WDCLK} \times 4)$. The Watchdog should be used in the following manner:

1. Set the Watchdog timer constant reload value in WDTC register.
2. Setup the Watchdog timer operating mode in WDMOD register.

3. Enable the Watchdog by writing 0xAA followed by 0x55 to the WDFEED register.
4. The Watchdog should be fed again before the Watchdog counter underflows to prevent reset/interrupt.

When the Watchdog is in the reset mode and the counter underflows, the CPU will be reset, loading the stack pointer and program counter from the vector table as in the case of external reset. The Watchdog time-out flag (WDTOF) can be examined to determine if the Watchdog has caused the reset condition. The WDTOF flag must be cleared by software.

5. Clocking and power control

The watchdog timer block uses two clocks: PCLK and WDCLK. PCLK is used for the APB accesses to the watchdog registers and is derived from the system clock (see [Figure 3-3](#)). The WDCLK is used for the watchdog timer counting and is derived from the wdt_clk in [Figure 3-3](#). Several clocks can be used as a clock source for wdt_clk clock: the IRC, the watchdog oscillator, and the main clock. The clock source is selected in the syscon block (see [Section 3-4.19](#)). The WDCLK has its own clock divider ([Section 3-4.19](#)), which can also disable this clock.

There is some synchronization logic between these two clock domains. When the WDMOD and WDTC registers are updated by APB operations, the new value will take effect in 3 WDCLK cycles on the logic in the WDCLK clock domain. When the watchdog timer is counting on WDCLK, the synchronization logic will first lock the value of the counter on WDCLK and then synchronize it with the PCLK for reading as the WDTV register by the CPU.

The watchdog oscillator can be powered down in the PDRUNCFG register ([Section 3-4.35](#)) if it is not used. The clock to the watchdog register block (PCLK) can be disabled in the AHBCLKCTRL register ([Section 3-4.14](#)) for power savings.

6. Register description

The Watchdog contains four registers as shown in [Table 14-196](#) below.

Table 196. Register overview: Watchdog timer (base address 0x4000 4000)

Name	Access	Address offset	Description	Reset Value ^[1]
WDMOD	R/W	0x000	Watchdog mode register. This register contains the basic mode and status of the Watchdog Timer.	0
WDTC	R/W	0x004	Watchdog timer constant register. This register determines the time-out value.	0xFF
WDFEED	WO	0x008	Watchdog feed sequence register. Writing 0xAA followed by 0x55 to this register reloads the Watchdog timer with the value contained in WDTC.	NA
WDTV	RO	0x00C	Watchdog timer value register. This register reads out the current value of the Watchdog timer.	0xFF

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

6.1 Watchdog Mode register (WDMOD - 0x4000 0000)

The WDMOD register controls the operation of the Watchdog through the combination of WDEN and RESET bits. Note that a watchdog feed must be performed before any changes to the WDMOD register take effect.

Table 197. Watchdog Mode register (WDMOD - address 0x4000 4000) bit description

Bit	Symbol	Description	Reset Value
0	WDEN	WDEN Watchdog enable bit (Set Only). When 1, the watchdog timer is running.	0
1	WDRESET	WDRESET Watchdog reset enable bit (Set Only). When 1, a watchdog time-out will cause a chip reset.	0
2	WDTOF	WDTOF Watchdog time-out flag. Set when the watchdog timer times out, cleared by software.	0 (Only after POR and BOD reset)
3	WDINT	WDINT Watchdog interrupt flag (Read Only, not clearable by software).	0
7:4	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
31:8	-	reserved	-

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

WDTOF The Watchdog time-out flag is set when the Watchdog times out. This flag is cleared by software or a POR or Brown-Out-Detect reset.

WDINT The Watchdog interrupt flag is set when the Watchdog times out. This flag is cleared when any reset occurs. Once the watchdog interrupt is serviced, it can be disabled in the NVIC or the watchdog interrupt request will be generated indefinitely. The intent of the watchdog interrupt is to allow debugging watchdog activity without resetting the device when the watchdog overflows.

Watchdog reset or interrupt will occur any time the watchdog is running and has an operating clock source. Any clock source works in Sleep mode, and the IRC works in Deep-sleep mode. If a watchdog interrupt occurs in Sleep or Deep-sleep mode, it will wake up the device.

Table 198. Watchdog operating modes selection

WDEN	WDRESET	Mode of Operation
0	X (0 or 1)	Debug/Operate without the Watchdog running.
1	0	Watchdog interrupt mode: debug with the Watchdog interrupt but no WDRESET enabled. When this mode is selected, a watchdog counter underflow will set the WDINT flag and the Watchdog interrupt request will be generated.
1	1	Watchdog reset mode: operate with the Watchdog interrupt and WDRESET enabled. When this mode is selected, a watchdog counter underflow will reset the microcontroller. Although the Watchdog interrupt is also enabled in this case (WDEN = 1) it will not be recognized since the watchdog reset will clear the WDINT flag.

6.2 Watchdog Timer Constant register (WDTC - 0x4000 4004)

The WDTC register determines the time-out value. Every time a feed sequence occurs the WDTC content is reloaded in to the Watchdog timer. It's a 32-bit register with 8 LSB set to 1 on reset. Writing values below 0xFF will cause 0x0000 00FF to be loaded to the WDTC. Thus the minimum time-out interval is $T_{WDCLK} \times 256 \times 4$.

Table 199. Watchdog Constant register (WDTC - address 0x4000 4004) bit description

Bit	Symbol	Description	Reset Value
31:0	Count	Watchdog time-out interval.	0x0000 00FF

6.3 Watchdog Feed register (WDFEED - 0x4000 4008)

Writing 0xAA followed by 0x55 to this register will reload the Watchdog timer with the WDTC value. This operation will also start the Watchdog if it is enabled via the WDMOD register. Setting the WDEN bit in the WDMOD register is not sufficient to enable the Watchdog. A valid feed sequence must be completed after setting WDEN before the Watchdog is capable of generating a reset. Until then, the Watchdog will ignore feed errors. After writing 0xAA to WDFEED, access to any Watchdog register other than writing 0x55 to WDFEED causes an immediate reset/interrupt when the Watchdog is enabled. The reset will be generated during the second PCLK following an incorrect access to a Watchdog register during a feed sequence.

Interrupts should be disabled during the feed sequence. An abort condition will occur if an interrupt happens during the feed sequence.

Table 200. Watchdog Feed register (WDFEED - address 0x4000 4008) bit description

Bit	Symbol	Description	Reset Value
7:0	Feed	Feed value should be 0xAA followed by 0x55.	NA
31:8	-	reserved	-

6.4 Watchdog Timer Value register (WDTV - 0x4000 400C)

The WDTV register is used to read the current value of Watchdog timer.

When reading the value of the 32-bit timer, the lock and synchronization procedure takes up to 6 WDCLK cycles plus 6 PCLK cycles, so the value of WDTV is older than the actual value of the timer when it's being read by the CPU.

Table 201. Watchdog Timer Value register (WDTV - address 0x4000 000C) bit description

Bit	Symbol	Description	Reset Value
31:0	Count	Counter timer value.	0x0000 00FF

7. Block diagram

The block diagram of the Watchdog is shown below in the [Figure 14–48](#). The synchronization logic (PCLK/WDCLK) is not shown in the block diagram.

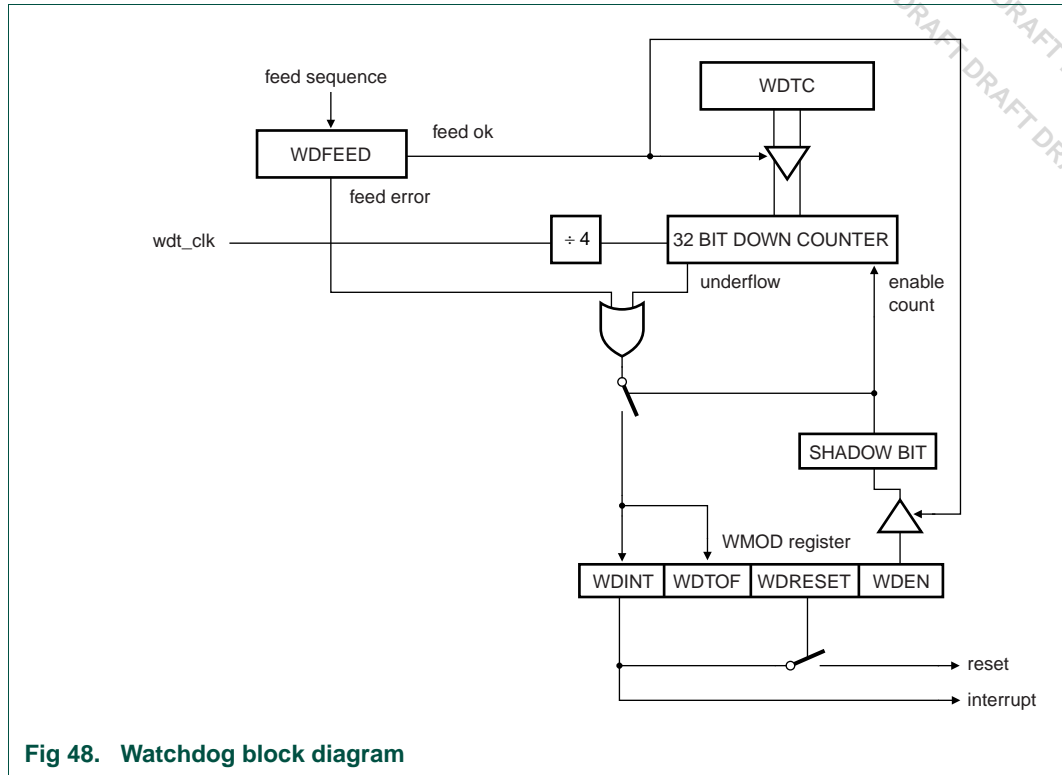


Fig 48. Watchdog block diagram

1. How to read this chapter

The system tick timer (SysTick timer) is part of the ARM Cortex-M0 core and is identical for all LPC111x parts.

2. Features

- Times intervals of 10 milliseconds.
- Uses dedicated exception vector.
- Clocked internally by a dedicated system tick timer clock.

3. Description

The SysTick timer is an integral part of the Cortex-M0. The SysTick timer is intended to generate a fixed 10 millisecond interrupt for use by an operating system or other system management software.

Since the SysTick timer is a part of the Cortex-M0, it facilitates porting of software by providing a standard timer that is available on Cortex-M0 based devices.

Refer to the *Cortex-M0 User Guide* for details.

4. Operation

The SysTick timer is a 24-bit timer that counts down to zero and generates an interrupt. The intent is to provide a fixed 10 millisecond time interval between interrupts. The SysTick timer is clocked from the CPU clock. 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 <td>.

The block diagram of the SysTick timer is shown below in the [Figure 15–49](#).

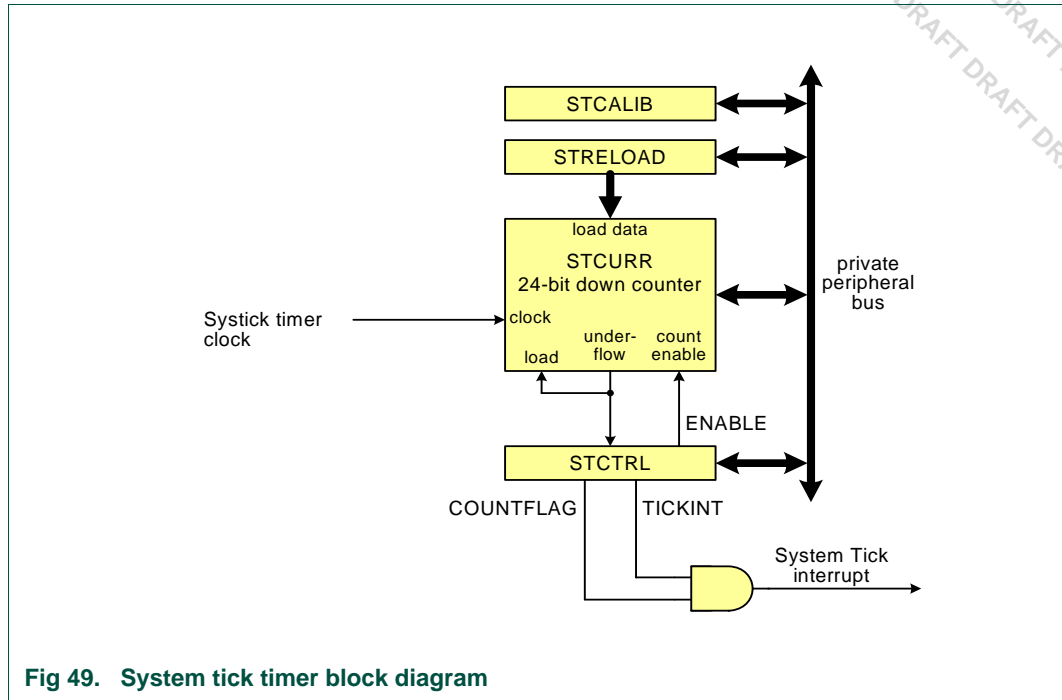


Fig 49. System tick timer block diagram

5. Register description

Table 202. SysTick timer register map (base address 0xE000 E000)

Name	Access	Address offset	Description	Reset value ^[1]
STCTRL	R/W	0x010	System Timer Control and status register	0x4
STRELOAD	R/W	0x014	System Timer Reload value register	0
STCURRE	R/W	0x018	System Timer Current value register	0
STCALIB	R/W	0x01C	System Timer Calibration value register	<td>

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

5.1 System Timer Control and status register (STCTRL - 0xE000 E010)

The STCTRL register contains control information for the SysTick timer, and provides a status flag.

Table 203. 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	-	Reserved	1

Table 203. System Timer Control and status register (STCTRL - 0xE000 E010) bit description

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

5.2 System Timer Reload value register (STRELOAD - 0xE000 E014)

The STRELOAD register is set to the value that will be loaded into the SysTick timer whenever it counts down to zero. This register is loaded by software as part of timer initialization. The 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 204. 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

5.3 System Timer Current value register (STCURRE - 0xE000 E018)

The STCURRE register returns the current count from the System Tick counter when it is read by software.

