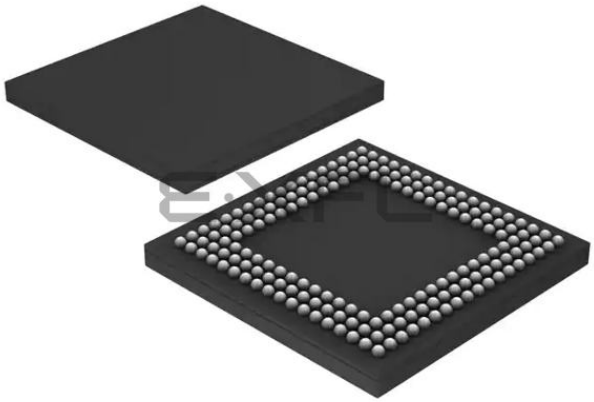

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M3
Core Size	32-Bit Single-Core
Speed	180MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, Microwire, MMC/SD, SPI, SSI, SSP, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, WDT
Number of I/O	49
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	136K x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	A/D 4x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	180-TFBGA
Supplier Device Package	180-TFBGA (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/lpc18s10fet180e

Table 71. Connection of interrupt sources to the NVIC

Interrupt ID	Exception Number	Vector Offset	Function	Flag(s)
28	44	0xB0	I2S0	-
29	45	0xB4	I2S1	-
30	46	0xB8	-	Reserved
31	47	0xBC	-	Reserved
32	48	0xC0	PIN_INT0	GPIO pin interrupt 0
33	49	0xC4	PIN_INT1	GPIO pin interrupt 1
34	50	0xC8	PIN_INT2	GPIO pin interrupt 2
35	51	0xCC	PIN_INT3	GPIO pin interrupt 3
36	52	0xD0	PIN_INT4	GPIO pin interrupt 4
37	53	0xD4	PIN_INT5	GPIO pin interrupt 5
38	54	0xD8	PIN_INT6	GPIO pin interrupt 6
39	55	0xDC	PIN_INT7	GPIO pin interrupt 7
40	56	0xE0	GINT0	GPIO global interrupt 0
41	57	0xE4	GINT1	GPIO global interrupt 1
42	58	0xE8	Event router	Combined interrupt from the event router sources
43	59	0xEC	C_CAN1 interrupt	-
44	60	0xF0	Reserved	-
45	61	0xF4	Reserved	-
46	62	0xF8	ATIMER	Alarm timer interrupt
47	63	0xFC	RTC	Combined RTC and event router/monitor interrupt
48	64	0x100	Reserved	-
49	65	0x104	WWDT	-
50	66	0x108	Reserved	-
51	67	0x10C	C_CAN0	-
52	68	0x110	QEI	-

8.7 Register description

The following table summarizes the registers in the NVIC as implemented in the LPC18xx. The Cortex-M3 User Guide provides a functional description of the NVIC registers.

Table 72. Register overview: NVIC (base address 0xE000 E000)

Name	Access	Address offset	Description	Reset value
ISER0	RW	0x100	Interrupt Set-Enable Register 0. This register allows enabling interrupts and reading back the interrupt enables for specific peripheral functions.	0
ISER1	RW	0x104	Interrupt Set-Enable Register 1. This register allows enabling interrupts and reading back the interrupt enables for specific peripheral functions.	0
ICER0	RW	0x180	Interrupt Clear-Enable Register 0. This register allows disabling interrupts and reading back the interrupt enables for specific peripheral functions.	0

Table 88. CREG0 register (CREG0, address 0x4004 3004) bit description *…continued*

Bit	Symbol	Value	Description	Reset value	Access
13:12	SAMPLECTRL		SAMPLE pin input/output control	0	R/W
		0x0	Reserved		
		0x1	Sample output from the event monitor/recorder.		
		0x2	Output from the event router.		
		0x3	Reserved.		
15:14	WAKEUP0CTRL		WAKEUP0 pin input/output control	0	R/W
		0x0	Input to the event router.		
		0x1	Output from the event router.		
		0x2	Reserved.		
		0x3	Input to the event router.		
17:16	WAKEUP1CTRL		WAKEUP1 pin input/output control	0	R/W
		0x0	Input to event router.		
		0x1	Output from the event router.		
		0x2	Reserved		
		0x3	Input to event router.		
31:18	-		Reserved	-	-

10.4.2 CREG1 control register

Table 89. CREG1 register (CREG1, address 0x4004 3008) bit description

Bit	Symbol	Value	Description	Reset value	Access
8:0	-	-	Reserved	-	-
9	USB0_PHY_PWREN_LP		USB0 PHY power control in low power mode. Set this bit to enable the power to USB0 PHY in low power mode. This enables wake-up using USB0 peripheral in deep-sleep mode.	0	R/W
		0	USB0 PHY power disabled in low power mode.		
		1	USB0 PHY power enabled in low power mode.		
10	USB1_PHY_PWREN_LP		USB1 PHY power control in low power mode. Set this bit to enable the power to USB1 PHY in low power mode. This enables wake-up using USB1 peripheral in deep-sleep mode.	0	R/W
		0	USB1 PHY power disabled in low power mode.		
		1	USB1 PHY power enabled in low power mode.		
31:11	-	-	Reserved	-	-

Remark: The wake-up from deep-sleep using USB0 and USB1 is supported only in flash based devices and is not supported in flashless devices.