Table 205. System Timer Current value register (STCURRE - 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

5.4 System Timer Calibration value register (STCALIB - 0xE000 E01C)

<td>

Table 206. System Timer Calibration value register (STCALIB - 0xE000 E01C) bit description

Bit	Symbol	Value	Description	Reset value
23:0	TENMS	<td>		<td>

Table 206. System Timer Calibration value register (STCALIB - 0xE000 E01C) bit description

Bit	Symbol	Value	Description	Reset value
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		<tbd>	0
31	NOREF		<tbd>	0

6. Example timer calculations

The following examples illustrate selecting SysTick timer values for different system configurations. All of the examples calculate an interrupt interval of 10 milliseconds, as the SysTick timer is intended to be used.

<tbd>

1. How to read this chapter

The ADC block is identical for all LPC111x parts.

2. Features

- 10-bit successive approximation Analog-to-Digital Converter (ADC).
- Input multiplexing among 8 pins.
- Power-down mode.
- Measurement range 0 to 3 V (typical). Do not exceed the $V_{DD(3V3)}$ voltage level.
- 10-bit conversion time $\geq 2.44 \mu\text{s}$.
- Burst conversion mode for single or multiple inputs.
- Optional conversion on transition on input pin or Timer Match signal.
- Individual result registers for each A/D channel to reduce interrupt overhead.

3. Pin description

[Table 16–207](#) gives a brief summary of the ADC related pins.

Table 207. ADC pin description

Pin	Type	Description
AD[7:0]	Input	Analog Inputs. The A/D converter cell can measure the voltage on any of these input signals. . Remark: While the pins are 5 V tolerant in digital mode, the maximum input voltage must not exceed $V_{DD(3V3)}$ (3.3 V typical) when the pins are configured as analog inputs.
$V_{DD(3V3)}$	Input	V_{REF} ; Reference voltage.

The ADC function must be selected via the IOCON registers in order to get accurate voltage readings on the monitored pin. For a pin hosting an ADC input, it is not possible to have a digital function selected and yet get valid ADC readings. An inside circuit disconnects ADC hardware from the associated pin whenever a digital function is selected on that pin.

4. Clocking and power control

The peripheral clock to the ADC (PCLK) and to the programmable ADC clock divider (see [Section 16–5.1](#)) is provided by the system clock (see [Figure 3–3](#)). This clock can be disabled through bit 13 in the AHBCLKCTRL register ([Section 3–4.14](#)) for power savings.

The ADC can be powered down at run-time using the PDRUNCFG register ([Section 3–4.35](#)).

Basic clocking for the A/D converters is determined by the APB clock (PCLK). A programmable divider is included in each converter to scale this clock to the 4.5 MHz (max) clock needed by the successive approximation process. An accurate conversion requires 11 clock cycles.

5. Register description

The ADC contains registers organized as shown in [Table 16–208](#).

Table 208. Register overview: ADC (base address 0x4001 C000)

Name	Access	Address offset	Description	Reset Value ^[1]
AD0CR	R/W	0x000	A/D Control Register. The AD0CR register must be written to select the operating mode before A/D conversion can occur.	0x0000 0001
AD0GDR	R/W	0x004	A/D Global Data Register. Contains the result of the most recent A/D conversion.	NA
-	-	0x008	Reserved.	-
AD0INTEN	R/W	0x00C	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.	0x0000 0100
AD0DR0	R/W	0x010	A/D Channel 0 Data Register. This register contains the result of the most recent conversion completed on channel 0	NA
AD0DR1	R/W	0x014	A/D Channel 1 Data Register. This register contains the result of the most recent conversion completed on channel 1.	NA
AD0DR2	R/W	0x018	A/D Channel 2 Data Register. This register contains the result of the most recent conversion completed on channel 2.	NA
AD0DR3	R/W	0x01C	A/D Channel 3 Data Register. This register contains the result of the most recent conversion completed on channel 3.	NA
AD0DR4	R/W	0x020	A/D Channel 4 Data Register. This register contains the result of the most recent conversion completed on channel 4.	NA
AD0DR5	R/W	0x024	A/D Channel 5 Data Register. This register contains the result of the most recent conversion completed on channel 5.	NA
AD0DR6	R/W	0x028	A/D Channel 6 Data Register. This register contains the result of the most recent conversion completed on channel 6.	NA
AD0DR7	R/W	0x02C	A/D Channel 7 Data Register. This register contains the result of the most recent conversion completed on channel 7.	NA
AD0STAT	RO	0x030	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 flag.	0

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

5.1 A/D Control Register (AD0CR - 0x4001 C000)

The A/D Control Register provides bits to select A/D channels to be converted, A/D timing, A/D modes, and the A/D start trigger.

Table 209: A/D Control Register (AD0CR - address 0x4001 C000) bit description

Bit	Symbol	Value	Description	Reset Value
7:0	SEL		Selects which of the AD7:0 pins is (are) to be sampled and converted. Bit 0 selects Pin AD0, bit 1 selects pin AD1,..., and bit 7 selects pin AD7. In software-controlled mode (BURST = 0), only one channel can be selected, i.e. only one of these bits should be 1. In hardware scan mode (BURST = 1), any numbers of channels can be selected, i.e any or all bits can be set to 1. If all bits are set to 0, channel 0 is selected automatically (SEL = 0x01).	0x01
15:8	CLKDIV		The APB clock (PCLK) is divided by CLKDIV +1 to produce the clock for the ADC, which should be less than or equal to 4.5 MHz. Typically, software should program the smallest value in this field that yields a clock of 4.5 MHz or slightly less, but in certain cases (such as a high-impedance analog source) a slower clock may be desirable.	0
16	BURST	0	Software-controlled mode: Conversions are software-controlled and require 11 clocks.	0
		1	Hardware scan mode: The AD converter does repeated conversions at the rate selected by the CLKS field, scanning (if necessary) through the pins selected by 1s in the SEL field. The first conversion after the start corresponds to the least-significant bit set to 1 in the SEL field, then the next higher bits (pins) set to 1 are scanned if applicable. Repeated conversions can be terminated by clearing this bit, but the conversion in progress when this bit is cleared will be completed. Important: START bits must be 000 when BURST = 1 or conversions will not start.	
19:17	CLKS		This field selects the number of clocks used for each conversion in Burst mode, and the number of bits of accuracy of the result in the LS bits of ADDR, between 11 clocks (10 bits) and 4 clocks (3 bits).	000
		000	11 clocks / 10 bits	
		001	10 clocks / 9 bits	
		010	9 clocks / 8 bits	
		011	8 clocks / 7 bits	
		100	7 clocks / 6 bits	
		101	6 clocks / 5 bits	
		110	5 clocks / 4 bits	
		111	4 clocks / 3 bits	
23:20	-		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 PIO0_2/SSEL/CT16B0_CAP0.	
		011	Start conversion when the edge selected by bit 27 occurs on PIO1_5/DIR/CT32B0_CAP0.	
		100	Start conversion when the edge selected by bit 27 occurs on CT32B0_MAT0 ^[1] .	
		101	Start conversion when the edge selected by bit 27 occurs on CT32B0_MAT1 ^[1] .	
		110	Start conversion when the edge selected by bit 27 occurs on CT16B0_MAT0 ^[1] .	
		111	Start conversion when the edge selected by bit 27 occurs on CT16B0_MAT1 ^[1] .	

Table 209: A/D Control Register (AD0CR - address 0x4001 C000) bit description

Bit	Symbol	Value	Description	Reset Value
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

[1] Note that this does not require that the timer match function appear on a device pin.

5.2 A/D Global Data Register (AD0GDR - 0x4001 C004)

The A/D Global Data Register contains the result of the most recent A/D conversion. This includes the data, DONE, and Overrun flags, and the number of the A/D channel to which the data relates.

Table 210: A/D Global Data Register (AD0GDR - address 0x4001 C004) bit description

Bit	Symbol	Description	Reset Value
5:0	Unused	These bits always read as zeroes. They provide compatible expansion room for future, higher-resolution A/D converters.	0
15:6	V/V _{REF}	When DONE is 1, this field contains a binary fraction representing the voltage on the ADn pin selected by the SEL field, divided by the voltage on the V _{DD(3V3)} pin. Zero in the field indicates that the voltage on the ADn pin was less than, equal to, or close to that on V _{SS} , while 0x3FF indicates that the voltage on ADn was close to, equal to, or greater than that on V _{REF} .	X
23:16	Unused	These bits always read as zeroes. They allow accumulation of successive A/D values without AND-masking, for at least 256 values without overflow into the CHN field.	0
26:24	CHN	These bits contain the channel from which the LS bits were converted.	X
29:27	Unused	These bits always read as zeroes. They could be used for expansion of the CHN field in future compatible A/D converters that can convert more channels.	0
30	OVERUN	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 LS bits. In non-FIFO operation, 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

5.3 A/D Status Register (AD0STAT - 0x4001 C030)

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 211: A/D Status Register (AD0STAT - address 0x4001 C030) bit description

Bit	Symbol	Description	Reset Value
7:0	Done7:0	These bits mirror the DONE status flags that appear in the result register for each A/D channel.	0
15:8	Overrun7:0	These bits mirror the OVERRUN status flags that appear in the result register for each A/D channel. Reading ADSTAT allows checking the status of all A/D channels simultaneously.	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	Unused	Unused, always 0.	0

5.4 A/D Interrupt Enable Register (AD0INTEN - 0x4001 C00C)

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 212: A/D Interrupt Enable Register (AD0INTEN - address 0x4001 C00C) bit description

Bit	Symbol	Description	Reset Value
7:0	ADINTEN 7:0	These bits allow control over which A/D channels generate interrupts for conversion completion. When bit 0 is one, completion of a conversion on A/D channel 0 will generate an interrupt, when bit 1 is one, completion of a conversion on A/D channel 1 will generate an interrupt, etc.	0x00
8	ADGINTEN	When 1, enables the global DONE flag in ADDR to generate an interrupt. When 0, only the individual A/D channels enabled by ADINTEN 7:0 will generate interrupts.	1
31:9	Unused	Unused, always 0.	0

5.5 A/D Data Registers (AD0DR0 to AD0DR7 - 0x4001 C010 to 0x4001 C02C)

The A/D Data Register hold the result when an A/D conversion is complete, and also include the flags that indicate when a conversion has been completed and when a conversion overrun has occurred.

Table 213: A/D Data Registers (AD0DR0 to AD0DR7 - addresses 0x4001 C010 to 0x4001 C02C) bit description

Bit	Symbol	Description	Reset Value
5:0	Unused	Unused, always 0. These bits always read as zeroes. They provide compatible expansion room for future, higher-resolution ADCs.	0
15:6	V/V _{REF}	When DONE is 1, this field contains a binary fraction representing the voltage on the ADn pin, divided by the voltage on the V _{REF} pin. Zero in the field indicates that the voltage on the ADn pin was less than, equal to, or close to that on V _{REF} , while 0x3FF indicates that the voltage on AD input was close to, equal to, or greater than that on V _{REF} .	NA
29:16	Unused	These bits always read as zeroes. They allow accumulation of successive A/D values without AND-masking, for at least 256 values without overflow into the CHN field.	0
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 LS 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.	0

6. Operation

6.1 Hardware-triggered conversion

If the BURST bit in the ADCR0 is 0 and the START field contains 010-111, the A/D converter will start a conversion when a transition occurs on a selected pin or timer match signal.

6.2 Interrupts

An interrupt is requested to the interrupt controller when the ADINT bit in the ADSTAT register is 1. The ADINT bit is one when any of the DONE bits of A/D channels that are enabled for interrupts (via the ADINTEN register) are one. Software can use the Interrupt Enable bit in the interrupt controller that corresponds to the ADC to control whether this results in an interrupt. The result register for an A/D channel that is generating an interrupt must be read in order to clear the corresponding DONE flag.

6.3 Accuracy vs. digital receiver

While the A/D converter can be used to measure the voltage on any ADC input pin, regardless of the pin's setting in the IOCON block, selecting the ADC in the IOCON registers function improves the conversion accuracy by disabling the pin's digital receiver (see also [Section 7-3.4](#)).

1. How to read this chapter

See [Table 17–214](#) for different flash configurations.

Table 214. LPC111x flash configurations

Type number	Flash
LPC1111	8 kB
LPC1112	16 kB
LPC1113	32 kB

Remark: For configuring flash memory access times, see [Section 17–10](#).

2. Boot loader

The boot loader controls initial operation after reset, and also provides the means to accomplish programming of the flash memory. This could be initial programming of a blank device, erasure and re-programming of a previously programmed device, or programming of the flash memory by the application program in a running system.

3. Features

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

4. Applications

The boot loader provides both In-System and In-Application programming interfaces for programming the on-chip flash memory.

5. Description

The boot loader code is executed every time the part is powered on or reset. The loader can execute the ISP command handler or the user application code. A LOW level after reset at the PIO0_1 pin is considered as an external hardware request to start the ISP command handler. Assuming that power supply pins are on their nominal levels when the rising edge on RESET pin is generated, it may take up to 3 ms before PIO0_1 is sampled and the decision on whether to continue with user code or ISP handler is made. If PIO0_1 is sampled low and the watchdog overflow flag is set, the external hardware request to start the ISP command handler is ignored. If there is no request for the ISP command

handler execution (PIO0_1 is sampled HIGH after reset), a search is made for a valid user program. If a valid user program is found then the execution control is transferred to it. If a valid user program is not found, the auto-baud routine is invoked.

Pin PIO0_1 that is used as hardware request for ISP requires special attention. Since PIO0_1 is in high impedance mode after reset, it is important that the user provides external hardware (a pull-up resistor or other device) to put the pin in a defined state. Otherwise unintended entry into ISP mode may occur.

5.1 Memory map after any reset

The boot block is 16 kB in size. The boot block is located in the memory region starting from the address 0x1FFF 0000. The boot loader is designed to run from this memory area, but both the ISP and IAP software use parts of the on-chip RAM. The RAM usage is described later in this chapter. The interrupt vectors residing in the boot block of the on-chip flash memory also become active after reset, i.e., the bottom 512 bytes of the boot block are also visible in the memory region starting from the address 0x0000 0000.

5.1.1 Criterion for Valid User Code

Criterion for valid user code: The reserved Cortex-M0 exception vector location 7 (offset 0x 0000 001C in the vector table) should contain the 2's complement of the check-sum of table entries 0 through 6. This causes the checksum of the first 8 table entries to be 0. The boot loader code checksums the first 8 locations in sector 0 of the flash. If the result is 0, then execution control is transferred to the user code.

If the signature is not valid, the auto-baud routine synchronizes with the host via serial port 0. The host should send a '?' (0x3F) as a synchronization character and wait for a response. The host side serial port settings should be 8 data bits, 1 stop bit and no parity. The auto-baud routine measures the bit time of the received synchronization character in terms of its own frequency and programs the baud rate generator of the serial port. It also sends an ASCII string ("Synchronized<CR><LF>") to the Host. In response to this host should send the same string ("Synchronized<CR><LF>"). The auto-baud routine looks at the received characters to verify synchronization. If synchronization is verified then "OK<CR><LF>" string is sent to the host. Host should respond by sending the crystal frequency (in kHz) at which the part is running. For example, if the part is running at 10 MHz, the response from the host should be "10000<CR><LF>". "OK<CR><LF>" string is sent to the host after receiving the crystal frequency. If synchronization is not verified then the auto-baud routine waits again for a synchronization character. For auto-baud to work correctly in case of user invoked ISP, the CCLK frequency should be greater than or equal to 10 MHz.

Once the crystal frequency is received the part is initialized and the ISP command handler is invoked. For safety reasons an "Unlock" command is required before executing the commands resulting in flash erase/write operations and the "Go" command. The rest of the commands can be executed without the unlock command. The Unlock command is required to be executed once per ISP session. The Unlock command is explained in [Section 17-7 "ISP commands" on page 222](#).

5.2 Communication protocol

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

5.2.1 ISP command format

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

5.2.2 ISP response format

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

5.2.3 ISP data format

The data stream is in UU-encode format. The UU-encode algorithm converts 3 bytes of binary data in to 4 bytes of printable ASCII character set. It is more efficient than Hex format which converts 1 byte of binary data in to 2 bytes of ASCII hex. The sender should send the check-sum after transmitting 20 UU-encoded lines. The length of any UU-encoded line should not exceed 61 characters(bytes) i.e. it can hold 45 data bytes. The receiver should compare it with the check-sum of the received bytes. If the check-sum matches then the receiver should respond with "OK<CR><LF>" to continue further transmission. If the check-sum does not match the receiver should respond with "RESEND<CR><LF>". In response the sender should retransmit the bytes.

A description of UU-encode is available at the wotsit webpage.

5.2.4 ISP flow control

A software XON/XOFF flow control scheme is used to prevent data loss due to buffer overrun. When the data arrives rapidly, the ASCII control character DC3 (stop) is sent to stop the flow of data. Data flow is resumed by sending the ASCII control character DC1 (start). The host should also support the same flow control scheme.

5.2.5 ISP command abort

Commands can be aborted by sending the ASCII control character "ESC". This feature is not documented as a command under "ISP Commands" section. Once the escape code is received the ISP command handler waits for a new command.

5.2.6 Interrupts during ISP

The boot block interrupt vectors located in the boot block of the flash are active after any reset.

5.2.7 Interrupts during IAP

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

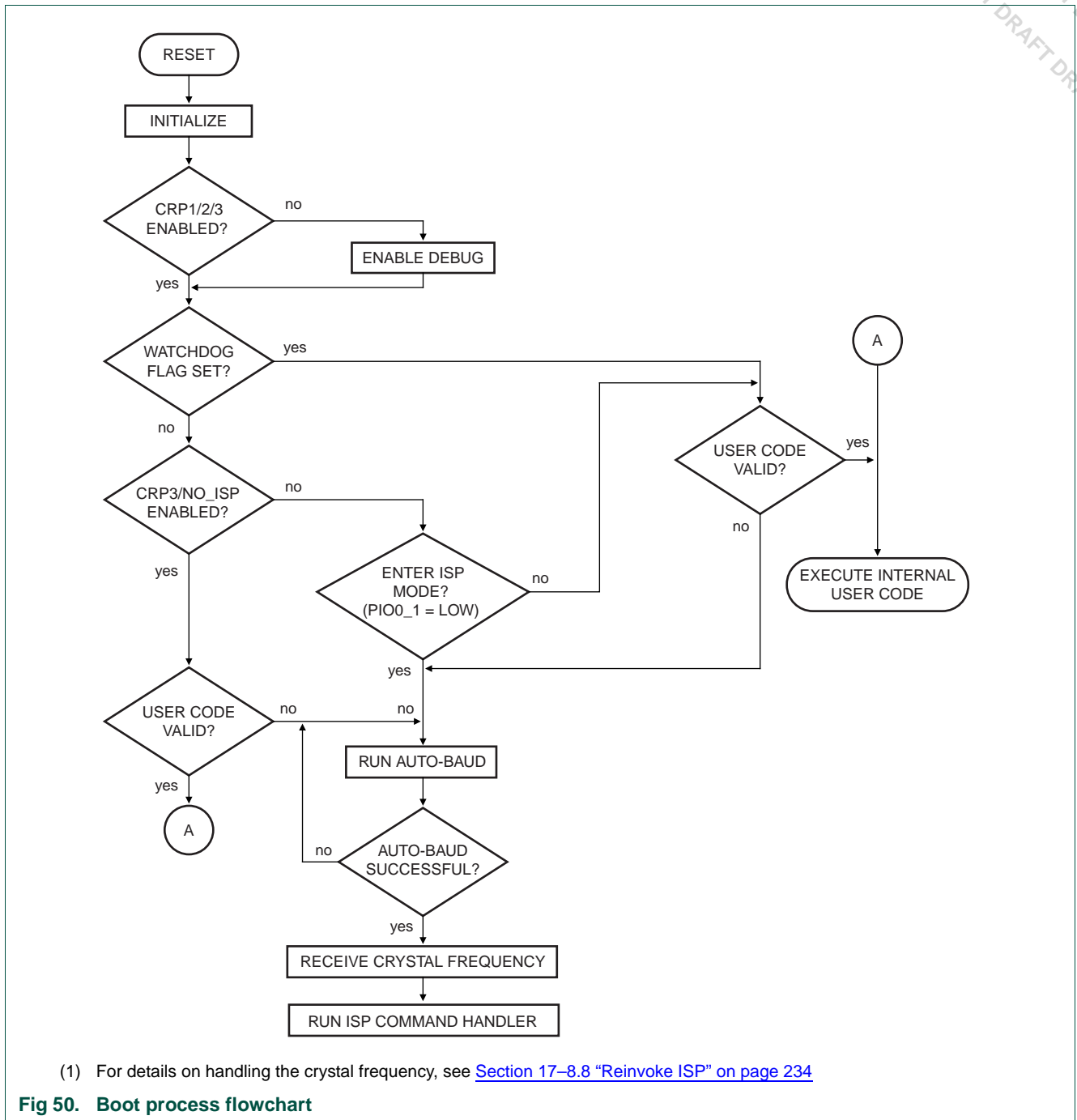
5.2.8 RAM used by ISP command handler

ISP commands use on-chip RAM from 0x1000 017C to 0x1000 025B. The user could use this area, but the contents may be lost upon reset. Flash programming commands use the top 32 bytes of on-chip RAM. The stack is located at RAM top – 32. The maximum stack usage is 256 bytes and it grows downwards.

5.2.9 RAM used by IAP command handler

Flash programming commands use the top 32 bytes of on-chip RAM. The maximum stack usage in the user allocated stack space is 128 bytes and it grows downwards.

5.3 Boot process flowchart



5.4 Sector numbers

Some IAP and ISP commands operate on "sectors" and specify sector numbers. The following table shows the correspondence between sector numbers and memory addresses for LPC111x devices.

Table 215. Sectors in a LPC111x device

Sector number	Sector size [kB]	Address range
0	4	0X0000 0000 - 0X0000 0FFF
1	4	0X0000 1000 - 0X0000 1FFF
2	4	0X0000 2000 - 0X0000 2FFF
3	4	0X0000 3000 - 0X0000 3FFF
4	4	0X0000 4000 - 0X0000 4FFF
5	4	0X0000 5000 - 0X0000 5FFF
6	4	0X0000 6000 - 0X0000 6FFF
7	4	0X0000 7000 - 0X0000 7FFF

6. Code Read Protection (CRP)

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

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

Table 216. Code Read Protection options

Name	Pattern programmed in 0x0000 02FC	Description
CRP1	0x12345678	<p>Access to chip via the JTAG pins is disabled. This mode allows partial flash update using the following ISP commands and restrictions:</p> <ul style="list-style-type: none"> • Write to RAM command cannot access RAM below 0x1000 0300. • Copy RAM to flash command can not write to Sector 0. • Erase command can erase Sector 0 only when all sectors are selected for erase. • Compare command is disabled. • Read Memory command is disabled. <p>This mode is useful when CRP is required and flash field updates are needed but all sectors can not be erased. Since compare command is disabled in case of partial updates the secondary loader should implement checksum mechanism to verify the integrity of the flash.</p>
CRP2	0x87654321	<p>Access to chip via the JTAG pins is disabled. The following ISP commands are disabled:</p> <ul style="list-style-type: none"> • Read Memory • Write to RAM • Go • Copy RAM to flash • Compare <p>When CRP2 is enabled the ISP erase command only allows erasure of all user sectors.</p>
CRP3	0x43218765	<p>Access to chip via the JTAG pins is disabled. ISP entry by pulling PIO0_1 LOW is disabled if a valid user code is present in flash sector 0. This mode effectively disables ISP override using PIO0_1 pin. It is up to the user's application to provide a flash update mechanism using IAP calls or call reinvoke ISP command to enable flash update via UART.</p> <p>Caution: If CRP3 is selected, no future factory testing can be performed on the device.</p>

Table 217. Code Read Protection hardware/software interaction

CRP option	User Code Valid	PIO0_1 pin at reset	JTAG enabled	LPC111x enters ISP mode	partial flash update in ISP mode
No	No	x	Yes	Yes	Yes
No	Yes	High	Yes	No	NA
No	Yes	Low	Yes	Yes	Yes
CRP1	Yes	High	No	No	NA
CRP1	Yes	Low	No	Yes	Yes
CRP2	Yes	High	No	No	NA
CRP2	Yes	Low	No	Yes	No
CRP3	Yes	x	No	No	NA
CRP1	No	x	No	Yes	Yes
CRP2	No	x	No	Yes	No
CRP3	No	x	No	Yes	No

Table 218. ISP commands allowed for different CRP levels

ISP command	CRP1	CRP2	CRP3 (no entry in ISP mode allowed)
Unlock	yes	yes	n/a
Set Baud Rate	yes	yes	n/a
Echo	yes	yes	n/a
Write to RAM	yes; above 0x1000 0300 only	no	n/a
Read Memory	no	no	n/a
Prepare sector(s) for write operation	yes	yes	n/a
Copy RAM to flash	yes; not to sector 0	no	n/a
Go	no	no	n/a
Erase sector(s)	yes; sector 0 can only be erased when all sectors are erased.	yes; all sectors only	n/a
Blank check sector(s)	no	no	n/a
Read Part ID	yes	yes	n/a
Read Boot code version	yes	yes	n/a
Compare	no	no	n/a
ReadUID	yes	yes	n/a

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

6.1 ISP entry protection

In addition to the three CRP modes, the user can prevent the sampling of pin `PIO0_1` for entering ISP mode and thereby release pin `PIO0_1` for other uses. This is called the `NO_ISP` mode. The `NO_ISP` mode can be entered by programming the pattern `0x4E69 7370` at location `0x0000 02FC`.

7. ISP commands

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

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

Table 219. ISP command summary

ISP Command	Usage	Described in
Unlock	U <Unlock Code>	Table 17-220
Set Baud Rate	B <Baud Rate> <stop bit>	Table 17-221
Echo	A <setting>	Table 17-222
Write to RAM	W <start address> <number of bytes>	Table 17-223
Read Memory	R <address> <number of bytes>	Table 17-224
Prepare sector(s) for write operation	P <start sector number> <end sector number>	Table 17-225
Copy RAM to flash	C <Flash address> <RAM address> <number of bytes>	Table 17-226
Go	G <address> <Mode>	Table 17-227
Erase sector(s)	E <start sector number> <end sector number>	Table 17-228
Blank check sector(s)	I <start sector number> <end sector number>	Table 17-229
Read Part ID	J	Table 17-230
Read Boot code version	K	Table 17-232
Compare	M <address1> <address2> <number of bytes>	Table 17-233
ReadUID	N	Table 17-234

7.1 Unlock <Unlock code>

Table 220. ISP Unlock command

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

7.2 Set Baud Rate <Baud Rate> <stop bit>

Table 221. ISP Set Baud Rate command

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

7.3 Echo <setting>

Table 222. ISP Echo command

Command	A
Input	Setting: ON = 1 OFF = 0
Return Code	CMD_SUCCESS PARAM_ERROR
Description	The default setting for echo command is ON. When ON the ISP command handler sends the received serial data back to the host.
Example	"A 0<CR><LF>" turns echo off.

7.4 Write to RAM <start address> <number of bytes>

The host should send the data only after receiving the CMD_SUCCESS return code. The host should send the check-sum after transmitting 20 UU-encoded lines. The checksum is generated by adding raw data (before UU-encoding) bytes and is reset after transmitting 20 UU-encoded lines. The length of any UU-encoded line should not exceed 61 characters(bytes) i.e. it can hold 45 data bytes. When the data fits in less than 20 UU-encoded lines then the check-sum should be of the actual number of bytes sent. The ISP command handler compares it with the check-sum of the received bytes. If the check-sum matches, the ISP command handler responds with "OK<CR><LF>" to continue further transmission. If the check-sum does not match, the ISP command handler responds with "RESEND<CR><LF>". In response the host should retransmit the bytes.

Table 223. ISP Write to RAM command

Command	W
Input	Start Address: RAM address where data bytes are to be written. This address should be a word boundary. Number of Bytes: Number of bytes to be written. Count should be a multiple of 4
Return Code	CMD_SUCCESS ADDR_ERROR (Address not on word boundary) ADDR_NOT_MAPPED COUNT_ERROR (Byte count is not multiple of 4) PARAM_ERROR CODE_READ_PROTECTION_ENABLED
Description	This command is used to download data to RAM. Data should be in UU-encoded format. This command is blocked when code read protection is enabled.
Example	"W 268436224 4<CR><LF>" writes 4 bytes of data to address 0x1000 0300.

7.5 Read Memory <address> <no. of bytes>

The data stream is followed by the command success return code. The check-sum is sent after transmitting 20 UU-encoded lines. The checksum is generated by adding raw data (before UU-encoding) bytes and is reset after transmitting 20 UU-encoded lines. The length of any UU-encoded line should not exceed 61 characters(bytes) i.e. it can hold 45 data bytes. When the data fits in less than 20 UU-encoded lines then the check-sum is of actual number of bytes sent. The host should compare it with the checksum of the received bytes. If the check-sum matches then the host should respond with

"OK<CR><LF>" to continue further transmission. If the check-sum does not match then the host should respond with "RESEND<CR><LF>". In response the ISP command handler sends the data again.

Table 224. ISP Read Memory command

Command	R
Input	<p>Start Address: Address from where data bytes are to be read. This address should be a word boundary.</p> <p>Number of Bytes: Number of bytes to be read. Count should be a multiple of 4.</p>
Return Code	<p>CMD_SUCCESS followed by <actual data (UU-encoded)> ADDR_ERROR (Address not on word boundary) ADDR_NOT_MAPPED COUNT_ERROR (Byte count is not a multiple of 4) PARAM_ERROR CODE_READ_PROTECTION_ENABLED</p>
Description	<p>This command is used to read data from RAM or flash memory. This command is blocked when code read protection is enabled.</p>
Example	<p>"R 268435456 4<CR><LF>" reads 4 bytes of data from address 0x1000 0000.</p>

7.6 Prepare sector(s) for write operation <start sector number> <end sector number>

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

Table 225. ISP Prepare sector(s) for write operation command

Command	P
Input	<p>Start Sector Number</p> <p>End Sector Number: Should be greater than or equal to start sector number.</p>
Return Code	<p>CMD_SUCCESS BUSY INVALID_SECTOR PARAM_ERROR</p>
Description	<p>This command must be executed before executing "Copy RAM to flash" or "Erase Sector(s)" command. Successful execution of the "Copy RAM to flash" or "Erase Sector(s)" command causes relevant sectors to be protected again. The boot block can not be prepared by this command. To prepare a single sector use the same "Start" and "End" sector numbers.</p>
Example	<p>"P 0 0<CR><LF>" prepares the flash sector 0.</p>

7.7 Copy RAM to flash <Flash address> <RAM address> <no of bytes>

Table 226. ISP Copy command

Command	C
Input	<p>Flash Address(DST): Destination flash address where data bytes are to be written. The destination address should be a 256 byte boundary.</p> <p>RAM Address(SRC): Source RAM address from where data bytes are to be read.</p> <p>Number of Bytes: Number of bytes to be written. Should be 256 512 1024 4096.</p>
Return Code	<p>CMD_SUCCESS </p> <p>SRC_ADDR_ERROR (Address not on word boundary) </p> <p>DST_ADDR_ERROR (Address not on correct boundary) </p> <p>SRC_ADDR_NOT_MAPPED </p> <p>DST_ADDR_NOT_MAPPED </p> <p>COUNT_ERROR (Byte count is not 256 512 1024 4096) </p> <p>SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION </p> <p>BUSY </p> <p>CMD_LOCKED </p> <p>PARAM_ERROR </p> <p>CODE_READ_PROTECTION_ENABLED</p>
Description	<p>This command is used to program the flash memory. The "Prepare Sector(s) for Write Operation" command should precede this command. The affected sectors are automatically protected again once the copy command is successfully executed. The boot block cannot be written by this command. This command is blocked when code read protection is enabled.</p>
Example	<p>"C 0 268467504 512<CR><LF>" copies 512 bytes from the RAM address 0x1000 0800 to the flash address 0.</p>

7.8 Go <address> <mode>

Table 227. ISP Go command

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

7.9 Erase sector(s) <start sector number> <end sector number>

Table 228. ISP Erase sector command

Command	E
Input	Start Sector Number End Sector Number: Should be greater than or equal to start sector number.
Return Code	CMD_SUCCESS BUSY INVALID_SECTOR SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION CMD_LOCKED PARAM_ERROR CODE_READ_PROTECTION_ENABLED
Description	This command is used to erase one or more sector(s) of on-chip flash memory. The boot block can not be erased using this command. This command only allows erasure of all user sectors when the code read protection is enabled.
Example	"E 2 3<CR><LF>" erases the flash sectors 2 and 3.

7.10 Blank check sector(s) <sector number> <end sector number>

Table 229. ISP Blank check sector command

Command	I
Input	Start Sector Number: End Sector Number: Should be greater than or equal to start sector number.
Return Code	CMD_SUCCESS SECTOR_NOT_BLANK (followed by <Offset of the first non blank word location> <Contents of non blank word location>) INVALID_SECTOR PARAM_ERROR
Description	This command is used to blank check one or more sectors of on-chip flash memory. Blank check on sector 0 always fails as first 64 bytes are re-mapped to flash boot block.
Example	"I 2 3<CR><LF>" blank checks the flash sectors 2 and 3.

7.11 Read Part Identification number

Table 230. ISP Read Part Identification command

Command	J
Input	None.
Return Code	CMD_SUCCESS followed by part identification number in ASCII (see Table 17-231 "LPC111x part identification numbers").
Description	This command is used to read the part identification number.

Table 231. LPC111x part identification numbers

Device	ASCII/dec coding	Hex coding
LPC1111	<td>	<td>
LPC1113	<td>	<td>

7.12 Read Boot code version number

Table 232. ISP Read Boot Code version number command

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

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

Table 233. ISP Compare command

Command	M
Input	<p>Address1 (DST): Starting flash or RAM address of data bytes to be compared. This address should be a word boundary.</p> <p>Address2 (SRC): Starting flash or RAM address of data bytes to be compared. This address should be a word boundary.</p> <p>Number of Bytes: Number of bytes to be compared; should be a multiple of 4.</p>
Return Code	CMD_SUCCESS (Source and destination data are equal) COMPARE_ERROR (Followed by the offset of first mismatch) COUNT_ERROR (Byte count is not a multiple of 4) ADDR_ERROR ADDR_NOT_MAPPED PARAM_ERROR
Description	This command is used to compare the memory contents at two locations. Compare result may not be correct when source or destination address contains any of the first 512 bytes starting from address zero. First 512 bytes are re-mapped to boot ROM
Example	"M 8192 268468224 4<CR><LF>" compares 4 bytes from the RAM address 0x1000 8000 to the 4 bytes from the flash address 0x2000.

7.14 ReadUID

Table 234. ReadUID command

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

7.15 ISP Return Codes

Table 235. ISP Return Codes Summary

Return Code	Mnemonic	Description
0	CMD_SUCCESS	Command is executed successfully. Sent by ISP handler only when command given by the host has been completely and successfully executed.
1	INVALID_COMMAND	Invalid command.
2	SRC_ADDR_ERROR	Source address is not on word boundary.
3	DST_ADDR_ERROR	Destination address is not on a correct boundary.
4	SRC_ADDR_NOT_MAPPED	Source address is not mapped in the memory map. Count value is taken in to consideration where applicable.
5	DST_ADDR_NOT_MAPPED	Destination address is not mapped in the memory map. Count value is taken in to consideration where applicable.
6	COUNT_ERROR	Byte count is not multiple of 4 or is not a permitted value.
7	INVALID_SECTOR	Sector number is invalid or end sector number is greater than start sector number.
8	SECTOR_NOT_BLANK	Sector is not blank.
9	SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION	Command to prepare sector for write operation was not executed.
10	COMPARE_ERROR	Source and destination data not equal.
11	BUSY	Flash programming hardware interface is busy.
12	PARAM_ERROR	Insufficient number of parameters or invalid parameter.
13	ADDR_ERROR	Address is not on word boundary.
14	ADDR_NOT_MAPPED	Address is not mapped in the memory map. Count value is taken in to consideration where applicable.
15	CMD_LOCKED	Command is locked.
16	INVALID_CODE	Unlock code is invalid.
17	INVALID_BAUD_RATE	Invalid baud rate setting.
18	INVALID_STOP_BIT	Invalid stop bit setting.
19	CODE_READ_PROTECTION_ENABLED	Code read protection enabled.

8. IAP commands

For in application programming the IAP routine should be called with a word pointer in register r0 pointing to memory (RAM) containing command code and parameters. Result of the IAP command is returned in the result table pointed to by register r1. The user can reuse the command table for result by passing the same pointer in registers r0 and r1. The parameter table should be big enough to hold all the results in case if number of results are more than number of parameters. Parameter passing is illustrated in the [Figure 17–51](#). The number of parameters and results vary according to the IAP command. The maximum number of parameters is 5, passed to the "Copy RAM to FLASH"

command. The maximum number of results is 4, returned by the "ReadUID" command. The command handler sends the status code INVALID_COMMAND when an undefined command is received. The IAP routine resides at 0x1FFF1FF0 location and it is thumb code.

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

Define the IAP location entry point. Since the 0th bit of the IAP location is set there will be a change to Thumb instruction set when the program counter branches to this address.

```
#define IAP_LOCATION 0x1fff1ff1
```

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

```
unsigned long command[5];  
unsigned long result[4];
```

or

```
unsigned long * command;  
unsigned long * result;  
command=(unsigned long *) 0x.....  
result= (unsigned long *) 0x.....
```

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

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

Setting function pointer:

```
iap_entry=(IAP) IAP_LOCATION;
```

Whenever you wish to call IAP you could use the following statement.

```
iap_entry (command, result);
```

As per the ARM specification (The ARM Thumb Procedure Call Standard SWS ESPC 0002 A-05) up to 4 parameters can be passed in the r0, r1, r2 and r3 registers respectively. Additional parameters are passed on the stack. Up to 4 parameters can be returned in the r0, r1, r2 and r3 registers respectively. Additional parameters are returned indirectly via memory. Some of the IAP calls require more than 4 parameters. If the ARM suggested scheme is used for the parameter passing/returning then it might create problems due to difference in the C compiler implementation from different vendors. The suggested parameter passing scheme reduces such risk.

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

Table 236. IAP Command Summary

IAP Command	Command Code	Described in
Prepare sector(s) for write operation	50 ₁₀	Table 17–237
Copy RAM to flash	51 ₁₀	Table 17–238
Erase sector(s)	52 ₁₀	Table 17–239
Blank check sector(s)	53 ₁₀	Table 17–240
Read Part ID	54 ₁₀	Table 17–241
Read Boot code version	55 ₁₀	Table 17–242
Compare	56 ₁₀	Table 17–243
Reinvoke ISP	57 ₁₀	Table 17–244
Read UID	58 ₁₀	Table 17–245

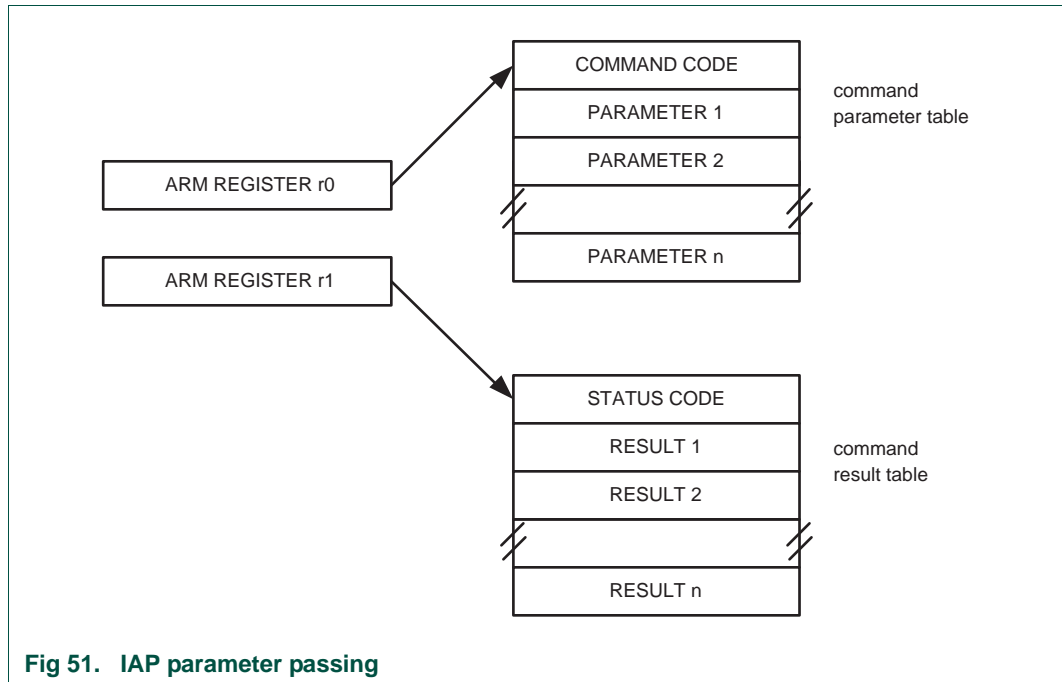


Fig 51. IAP parameter passing

8.1 Prepare sector(s) for write operation

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

Table 237. IAP Prepare sector(s) for write operation command

Command	Prepare sector(s) for write operation
Input	<p>Command code: 50₁₀</p> <p>Param0: Start Sector Number</p> <p>Param1: End Sector Number (should be greater than or equal to start sector number).</p>

Table 237. IAP Prepare sector(s) for write operation command

Command	Prepare sector(s) for write operation
Return Code	CMD_SUCCESS BUSY INVALID_SECTOR
Result	None
Description	This command must be executed before executing "Copy RAM to flash" or "Erase Sector(s)" command. Successful execution of the "Copy RAM to flash" or "Erase Sector(s)" command causes relevant sectors to be protected again. The boot sector can not be prepared by this command. To prepare a single sector use the same "Start" and "End" sector numbers.

8.2 Copy RAM to flash

Table 238. IAP Copy RAM to flash command

Command	Copy RAM to flash
Input	Command code: 51 ₁₀ Param0(DST): Destination flash address where data bytes are to be written. This address should be a 256 byte boundary. Param1(SRC): Source RAM address from which data bytes are to be read. This address should be a word boundary. Param2: Number of bytes to be written. Should be 256 512 1024 4096. Param3: System Clock Frequency (CCLK) in kHz.
Return Code	CMD_SUCCESS SRC_ADDR_ERROR (Address not a word boundary) DST_ADDR_ERROR (Address not on correct boundary) SRC_ADDR_NOT_MAPPED DST_ADDR_NOT_MAPPED COUNT_ERROR (Byte count is not 256 512 1024 4096) SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION BUSY
Result	None
Description	This command is used to program the flash memory. The affected sectors should be prepared first by calling "Prepare Sector for Write Operation" command. The affected sectors are automatically protected again once the copy command is successfully executed. The boot sector can not be written by this command.

8.3 Erase Sector(s)

Table 239. IAP Erase Sector(s) command

Command	Erase Sector(s)
Input	<p>Command code: 52₁₀</p> <p>Param0: Start Sector Number</p> <p>Param1: End Sector Number (should be greater than or equal to start sector number).</p> <p>Param2: System Clock Frequency (CCLK) in kHz.</p>
Return Code	<p>CMD_SUCCESS </p> <p>BUSY </p> <p>SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION </p> <p>INVALID_SECTOR</p>
Result	None
Description	This command is used to erase a sector or multiple sectors of on-chip flash memory. The boot sector can not be erased by this command. To erase a single sector use the same "Start" and "End" sector numbers.

8.4 Blank check sector(s)

Table 240. IAP Blank check sector(s) command

Command	Blank check sector(s)
Input	<p>Command code: 53₁₀</p> <p>Param0: Start Sector Number</p> <p>Param1: End Sector Number (should be greater than or equal to start sector number).</p>
Return Code	<p>CMD_SUCCESS </p> <p>BUSY </p> <p>SECTOR_NOT_BLANK </p> <p>INVALID_SECTOR</p>
Result	<p>Result0: Offset of the first non blank word location if the Status Code is SECTOR_NOT_BLANK.</p> <p>Result1: Contents of non blank word location.</p>
Description	This command is used to blank check a sector or multiple sectors of on-chip flash memory. To blank check a single sector use the same "Start" and "End" sector numbers.

8.5 Read Part Identification number

Table 241. IAP Read Part Identification command

Command	Read part identification number
Input	<p>Command code: 54₁₀</p> <p>Parameters: None</p>
Return Code	CMD_SUCCESS
Result	Result0: Part Identification Number.
Description	This command is used to read the part identification number.

8.6 Read Boot code version number

Table 242. IAP Read Boot Code version number command

Command	Read boot code version number
Input	Command code: 55 ₁₀ Parameters: None
Return Code	CMD_SUCCESS
Result	Result0: 2 bytes of boot code version number in ASCII format. It is to be interpreted as <byte1(Major)>.<byte0(Minor)>
Description	This command is used to read the boot code version number.

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

Table 243. IAP Compare command

Command	Compare
Input	Command code: 56 ₁₀ Param0(DST): Starting flash or RAM address of data bytes to be compared. This address should be a word boundary. Param1(SRC): Starting flash or RAM address of data bytes to be compared. This address should be a word boundary. Param2: Number of bytes to be compared; should be a multiple of 4.
Return Code	CMD_SUCCESS COMPARE_ERROR COUNT_ERROR (Byte count is not a multiple of 4) ADDR_ERROR ADDR_NOT_MAPPED
Result	Result0: Offset of the first mismatch if the Status Code is COMPARE_ERROR.
Description	This command is used to compare the memory contents at two locations. The result may not be correct when the source or destination includes any of the first 512 bytes starting from address zero. The first 512 bytes can be re-mapped to RAM.

8.8 Reinvoke ISP

Table 244. Reinvoke ISP

Command	Compare
Input	Command code: 57 ₁₀
Return Code	None
Result	None.
Description	This command is used to invoke the bootloader in ISP mode. It maps boot vectors, sets PCLK = CCLK, configures UART pins RXD and TXD, resets counter/timer CT32B1 and resets the U0FDR (see Table 9-130). This command may be used when a valid user program is present in the internal flash memory and the PIO0_1 pin is not accessible to force the ISP mode.

8.9 ReadUID

Table 245. IAP ReadUID command

Command	Compare
Input	Command code: 58 ₁₀
Return Code	CMD_SUCCESS
Result	Result0: The first 32-bit word (at the lowest address). Result1: The second 32-bit word. Result2: The third 32-bit word. Result3: The fourth 32-bit word.
Description	This command is used to read the unique ID.

8.10 IAP Status Codes

Table 246. IAP Status Codes Summary

Status Code	Mnemonic	Description
0	CMD_SUCCESS	Command is executed successfully.
1	INVALID_COMMAND	Invalid command.
2	SRC_ADDR_ERROR	Source address is not on a word boundary.
3	DST_ADDR_ERROR	Destination address is not on a correct boundary.
4	SRC_ADDR_NOT_MAPPED	Source address is not mapped in the memory map. Count value is taken in to consideration where applicable.
5	DST_ADDR_NOT_MAPPED	Destination address is not mapped in the memory map. Count value is taken in to consideration where applicable.
6	COUNT_ERROR	Byte count is not multiple of 4 or is not a permitted value.
7	INVALID_SECTOR	Sector number is invalid.
8	SECTOR_NOT_BLANK	Sector is not blank.
9	SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION	Command to prepare sector for write operation was not executed.
10	COMPARE_ERROR	Source and destination data is not same.
11	BUSY	flash programming hardware interface is busy.

9. Serial Wire Debug (SWD) flash programming interface

Debug tools can write parts of the flash image to RAM and then execute the IAP call "Copy RAM to flash" repeatedly with proper offset.

10. Flash memory access

Depending on the system clock frequency, access to the flash memory can be configured with various access times by writing to the FLASHCFG register at address 0x4003 C010.

Remark: Improper setting of this register may result in incorrect operation of the LPC111x flash memory.

Table 247. Flash configuration register (FLASHCFG, address 0x4003 C010) bit description

Bit	Symbol	Value	Description	Reset value
1:0	FLASHTIM		Flash memory access time. FLASHTIM +1 is equal to the number of system clocks used for flash access.	10
		00	1 system clock flash access time (for system clock frequencies of up to 20 MHz).	
		01	2 system clocks flash access time (for system clock frequencies of up to 40 MHz).	
		10	3 system clocks flash access time (for system clock frequencies of up to 50 MHz).	
		11	Reserved.	
31:2	-	-	Reserved. User software must not change the value of these bits. Bits 31:2 must be written back exactly as read.	<tbd>

1. How to read this chapter

The debug functionality is identical for all LPC111x parts.

2. Features

- Supports ARM Serial Wire Debug mode.
- Direct debug access to all memories, registers, and peripherals.
- No target resources are required for the debugging session.
- Trace port provides CPU instruction trace capability. Output via a Serial Wire Viewer.
- <tb> breakpoints. <tb> instruction breakpoints that can also be used to remap instruction addresses for code patches. Two data comparators that can be used to remap addresses for patches to literal values.
- <tb> data watchpoints that can also be used as trace triggers.
- Instrumentation Trace Macrocell allows additional software controlled trace.

3. Introduction

Debug and trace functions are integrated into the ARM Cortex-M0. Serial wire debug and trace functions are supported. The ARM Cortex-M0 is configured to support up to four breakpoints and two watchpoints.

4. Description

Debugging with the LPC111x uses the Serial Wire Debug mode.

5. Pin description

The tables below indicate the various pin functions related to debug and trace. Some of these functions share pins with other functions which therefore may not be used at the same time.

Table 248. JTAG pin description

Pin Name	Type	Description
TCK	Input	JTAG Test Clock. This pin is the clock for debug logic when in the JTAG debug mode.
TMS	Input	JTAG Test Mode Select. The TMS pin selects the next state in the TAP state machine.

Table 248. JTAG pin description

Pin Name	Type	Description
TDI	Input	JTAG Test Data In. This is the serial data input for the shift register.
TDO	Output	JTAG Test Data Output. This is the serial data output from the shift register. Data is shifted out of the device on the negative edge of the TCK signal.
$\overline{\text{TRST}}$	Input	JTAG Test Reset. The $\overline{\text{TRST}}$ pin can be used to reset the test logic within the debug logic.

Table 249. Serial Wire Debug pin description

Pin Name	Type	Description
SWCLK	Input	Serial Wire Clock. This pin is the clock for debug logic when in the Serial Wire Debug mode (SWDCLK). In JTAG mode this is the TCK pin.
SWDIO	Input / Output	Serial wire debug data input/output. The SWDIO pin is used by an external debug tool to communicate with and control the LPC111x.

6. Debug Notes

Important: The user should be aware of certain limitations during debugging. The most important is that, due to limitations of the ARM Cortex-M0 integration, the LPC111x cannot wake up in the usual manner from Deep-sleep mode. It is recommended not to use this mode during debug.

Another issue is that debug mode changes the way in which reduced power modes work internal to the ARM Cortex-M0 CPU, and this ripples through the entire system. These differences mean that power measurements should not be made while debugging, the results will be higher than during normal operation in an application.

During a debugging session, the System Tick Timer is automatically stopped whenever the CPU is stopped. Other peripherals are not affected.

1. Abbreviations

Table 250. Abbreviations

Acronym	Description
---------	-------------

2. Legal information

2.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

2.2 Disclaimers

General — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

2.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

<Name> — is a trademark of NXP B.V.

3. Tables

Table 1. Ordering information	4	Table 26. WDT clock source update enable register (WDTCLKUEN, address 0x4004 80D4) bit description	21
Table 2. Ordering options for UM10398	4	Table 27. WDT clock divider register (WDTCLKDIV, address 0x4004 80D8) bit description	21
Table 3. LPC111x memory configuration	6	Table 28. CLKOUT clock source select register (CLKOUTCLKSEL, address 0x4004 80E0) bit description	21
Table 4. Pin summary.	8	Table 29. CLKOUT clock source update enable register (CLKOUTUEN, address 0x4004 80E4) bit description	22
Table 5. Register overview: system control block (base address 0x4004 8000)	9	Table 30. CLKOUT clock divider registers (CLKOUTCLKDIV, address 0x4004 80E8) bit description	22
Table 6. System memory remap register (SYSMEMREMAP, address 0x4004 8000) bit description	11	Table 31. POR captured PIO status registers 0 (PIOPORCAP0, address 0x4004 8100) bit description	22
Table 7. Peripheral reset control register (PRESETCTRL, address 0x4004 8004) bit description.	11	Table 32. POR captured PIO status registers 1 (PIOPORCAP1, address 0x4004 8104) bit description	23
Table 8. System PLL control register (SYSPLLCTRL, address 0x4004 8008) bit description	12	Table 33. BOD control register (BODCTRL, address 0x4004 8150) bit description.	23
Table 9. System PLL status register (SYSPLLSTAT, address 0x4004 800C) bit description	13	Table 34. System tick timer calibration register (SYSTCKCAL, address 0x4004 8158) bit description	24
Table 10. System oscillator control register (SYSOSCCTRL, address 0x4004 8020) bit description.	13	Table 35. Start logic edge control register 0 (STARTAPRP0, address 0x4004 8200) bit description	24
Table 11. Watchdog oscillator control register (WDTOSCCTRL, address 0x4004 8024) bit description	14	Table 36. Start logic signal enable register 0 (STARTERP0, address 0x4004 8204) bit description	24
Table 12. Internal resonant crystal control register (IRCCTRL, address 0x4004 8028) bit description 14		Table 37. Start logic reset register 0 (STARTRSRP0CLR, address 0x4004 8208) bit description	25
Table 13. System reset status register (SYSRSTSTAT, address 0x4004 8030) bit description.	15	Table 38. Start logic status register 0 (STARTSRP0, address 0x4004 820C) bit description	25
Table 14. System PLL clock source select register (SYSPLLCLKSEL, address 0x4004 8040) bit description	15	Table 39. Deep-sleep configuration register (PDSLEEP_CFG, address 0x4004 8230) bit description	26
Table 15. System PLL clock source update enable register (SYSPLLKEN, address 0x4004 8044) bit description	16	Table 40. Wake-up configuration register (PDAWAKECFG, address 0x4004 8234) bit description	27
Table 16. Main clock source select register (MAINCLKSEL, address 0x4004 8070) bit description.	16	Table 41. Power-down configuration register (PDRUNCFG, address 0x4004 8238) bit description	28
Table 17. Main clock source update enable register (MAINCLKUEN, address 0x4004 8074) bit description	16	Table 42. Device ID register (DEVICE_ID, address 0x4004 83F4) bit description	29
Table 18. System AHB clock divider register (SYSAHBCLKDIV, address 0x4004 8078) bit description	17	Table 43. LPC111x power and clock control options	30
Table 19. System AHB clock control register (AHBCLKCTRL, address 0x4004 8080) bit description	17	Table 44. PLL operating modes.	34
Table 20. SSP0 clock divider register (SSP0CLKDIV, address 0x4004 8094) bit description.	19	Table 45. PLL frequency parameters.	35
Table 21. UART clock divider register (UARTCLKDIV, address 0x4004 8098) bit description.	19	Table 46. Register overview: PMU (base address 0x4003 8000)	38
Table 22. SSP1 clock divider register (SSP1CLKDIV, address 0x4004 809C) bit description	19	Table 47. Power control register (PCON, address 0x4003 8000) bit description	38
Table 23. TRACECLKDIV clock divider register (TRACECLKDIV, address 0x4004 80AC) bit description	20	Table 48. General purpose registers 0 to 3 (GPREG0 - GPREG3, address 0x4003 8004 to 0x4003 8010) bit description	39
Table 24. SYSTICK clock divider register (SYSTICKCLKDIV, address 0x4004 80B0) bit description	20	Table 49. General purpose register 4 (GPREG4, address 0x4003 8014) bit description	39
Table 25. WDT clock source select register (WDTCLKSEL, address 0x4004 80D0) bit description	20	Table 50. Connection of interrupt sources to the Vectored Interrupt Controller.	41

Table 51. GPIO configuration	43	address 0x4004 402C) bit description	60
Table 52. Register overview: GPIO (base address port 0: 0x5000 0000; port 1: 0x5001 0000, port 2: 0x5002 0000; port 3: 0x5003 0000)	43	Table 74. IOCON_PIO0_4 register (IOCON_PIO0_4 address 0x4004 4030) bit description	61
Table 53. GPIO0DATA register (GPIO0DATA, address 0x5000 0000 to 0x5000 3FFC; GPIO1DATA, address 0x5001 0000 to 0x5001 3FFC; GPIO2DATA, address 0x5002 0000 to 0x5002 3FFC; GPIO3DATA, address 0x5003 0000 to 0x5003 3FFC) bit description	44	Table 75. IOCON_PIO0_5 register (IOCON_PIO0_5 address 0x4004 4034) bit description	61
Table 54. GPIO0DIR register (GPIO0DIR, address 0x5000 8000 to GPIO3DIR, address 0x5003 8000) bit description	44	Table 76. IOCON_PIO1_9 register (IOCON_PIO1_9 address 0x4004 4038) bit description	62
Table 55. GPIO0IS register (GPIO0IS, address 0x5000 8004 to GPIO3IS, address 0x5003 8004) bit description	45	Table 77. IOCON_PIO3_4 register (IOCON_PIO3_4, address 0x4004 403C) bit description	62
Table 56. GPIO0IBE register (GPIO0IBE, address 0x5000 8008 to GPIO3IBE, address 0x5003 8008) bit description	45	Table 78. IOCON_PIO2_4 register (IOCON_PIO2_4, address 0x4004 4040) bit description	63
Table 57. GPIO0IEV register (GPIO0IEV, address 0x5000 800C to GPIO3IEV, address 0x5003 800C) bit description	45	Table 79. IOCON_PIO2_5 register (IOCON_PIO2_5, address 0x4004 4044) bit description	63
Table 58. GPIO0IE register (GPIO0IE, address 0x5000 8010 to GPIO3IE, address 0x5003 8010) bit description	46	Table 80. IOCON_PIO3_5 register (IOCON_PIO3_5, address 0x4004 4048) bit description	64
Table 59. GPIO0IRS register (GPIO0IRS, address 0x5000 8014 to GPIO3IRS, address 0x5003 8014) bit description	46	Table 81. IOCON_PIO0_6 register (IOCON_PIO0_6 address 0x4004 404C) bit description	64
Table 60. GPIO0MIS register (GPIO0MIS, address 0x5000 8018 to GPIO3MIS, address 0x5003 8018) bit description	46	Table 82. IOCON_PIO0_7 register (IOCON_PIO0_7, address 0x4004 4050) bit description	65
Table 61. GPIO0IC register (GPIO0IC, address 0x5000 801C to GPIO3IC, address 0x5003 801C) bit description	47	Table 83. IOCON_PIO2_9 register (IOCON_PIO2_9, address 0x4004 4054) bit description	65
Table 62. Register overview: I/O configuration (base address 0x4004 4000)	51	Table 84. IOCON_PIO2_10 register (IOCON_PIO2_10, address 0x4004 4058) bit description	66
Table 63. I/O configuration registers ordered by port number 53		Table 85. IOCON_PIO2_2 register (IOCON_PIO2_2, address 0x4004 405C) bit description	66
Table 64. IOCON_PIO2_6 register (IOCON_PIO2_6, address 0x4004 4000) bit description	54	Table 86. IOCON_PIO0_8 register (IOCON_PIO0_8, address 0x4004 4060) bit description	67
Table 65. IOCON_PIO2_0 register (IOCON_PIO2_0, address 0x4004 4008) bit description	55	Table 87. IOCON_PIO0_9 register (IOCON_PIO0_9, address 0x4004 4064) bit description	67
Table 66. IOCON_nRESET_PIO0_0 register (IOCON_nRESET_PIO0_0, address 0x4004 400C) bit description.	55	Table 88. IOCON_JTAG_TCK_PIO0_10 register (IOCON_JTAG_TCK_PIO0_10, address 0x4004 4068) bit description	68
Table 67. IOCON_PIO0_1 register (IOCON_PIO0_1, address 0x4004 4010) bit description	56	Table 89. IOCON_PIO1_10 register (IOCON_PIO1_10, address 0x4004 406C) bit description	69
Table 68. IOCON_PIO1_8 register (IOCON_PIO1_8, address 0x4004 4014) bit description	56	Table 90. IOCON_PIO2_11 register (IOCON_PIO2_11, address 0x4004 4070) bit description	69
Table 69. IOCON_PIO0_2 register (IOCON_PIO0_2, address 0x4004 401C) bit description	57	Table 91. IOCON_JTAG_TDI_PIO0_11 register (IOCON_JTAG_TDI_PIO0_11, address 0x4004 4074) bit description	70
Table 70. IOCON_PIO2_7 register (IOCON_PIO2_7, address 0x4004 4020) bit description	57	Table 92. IOCON_JTAG_TMS_PIO1_0 register (IOCON_JTAG_TMS_PIO1_0, address 0x4004 4078) bit description	71
Table 71. IOCON_PIO2_8 register (IOCON_PIO2_8, address 0x4004 4024) bit description	59	Table 93. IOCON_JTAG_TDO_PIO1_1 register (IOCON_JTAG_TDO_PIO1_1, address 0x4004 407C) bit description	72
Table 72. IOCON_PIO2_1 register (IOCON_PIO2_1, address 0x4004 4028) bit description	60	Table 94. IOCON_JTAG_nTRST_PIO1_2 register (IOCON_JTAG_nTRST_PIO1_2, address 0x4004 4080) bit description	73
Table 73. IOCON_PIO0_3 register (IOCON_PIO0_3		Table 95. IOCON_PIO3_0 register (IOCON_PIO3_0, address 0x4004 4084) bit description	73
		Table 96. IOCON_PIO3_1 register (IOCON_PIO3_1, address 0x4004 4088) bit description	74
		Table 97. IOCON_PIO2_3 register (IOCON_PIO2_3, address 0x4004 408C) bit description	74
		Table 98. IOCON_SWDIO_PIO1_3 register (IOCON_SWDIO_PIO1_3, address 0x4004 4090) bit description	75
		Table 99. IOCON_PIO1_4 register (IOCON_PIO1_4,	

address 0x4004 4094) bit description	76	0x4000 800C) bit description	104
Table 100. IOCON_PIO1_11 register (IOCON_PIO1_11		Table 127. UART Line Status Register (UOLSR - address	
address 0x4004 4098) bit description	77	0x4000 8014, Read Only) bit description	105
Table 101. IOCON_PIO3_2 register (IOCON_PIO3_2,		Table 128. UART Modem Status Register (U0MSR - address	
address 0x4004 409C) bit description	77	0x4000 8018) bit description	106
Table 102. IOCON_PIO1_5 register (IOCON_PIO1_5,		Table 129. UART Scratch Pad Register (U0SCR - address	
address 0x4004 40A0) bit description	78	0x4000 8014) bit description	107
Table 103. IOCON_PIO1_6 register (IOCON_PIO1_6,		Table 130. Auto-baud Control Register (U0ACR - address	
address 0x4004 40A4) bit description	78	0x4000 8020) bit description	107
Table 104. IOCON_PIO1_7 register (IOCON_PIO1_7,		Table 131. UART Fractional Divider Register (U0FDR -	
address 0x4004 40A8) bit description	79	address 0x4000 8028) bit description	111
Table 105. IOCON_PIO3_3 register (IOCON_PIO3_3,		Table 132. Fractional Divider setting look-up table	113
address 0x4004 40AC) bit description	79	Table 133. UART Transmit Enable Register (U0TER -	
Table 106. IOCON_SCK location register		address 0x4000 8030) bit description	114
(IOCON_SCK_LOC, address 0x4004 40B0) bit		Table 134. UART RS485 Control register (U0RS485CTRL -	
description	80	address 0x4000 804C) bit description	114
Table 107. IOCON_DSR location register		Table 135. UART RS-485 Address Match register	
(IOCON_DSR_LOC, address 0x4004 40B4) bit		(U0RS485ADRMATCH - address 0x4000 8050)	
description	80	bit description.	115
Table 108. IOCON_DCD location register		Table 136. UART RS-485 Delay value register	
(IOCON_DCD_LOC, address 0x4004 40B8) bit		(U0RS485DLY - address 0x4000 80454) bit	
description	80	description	115
Table 109. IOCON_RI location register (IOCON_RI_LOC,		Table 137. UART FIFO Level register (U0FIFOLVL - address	
address 0x4004 40BC) bit description	81	0x4000 8058, Read Only) bit description	117
Table 110. LPC111x pin configurations	82	Table 138. I ² C-bus pin description.	120
Table 111. LPC1113 pin description table (LQFP48 package)		Table 139. Register overview: I ² C (base address 0x4000	
.	85	0000)	121
Table 112. LPC1113 pin description table (PLCC44		Table 140. I ² C Control Set register (I2CONSET - address	
package)	88	0x4000 0000) bit description	122
Table 113. LPC1111/1112/1113 pin description table		Table 141. I ² C Status register (I2STAT - 0x4000 0004) bit	
(HVQFN33 package)	91	description	124
Table 114. UART pin description	94	Table 142. I ² C Data register (I2DAT - 0x4000 0008) bit	
Table 115. Register overview: UART (base address: 0x4000		description	124
8000)	96	Table 143. I ² C Slave Address registers (I2ADR0 -	
Table 116. UART Receiver Buffer Register (U0RBR -		0x4000 000C) bit description	124
address 0x4000 8000 when DLAB = 0, Read		Table 144. I ² C SCL HIGH duty cycle register (I2SCLH -	
Only) bit description	97	address 0x4000 0010) bit description	124
Table 117. UART Transmitter Holding Register (U0THR -		Table 145. I ² C SCL Low duty cycle register (I2SCLL -	
address 0x4000 8000 when DLAB = 0, Write		0x4000 0014) bit description	125
Only) bit description	97	Table 146. I2SCLL + I2SCLH values for selected I ² C clock	
Table 118. UART Divisor Latch LSB Register (U0DLL -		values.	125
address 0x4000 8000 when DLAB = 1) bit		Table 147. I ² C Control Clear register (I2CONCLR -	
description	98	0x4000 0018) bit description	125
Table 119. UART Divisor Latch MSB Register (U0DLM -		Table 148. I ² C Monitor mode control register (I2CMMCTRL0	
address 0x4000 8004 when DLAB = 1) bit		- 0x4000 001C) bit description	126
description	98	Table 149. I ² C Slave Address registers (I2ADR[0, 1, 2, 3]-	
Table 120. UART Interrupt Enable Register (U0IER - address		0x4000 00[0C, 20, 24, 28]) bit description	128
0x4000 8004 when DLAB = 0) bit description	98	Table 150. I ² C Data buffer register (I2CDATA_BUFFER -	
Table 121. UART Interrupt Identification Register (U0IIR -		0x4000 002C) bit description	128
address 0x4004 8008, Read Only) bit description		Table 151. I ² C Mask registers (I2MASK[0, 1, 2, 3] -	
99		0x4000 00[30, 34, 38, 3C]) bit description	129
Table 122. UART Interrupt Handling	100	Table 152. I2CONSET used to configure Master mode	129
Table 123. UART FIFO Control Register (U0FCR - address		Table 153. I2CONSET used to configure Slave mode	131
0x4000 8008, Write Only) bit description	101	Table 154. Abbreviations used to describe an I ² C operation.	
Table 124. UART0 Modem Control Register (U0MCR -		137	
address 0x4000 8010) bit description.	102	Table 155. I2CONSET used to initialize Master Transmitter	
Table 125. Modem status interrupt generation.	103	mode	137
Table 126. UART Line Control Register (U0LCR - address		Table 156. I2ADR usage in Slave Receiver mode.	138

Table 157. I2CONSET used to initialize Slave Receiver mode 138	address 0x4001 003C) bit description 183
Table 158. Master Transmitter mode 144	Table 183. External match control 183
Table 159. Master Receiver mode 145	Table 184. Count Control Register (TMR16B0CTCR - address 0x4000 C070 and TMR16B1CTCR - address 0x4001 0070) bit description 184
Table 160. Slave Receiver mode 146	Table 185. PWM Control Register (TMR16B0PWMC - address 0x4000 C074 and TMR16B1PWMC - address 0x4001 0074) bit description 185
Table 161. Slave Transmitter mode 148	Table 186. Counter/timer pin description 189
Table 162. Miscellaneous States 150	Table 187. Register overview: 32-bit counter/timer 0 CT32B0 (base address 0x4001 4000) 189
Table 163. SSP pin descriptions 162	Table 188. Register overview: 32-bit counter/timer 1 CT32B1 (base address 0x4001 8000) 190
Table 164. Register overview: SSP0 (base address 0x4004 0000) 163	Table 189. Interrupt Register (TMR32B0IR - address 0x4001 4000 and TMR32B1IR - address 0x4001 8000) bit description 191
Table 165. Register overview: SSP1 (base address 0x4005 8000) 163	Table 190. Timer Control Register (TMR32B0TCR - address 0x4001 4004 and TMR32B1TCR - address 0x4001 8004) bit description 192
Table 166. SSP Control Register 0 (SSP0CR0 - address 0x4004 0000, SSP1CR0 - address 0x4005 8000) bit description 164	Table 191. Match Control Register (TMR32B0MCR - address 0x4001 4014 and TMR32B1MCR - address 0x4001 8014) bit description 192
Table 167. SSP Control Register 1 (SSP0CR1 - address 0x4004 0004, SSP1CR1 - address 0x4005 8004) bit description 165	Table 192. Capture Control Register (TMR32B0CCR - address 0x4001 4028 and TMR32B1CCR - address 0x4001 8028) bit description 194
Table 168. SSP Data Register (SSP0DR - address 0x4004 0008, SSP1DR - address 0x4005 8008) bit description 165	Table 193. External Match Register (TMR32B0EMR - address 0x4001 403C and TMR32B1EMR - address 0x4001 803C) bit description 195
Table 169. SSP Status Register (SSP0SR - address 0x4004 000C, SSP1SR - address 0x4005 800C) bit description 166	Table 194. External match control 195
Table 170. SSP0 Clock Prescale Register (SSP0CPSR - address 0x4004 0010, SSP1CPSR - address 0x4005 8010) bit description 166	Table 195. Count Control Register (TMR32B0CTCR - address 0x4001 4070 and TMR32B1TCR - address 0x4001 8070) bit description 196
Table 171. SSP Interrupt Mask Set/Clear register (SSP0IMSC - address 0x4004 0014, SSP1IMSC - address 0x4005 8014) bit description. 167	Table 196. PWM Control Register (TMR32B0PWMC - 0x4001 4074 and TMR32B1PWMC - 0x4001 8074) bit description. 197
Table 172. SSP Raw Interrupt Status register (SSP0RIS - address 0x4004 0018, SSP1RIS - address 0x4005 8018) bit description 167	Table 197. Register overview: Watchdog timer (base address 0x4000 4000) 201
Table 173. SSP Masked Interrupt Status register (SSP0MIS - address 0x4004 001C, SSP1MIS - address 0x4005 801C) bit description 168	Table 198. Watchdog Mode register (WDMOD - address 0x4000 4000) bit description 202
Table 174. SSP interrupt Clear Register (SSP0ICR - address 0x4004 0020, SSP1ICR - address 0x4005 8020) bit description 168	Table 199. Watchdog operating modes selection 202
Table 175. Counter/timer pin description 177	Table 200. Watchdog Constant register (WDTC - address 0x4000 4004) bit description 203
Table 176. Register overview: 16-bit counter/timer 0 CT16B0 (base address 0x4000 C000) 177	Table 201. Watchdog Feed register (WDFEED - address 0x4000 4008) bit description 203
Table 177. Register overview: 16-bit counter/timer 1 CT16B1 (base address 0x4001 0000) 178	Table 202. Watchdog Timer Value register (WDTV - address 0x4000 000C) bit description 203
Table 178. Interrupt Register (TMR16B0IR - address 0x4000 C000 and TMR16B1IR - address 0x4001 0000) bit description 179	Table 203. SysTick timer register map (base address 0xE000 E000) 206
Table 179. Timer Control Register (TMR16B0TCR - address 0x4000 C004 and TMR16B1TCR - address 0x4001 0004) bit description 180	Table 204. System Timer Control and status register (STCTRL - 0xE000 E010) bit description. 206
Table 180. Match Control Register (TMR16B0MCR - address 0x4000 C014 and TMR16B1MCR - address 0x4001 0014) bit description 180	Table 205. System Timer Reload value register (STRELOAD - 0xE000 E014) bit description. 207
Table 181. Capture Control Register (TMR16B0CCR - address 0x4000 C028 and TMR16B1CCR - address 0x4001 0028) bit description. 182	Table 206. System Timer Current value register (STCURREG - 0xE000 E018) bit description. 207
Table 182. External Match Register (TMR16B0EMR - address 0x4000 C03C and TMR16B1EMR -	Table 207. System Timer Calibration value register (STCALIB - 0xE000 E01C) bit description. 207

C000)210

Table 210:A/D Control Register (AD0CR - address
0x4001 C000) bit description211

Table 211:A/D Global Data Register (AD0GDR - address
0x4001 C004) bit description212

Table 212:A/D Status Register (AD0STAT - address
0x4001 C030) bit description213

Table 213:A/D Interrupt Enable Register (AD0INTEN -
address 0x4001 C00C) bit description213

Table 214:A/D Data Registers (AD0DR0 to AD0DR7 -
addresses 0x4001 C010 to 0x4001 C02C) bit
description214

Table 215.LPC111x flash configurations215

Table 216.Sectors in a LPC111x device220

Table 217.Code Read Protection options221

Table 218.Code Read Protection hardware/software
interaction221

Table 219.ISP commands allowed for different CRP levels .
222

Table 220.ISP command summary223

Table 221.ISP Unlock command223

Table 222.ISP Set Baud Rate command223

Table 223.ISP Echo command224

Table 224.ISP Write to RAM command224

Table 225.ISP Read Memory command225

Table 226.ISP Prepare sector(s) for write operation
command225

Table 227.ISP Copy command226

Table 228.ISP Go command226

Table 229.ISP Erase sector command227

Table 230.ISP Blank check sector command227

Table 231.ISP Read Part Identification command227

Table 232.LPC111x part identification numbers227

Table 233.ISP Read Boot Code version number command .
228

Table 234.ISP Compare command228

Table 235.ReadUID command228

Table 236.ISP Return Codes Summary229

Table 237.IAP Command Summary231

Table 238.IAP Prepare sector(s) for write operation
command231

Table 239.IAP Copy RAM to flash command232

Table 240.IAP Erase Sector(s) command233

Table 241.IAP Blank check sector(s) command233

Table 242.IAP Read Part Identification command233

Table 243.IAP Read Boot Code version number command .
234

Table 244.IAP Compare command234

Table 245.Reinvoke ISP234

Table 246.IAP ReadUID command235

Table 247.IAP Status Codes Summary235

Table 248.Flash configuration register (FLASHCFG, address
0x4003 C010) bit description236

Table 249.JTAG pin description237

Table 250.Serial Wire Debug pin description238

Table 251.Abbreviations239

4. Figures

Fig 1. LPC111x block diagram 5

Fig 2. LPC111x memory map 7

Fig 3. LPC111x CGU block diagram 9

Fig 4. System PLL block diagram 33

Fig 5. Masked write operation to the GPIODATA register 47

Fig 6. Masked read operation 48

Fig 7. Standard I/O pin configuration 50

Fig 8. Pin configuration LQFP48 package 83

Fig 9. Pin configuration PLCC44 package 84

Fig 10. Pin configuration HVQFN 33 package 85

Fig 11. Auto-RTS Functional Timing 103

Fig 12. Auto-CTS Functional Timing 104

Fig 13. Auto-baud a) mode 0 and b) mode 1 waveform 110

Fig 14. Algorithm for setting UART dividers 112

Fig 15. UART block diagram 118

Fig 16. I²C-bus configuration 120

Fig 17. Format in the Master Transmitter mode 130

Fig 18. Format of Master Receiver mode 130

Fig 19. A Master Receiver switches to Master Transmitter after sending Repeated START 131

Fig 20. Format of Slave Receiver mode 131

Fig 21. Format of Slave Transmitter mode 132

Fig 22. I²C serial interface block diagram 133

Fig 23. Arbitration procedure 135

Fig 24. Serial clock synchronization 135

Fig 25. Format and states in the Master Transmitter mode 140

Fig 26. Format and states in the Master Receiver mode 141

Fig 27. Format and states in the Slave Receiver mode 142

Fig 28. Format and states in the Slave Transmitter mode 143

Fig 29. Simultaneous Repeated START conditions from two masters 151

Fig 30. Forced access to a busy I²C-bus 152

Fig 31. Recovering from a bus obstruction caused by a LOW level on SDA 152

Fig 32. Texas Instruments Synchronous Serial Frame Format: a) Single and b) Continuous/back-to-back Two Frames Transfer 169

Fig 33. SPI frame format with CPOL=0 and CPHA=0 (a) Single and b) Continuous Transfer) 170

Fig 34. SPI frame format with CPOL=0 and CPHA=1 171

Fig 35. SPI frame format with CPOL = 1 and CPHA = 0 (a) Single and b) Continuous Transfer) 172

Fig 36. SPI Frame Format with CPOL = 1 and CPHA = 1 173

Fig 37. Microwire frame format (single transfer) 174

Fig 38. Microwire frame format (continuous transfers) 174

Fig 39. Microwire frame format setup and hold details 175

Fig 40. Sample PWM waveforms with a PWM cycle length of 100 (selected by MR3) and MAT3:0 enabled as PWM outputs by the PWCON register 186

Fig 41. A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled 186

Fig 42. A timer cycle in which PR=2, MRx=6, and both interrupt and stop on match are enabled 186

Fig 43. 16-bit counter/timer block diagram 187

Fig 44. Sample PWM waveforms with a PWM cycle length of 100 (selected by MR3) and MAT3:0 enabled as PWM outputs by the PWCON register 198

Fig 45. A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled 198

Fig 46. A timer cycle in which PR=2, MRx=6, and both interrupt and stop on match are enabled 198

Fig 47. 32-bit counter/timer block diagram 199

Fig 48. Watchdog block diagram 204

Fig 49. System tick timer block diagram 206

Fig 50. Boot process flowchart 219

Fig 51. IAP parameter passing 231

5. Contents

Chapter 1: LPC111x Introductory information

1	Introduction	3	3	Ordering information	4
2	Features	3	4	Block diagram	5

Chapter 2: LPC111x Memory map

1	How to read this chapter	6	2	Memory map	6
---	--------------------------------	---	---	------------------	---

Chapter 3: LPC111x System configuration

1	Introduction	8	4.30	Start logic edge control register 0	24
2	Pin description	8	4.31	Start logic signal enable register 0	24
3	Clocking and power control	8	4.32	Start logic reset register 0	25
4	Register description	9	4.33	Start logic status register 0	25
4.1	System memory remap register	11	4.34	Deep-sleep mode configuration register	26
4.2	Peripheral reset control register	11	4.35	Wake-up configuration register	27
4.3	System PLL control register	12	4.36	Power-down configuration register	27
4.4	System PLL status register	12	4.37	Device ID register	29
4.5	System oscillator control register	13	5	Reset	29
4.6	Watchdog oscillator control register	13	6	Brown-out detection	29
4.7	Internal resonant crystal control register	14	7	Power management	30
4.8	System reset status register	15	7.1	Sleep mode	31
4.9	System PLL clock source select register	15	7.2	Deep-sleep mode	31
4.10	System PLL clock source update enable register	16	7.3	Deep power-down mode	31
4.11	Main clock source select register	16	8	Deep-sleep mode	31
4.12	Main clock source update enable register	16	8.1	Entering Deep-sleep mode	31
4.13	System AHB clock divider register	17	8.2	Powering down the 12 MHz IRC oscillator	32
4.14	System AHB clock control register	17	8.3	Start logic	32
4.15	SSP0 clock divider register	19	9	System PLL functional description	32
4.16	UART clock divider register	19	9.1	Lock detector	33
4.17	SSP1 clock divider register	19	9.2	Direct output mode	34
4.18	Trace clock divider register	20	9.3	Power-down control	34
4.19	SYSTICK clock divider register	20	9.4	Operating modes	34
4.20	WDT clock source select register	20	9.5	Divider ratio programming	34
4.21	WDT clock source update enable register	21	9.6	Post divider	34
4.22	WDT clock divider register	21	9.6.1	Feedback divider	34
4.23	CLKOUT clock source select register	21	9.6.2	Changing the divider values	34
4.24	CLKOUT clock source update enable register	22	9.6.3	Frequency selection	35
4.25	CLKOUT clock divider register	22	9.6.4	Mode 1 (Normal mode)	35
4.26	POR captured PIO status register 0	22	9.6.5	Mode 2 (Direct CCO mode)	35
4.27	POR captured PIO status register 1	23		Mode 3 (Power-down mode)	36
4.28	BOD control register	23		Mode 4 (Bypass mode)	36
4.29	System tick counter calibration register	24		Mode 5 (Direct bypass mode)	37

Chapter 4: LPC111x PMU (Power Management Unit)

1	Introduction	38	2.3	General purpose register 4	39
2	Register description	38	3	Functional description	39
2.1	Power control register	38	3.1	Entering Deep power-down mode	39
2.2	General purpose registers 0 to 3	38	3.2	Exiting Deep power-down mode	39

Chapter 5: LPC111x Interrupt controller

1	Introduction	41	2	Features	41
---	--------------------	----	---	----------------	----

3 Interrupt sources 41

Chapter 6: LPC111x General Purpose I/O (GPIO)

1	How to read this chapter	43	3.6	GPIO interrupt mask register	45
2	Introduction	43	3.7	GPIO raw interrupt status register	46
2.1	Features	43	3.8	GPIO masked interrupt status register	46
3	Register description	43	3.9	GPIO interrupt clear register	47
3.1	GPIO data register	44	4	Functional description	47
3.2	GPIO data direction register	44	4.1	Write/read data operation	47
3.3	GPIO interrupt sense register	45		Write operation	47
3.4	GPIO interrupt both edges sense register	45		Read operation	47
3.5	GPIO interrupt event register	45			

Chapter 7: LPC111x I/O configuration

1	How to read this chapter	49	3.4	A/D-mode	51
2	Introduction	49	3.5	I ² C mode	51
3	General description	49	4	Register description	51
3.1	Pin function	50	4.1	I/O configuration registers IOCON_PIO _n	54
3.2	Pin mode	50	4.2	IOCON location registers	79
3.3	Hysteresis	50			

Chapter 8: LPC111x Pin configuration

1	How to read this chapter	82	3	LPC111x Pin description	85
2	LPC111x Pin configuration	83			

Chapter 9: LPC111x Universal Asynchronous Receiver/Transmitter (UART)

1	How to read this chapter	94	5.11	UART Scratch Pad Register (U0SCR - 0x4000 801C)	107
2	Features	94	5.12	UART Auto-baud Control Register (U0ACR - 0x4000 8020)	107
3	Pin description	94	5.13	Auto-baud	108
4	Clocking and power control	95	5.14	Auto-baud modes	109
5	Register description	95	5.15	UART Fractional Divider Register (U0FDR - 0x4000 8028)	110
5.1	UART Receiver Buffer Register (U0RBR - 0x4000 8000, when DLAB = 0, Read Only) ..	97	5.15.1	Baud rate calculation	111
5.2	UART Transmitter Holding Register (U0THR - 0x4000 8000 when DLAB = 0, Write Only) ..	97	5.15.1.1	Example 1: UART_PCLK = 14.7456 MHz, BR = 9600	113
5.3	UART Divisor Latch LSB and MSB Registers (U0DLL - 0x4000 8000 and U0DLM - 0x4000 8004, when DLAB = 1)	97	5.15.1.2	Example 2: UART_PCLK = 12 MHz, BR = 115200	113
5.4	UART Interrupt Enable Register (U0IER - 0x4000 8004, when DLAB = 0)	98	5.16	UART Transmit Enable Register (U0TER - 0x4000 8030)	113
5.5	UART Interrupt Identification Register (U0IIR - 0x4004 8008, Read Only)	99	5.17	UART RS485 Control register (U0RS485CTRL - 0x4000 804C)	114
5.6	UART FIFO Control Register (U0FCR - 0x4000 8008, Write Only)	101	5.18	UART RS-485 Address Match register (U0RS485ADRMATCH - 0x4000 8050)	115
5.7	UART Modem Control Register	101	5.19	UART1 RS-485 Delay value register (U0RS485DLY - 0x4000 8054)	115
5.7.1	Auto-flow control	102	5.20	RS-485/EIA-485 modes of operation	115
5.7.1.1	Auto-RTS	102		RS-485/EIA-485 Normal Multidrop Mode (NMM)	116
5.7.1.2	Auto-CTS	103		RS-485/EIA-485 Auto Address Detection (AAD) mode	116
5.8	UART Line Control Register (U0LCR - 0x4000 800C)	104		RS-485/EIA-485 Auto Direction Control	116
5.9	UART Line Status Register (U0LSR - 0x4000 8014, Read Only)	105		RS485/EIA-485 driver delay time	117
5.10	UART Modem Status Register	106		RS485/EIA-485 output inversion	117

5.21 UART FIFO Level register (U0FIFOLVL - 0x4000 8058, Read Only) 117

6 Architecture 117

Chapter 10: LPC111x I2C-bus interface

1 How to read this chapter 119

2 Features 119

3 Applications 119

4 General description 119

4.1 I²C Fast-mode Plus 120

5 Pin description 120

6 Clocking and power control 121

7 Register description 121

7.1 I²C Control Set register (I2CONSET - 0x4000 0000) 122

7.2 I²C Status register (I2STAT - 0x4000 0004) 124

7.3 I²C Data register (I2DAT - 0x4000 0008) 124

7.4 I²C Slave Address register (I2ADR0- 0x4000 000C) 124

7.5 I²C SCL HIGH duty cycle register and LOW duty cycle register (I2SCLH - 0x4000 0010 and I2SCLL - 0x4000 0014) 124

7.5.1 Selecting the appropriate I²C data rate and duty cycle 125

7.6 I²C Control Clear register (I2CONCLR - 0x4000 0018) 125

7.7 I²C Monitor mode control register (I2CMMCTRL0 - 0x4000 001C) 126

7.7.1 Interrupt in Monitor mode 127

7.7.2 Loss of arbitration in Monitor mode 127

7.8 I²C Slave Address registers (I2ADR[0, 1, 2, 3]- 0x4000 00[0C, 20, 24, 28]) 127

7.9 I²C Data buffer register (I2CDATA_BUFFER - 0x4000 002C) 128

7.10 I²C Mask registers (I2MASK[0, 1, 2, 3] - 0x4000 00[30, 34, 38, 3C]) 128

8 I²C operating modes 129

8.1 Master Transmitter mode 129

8.2 Master Receiver mode 130

8.3 Slave Receiver mode 131

8.4 Slave Transmitter mode 132

9 Functional description 132

9.1 Input filters and output stages 132

9.2 Address Registers, I2ADDR0 to I2ADDR3 134

9.3 Address mask registers, I2MASK0 to I2MASK3 134

9.4 Comparator 134

9.5 Shift register, I2DAT 134

9.6 Arbitration and synchronization logic 134

9.7 Serial clock generator 135

9.8 Timing and control 136

9.9 Control register, I2CONSET and I2CONCLR 136

9.10 Status decoder and status register 136

10 Details of I²C operating modes 136

10.1 Master Transmitter mode 137

10.2 Master Receiver mode 138

10.3 Slave Receiver mode 138

10.4 Slave Transmitter mode 143

10.5 Miscellaneous states 149

10.6 I2STAT = 0xF8 149

10.7 I2STAT = 0x00 149

10.8 Some special cases 150

10.9 Simultaneous Repeated START conditions from two masters 150

10.10 Data transfer after loss of arbitration 150

10.11 Forced access to the I²C-bus 150

10.12 I²C-bus obstructed by a LOW level on SCL or SDA 151

10.13 Bus error 151

10.14 I²C state service routines 152

10.15 Initialization 152

10.16 I²C interrupt service 153

10.17 The state service routines 153

10.18 Adapting state services to an application 153

11 Software example 153

11.1 Initialization routine 153

11.2 Start Master Transmit function 153

11.3 Start Master Receive function 154

11.4 I²C interrupt routine 154

11.5 Non mode specific states 154

11.5.1 State: 0x00 154

11.5.2 Master States 154

11.5.3 State: 0x08 154

11.5.4 State: 0x10 155

11.6 Master Transmitter states 155

11.6.1 State: 0x18 155

11.6.2 State: 0x20 155

11.6.3 State: 0x28 155

11.6.4 State: 0x30 156

11.6.5 State: 0x38 156

11.7 Master Receive states 156

11.7.1 State: 0x40 156

11.7.2 State: 0x48 156

11.7.3 State: 0x50 156

11.7.4 State: 0x58 157

11.8 Slave Receiver states 157

11.8.1 State: 0x60 157

11.8.2 State: 0x68 157

11.8.3 State: 0x70 157

11.8.4 State: 0x78 158

11.8.5 State: 0x80 158

11.8.6 State: 0x88 158

11.8.7 State: 0x90 158

11.8.8 State: 0x98 158

11.8.9 State: 0xA0 159

11.9 Slave Transmitter states 159

11.9.1 State: 0xA8 159

11.9.2 State: 0xB0 159

11.9.3 State: 0xB8 159

11.9.4 State: 0xC0 160

11.9.5 State: 0xC8 160

Chapter 11: LPC111x SSP0/1

1	How to read this chapter	161	6.9	SSP Interrupt Clear Register	168
2	Features	161	7	Functional description	168
3	General description	161	7.1	Texas Instruments synchronous serial frame format	168
4	Pin description	162	7.2	SPI frame format	169
5	Clocking and power control	163	7.2.1	Clock Polarity (CPOL) and Phase (CPHA) control	169
6	Register description	163	7.2.2	SPI format with CPOL=0,CPHA=0.	170
6.1	SSP Control Register 0	164	7.2.3	SPI format with CPOL=0,CPHA=1.	171
6.2	SSP0 Control Register 1	164	7.2.4	SPI format with CPOL = 1,CPHA = 0.	171
6.3	SSP Data Register	165	7.2.5	SPI format with CPOL = 1,CPHA = 1.	173
6.4	SSP Status Register	166	7.3	Semiconductor Microwire frame format	173
6.5	SSP Clock Prescale Register	166	7.3.1	Setup and hold time requirements on CS with respect to SK in Microwire mode	175
6.6	SSP Interrupt Mask Set/Clear Register	166			
6.7	SSP Raw Interrupt Status Register	167			
6.8	SSP Masked Interrupt Status Register	167			

Chapter 12: LPC111x 16-bit counter/timer (CT16B0/1)

1	How to read this chapter	176	7.6	Match Control Register (TMR16B0MCR and TMR16B1MCR)	180
2	Features	176	7.7	Match Registers (TMR16B0MR0/1/2/3 - addresses 0x4000 C018/1C/20/24 and TMR16B1MR0/1/2/3 - addresses 0x4001 0018/1C/20/24)	181
3	Applications	176	7.8	Capture Control Register (TMR16B0CCR and TMR16B1CCR)	182
4	Description	176	7.9	Capture Register (CT16B0CR0 - address 0x4000 C02C and CT16B1CR0 - address 0x4001 002C)	182
5	Pin description	177	7.10	External Match Register (TMR16B0EMR and TMR16B1EMR)	182
6	Clocking and power control	177	7.11	Count Control Register (TMR16B0CTCR and TMR16B1CTCR)	183
7	Register description	177	7.12	PWM Control register (TMR16B0PWMC and TMR16B1PWMC)	184
7.1	Interrupt Register (TMR16B0IR and TMR16B1IR)	179	7.13	Rules for single edge controlled PWM outputs	185
7.2	Timer Control Register (TMR16B0TCR and TMR16B1TCR)	179	8	Example timer operation	186
7.3	Timer Counter (TMR16B0TC - address 0x4000 C008 and TMR16B1TC - address 0x4001 0008)	180	9	Architecture	187
7.4	Prescale Register (TMR16B0PR - address 0x4000 C00C and TMR16B1PR - address 0x4001 000C)	180			
7.5	Prescale Counter register (TMR16B0PC - address 0x4000 C010 and TMR16B1PC - address 0x4001 0010)	180			

Chapter 13: LPC111x 32 bit counter/timer (CT32B0/1)

1	How to read this chapter	188	7.3	Timer Counter (TMR32B0TC - address 0x4001 4008 and TMR32B1TC - address 0x4001 8008)	192
2	Features	188	7.4	Prescale Register (TMR32B0PR - address 0x4001 400C and TMR32B1PR - address 0x4001 800C)	192
3	Applications	188	7.5	Prescale Counter Register (TMR32B0PC - address 0x4001 4010 and TMR32B1PC - address 0x4001 8010)	192
4	Description	188	7.6	Match Control Register (TMR32B0MCR and TMR32B1MCR)	192
5	Pin description	189			
6	Clocking and power control	189			
7	Register description	189			
7.1	Interrupt Register (TMR32B0IR and TMR32B1IR)	191			
7.2	Timer Control Register (TMR32B0TCR and TMR32B1TCR)	191			

Chapter 19: LPC111x Supplementary information

7.7	Match Registers (TMR32B0MR0/1/2/3 - addresses 0x4001 4018/1C/20/24 and TMR32B1MR0/1/2/3 addresses 0x4001 8018/1C/20/24)	193	7.10	External Match Register (TMR32B0EMR and TMR32B1EMR)	194
7.8	Capture Control Register (TMR32B0CCR and TMR32B1CCR)	194	7.11	Count Control Register (TMR32B0CTCR and TMR32B1TCR)	195
7.9	Capture Register (TMR32B0CR0 - address 0x4001 402C and TMR32B1CR0 - address 0x4001 802C)	194	7.12	PWM Control Register (TMR32B0PWMC and TMR32B1PWMC)	196
			7.13	Rules for single edge controlled PWM outputs	197
			8	Example timer operation	198
			9	Architecture	199

Chapter 14: LPC111x WatchDog Timer (WDT)

1	How to read this chapter	200	6.2	Watchdog Timer Constant register (WDTC - 0x4000 4004)	203
2	Features	200	6.3	Watchdog Feed register (WDFEED - 0x4000 4008)	203
3	Applications	200	6.4	Watchdog Timer Value register (WDTV - 0x4000 400C)	203
4	Description	200	7	Block diagram	203
5	Clocking and power control	201			
6	Register description	201			
6.1	Watchdog Mode register (WDMOD - 0x4000 0000)	202			

Chapter 15: LPC111x System tick timer

1	How to read this chapter	205	5.2	System Timer Reload value register (STRELOAD - 0xE000 E014)	207
2	Features	205	5.3	System Timer Current value register (STCURRE - 0xE000 E018)	207
3	Description	205	5.4	System Timer Calibration value register (STCALIB - 0xE000 E01C)	207
4	Operation	205	6	Example timer calculations	208
5	Register description	206			
5.1	System Timer Control and status register (STCTRL - 0xE000 E010)	206			

Chapter 16: LPC111x Analog-to-Digital Converter (ADC)

1	How to read this chapter	209	5.3	A/D Status Register (AD0STAT - 0x4001 C030)	212
2	Features	209	5.4	A/D Interrupt Enable Register (AD0INTEN - 0x4001 C00C)	213
3	Pin description	209	5.5	A/D Data Registers (AD0DR0 to AD0DR7 - 0x4001 C010 to 0x4001 C02C)	213
4	Clocking and power control	209	6	Operation	214
5	Register description	210	6.1	Hardware-triggered conversion	214
5.1	A/D Control Register (AD0CR - 0x4001 C000)	210	6.2	Interrupts	214
5.2	A/D Global Data Register (AD0GDR - 0x4001 C004)	212	6.3	Accuracy vs. digital receiver	214

Chapter 17: LPC111x Flash memory programming firmware

1	How to read this chapter	215	3	Features	215
2	Boot loader	215	4	Applications	215

continued >>

Chapter 19: LPC111x Supplementary information

5	Description	215	7.8	Go <address> <mode>	226
5.1	Memory map after any reset	216	7.9	Erase sector(s) <start sector number> <end sector number>	227
5.1.1	Criterion for Valid User Code	216	7.10	Blank check sector(s) <sector number> <end sector number>	227
5.2	Communication protocol	217	7.11	Read Part Identification number	227
5.2.1	ISP command format	217	7.12	Read Boot code version number	228
5.2.2	ISP response format	217	7.13	Compare <address1> <address2> <no of bytes>	228
5.2.3	ISP data format	217	7.14	ReadUID	228
5.2.4	ISP flow control	217	7.15	ISP Return Codes	229
5.2.5	ISP command abort	217	8	IAP commands	229
5.2.6	Interrupts during ISP	217	8.1	Prepare sector(s) for write operation	231
5.2.7	Interrupts during IAP	217	8.2	Copy RAM to flash	232
5.2.8	RAM used by ISP command handler	218	8.3	Erase Sector(s)	233
5.2.9	RAM used by IAP command handler	218	8.4	Blank check sector(s)	233
5.3	Boot process flowchart	219	8.5	Read Part Identification number	233
5.4	Sector numbers	219	8.6	Read Boot code version number	234
6	Code Read Protection (CRP)	220	8.7	Compare <address1> <address2> <no of bytes>	234
6.1	ISP entry protection	222	8.8	Reinvoke ISP	234
7	ISP commands	222	8.9	ReadUID	235
7.1	Unlock <Unlock code>	223	8.10	IAP Status Codes	235
7.2	Set Baud Rate <Baud Rate> <stop bit>	223	9	Serial Wire Debug (SWD) flash programming interface	235
7.3	Echo <setting>	224	10	Flash memory access	235
7.4	Write to RAM <start address> <number of bytes>	224			
7.5	Read Memory <address> <no. of bytes>	224			
7.6	Prepare sector(s) for write operation <start sector number> <end sector number>	225			
7.7	Copy RAM to flash <Flash address> <RAM address> <no of bytes>	226			

Chapter 18: LPC111x Serial Wire Debug and Trace

1	How to read this chapter	237	4	Description	237
2	Features	237	5	Pin description	237
3	Introduction	237	6	Debug Notes	238

Chapter 19: LPC111x Supplementary information

1	Abbreviations	239	3	Tables	241
2	Legal information	240	4	Figures	246
2.1	Definitions	240	5	Contents	247
2.2	Disclaimers	240			
2.3	Trademarks	240			

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.



© NXP B.V. 2009. All rights reserved.

For more information, please visit: <http://www.nxp.com>
 For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 19 October 2009
 Document identifier: UM10398_0