12.7.4.3.5 Mode 1d: Normal operating mode with post-divider and with pre-divider

In normal operating mode 1d none of the dividers are bypassed. The operating frequencies are:

Fout = Fcco /(2 x P) = M x Fin /(N x P) Λ (275 MHz ≤Fcco ≤550 MHz, 4 kHz ≤Fin/N ≤150 MHz)

The divider ratios are programmable:

- Pre-divider N (N, 1 to 256)
- Feedback-divider M (M, 1 to 2^{15})
- Post-divider P (P, 1 to 32)

12.7.4.3.6 Mode 3: Power down mode (pd)

In this mode (pd = '1'), the oscillator will be stopped, the lock output will be made low, and the internal current reference will be turned off. During pd it is possible to load new divider ratios at the input buses (msel, psel, nsel). Power-down mode is ended by making pd low, causing the PLL to start up. The lock signal will be made high once the PLL has regained lock on the input clock.

12.7.4.4 Settings for USB0

Table 136 shows the divider settings used for configuring an output frequency F_{out} of 480 MHz for USB0.

12.7.4.5 Usage notes

In order to set up the PLL0, follow these steps:

1. Power down the PLL0 by setting bit 0 in the PLL0 control register (PLL0USB_CTRL or PLL0AUDIO_CTRL) to 1. This step is only needed if the PLL0 is currently enabled.
2. Configure the PLL0 m, n, and p divider values in the PLL0_M and PLL0_NP registers.
3. Power up the PLL0 by setting bit 0 in the PLL0 control register (PLL0USB_CTRL or PLL0AUDIO_CTRL) to 0.
4. Wait for the PLL0 to lock by monitoring the LOCK bit in the PLL0_STAT register.
5. Enable the PLL0 clock output in the PLL0_CTRL register.

Remark: You can change the PLL0 settings while the PLL0 is running when you need to configure the PLL0 for high output frequencies (see Section 12.2.1).

12.7.5 Fractional divider for PLL0AUDIO

The PLL0 for audio applications (PLL0AUDIO) includes an additional fractional divider. The SEL_EXT bit in the PLL0AUDIO control register determines whether the fractional divider is used (SEL_EXT=0) or bypassed (SEL_EXT=1). In the latter case, PLL0AUDIO operates exactly as PLL0USB and the MDEC value is used directly to control the feedback divider.

When the fractional divider is active, the sigma-delta modulator block generates divider values M and M+1 in the correct proportion so that an average division ratio of M+K/L is realized where 0<=K<=L and M, K, and L are integer values. M Is determined by the integer part of the PLLFRACT_CTRL register (PLLFRACT[21:15]) and K is determined by

13.1 How to read this chapter

Flash/EEPROM, Ethernet, USB0, USB1, and LCD related clocks are not available on all packages. See Table 2 and Table 4.

13.2 Basic configuration

The CCU1/2 are configured as follows:

- See Table 139 for clocking and power control.
- All branch clocks are enabled by default.
- Do not reset the CCUs during normal operation.
- Configure the output clock for the EMC clock divider (Table 148) together with bit 16 in the CREG6 register (Table 96).

Table 139. CCU clocking and power control

	Base clock	Branch clock	Operating frequency
CCU1	BASE_M3_CLK	CLK_M3_BUS	up to 180 MHz
CCU2	BASE_M3_CLK	CLK_M3_BUS	up to 180 MHz

Remark: The CCU registers for a given branch clock are only read and write accessible when the branch clock is enabled.

13.3 Features

The CCUs switch the clocks to individual peripherals on or off.

- Auto mode activates the AHB disable protocol before switching off the branch clock.
- In Wake-up mode, clocks can be selected to run automatically after a wake-up event.

13.4 General description

Each CGU base clock has several clock branches which can be turned on or off independently by the Clock Control Units CCU1 or CCU2. The branch clocks are distributed between CCU1 and CCU2.

Table 170. LPC18xx Pin description (flash-based parts) *…continued*

Pin name	LBGA256	TFBGA100	LQFP144	LQFP208		Reset state [1]	Type	Description
P3_6	B13	C7	122	174	[2]	N; PU	I/O	**GPIO0[6] —** General purpose digital input/output pin.
							-	**R —** Function reserved.
							I/O	**SSP0_SSEL —** Slave Select for SSP0.
							I/O	**SPIFI_MISO —** Input 1 in SPIFI quad mode; SPIFI output IO1.
							-	**R —** Function reserved.
							I/O	**SSP0_MISO —** Master In Slave Out for SSP0.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
P3_7	C11	D7	123	176	[2]	N; PU	-	**R —** Function reserved.
							-	**R —** Function reserved.
							I/O	**SSP0_MISO —** Master In Slave Out for SSP0.
							I/O	**SPIFI_MOSI —** Input 0 in SPIFI quad mode; SPIFI output IO0.
							I/O	**GPIO5[10] —** General purpose digital input/output pin.
							I/O	**SSP0_MOSI —** Master Out Slave in for SSP0.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
P3_8	C10	E7	124	179	[2]	N; PU	-	**R —** Function reserved.
							-	**R —** Function reserved.
							I/O	**SSP0_MOSI —** Master Out Slave in for SSP0.
							I/O	**SPIFI_CS —** SPIFI serial flash chip select.
							I/O	**GPIO5[11] —** General purpose digital input/output pin.
							I/O	**SSP0_SSEL —** Slave Select for SSP0.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
P4_0	D5	-	1	1	[2]	N; PU	I/O	**GPIO2[0] —** General purpose digital input/output pin.
							O	**MCOA0 —** Motor control PWM channel 0, output A.
							I	**NMI —** External interrupt input to NMI.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							O	**LCD_VD13 —** LCD data.
							I/O	**U3_UCLK —** Serial clock input/output for USART3 in synchronous mode.
							-	**R —** Function reserved.

Table 170. LPC18xx Pin description (flash-based parts) *…continued*

Pin name	LBGA256	TFBGA100	LQFP144	LQFP208		Reset state [1]	Type	Description
P6_11	H12	C9	101	143	[2]	N; PU	I/O	**GPIO3[7] —** General purpose digital input/output pin.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							O	**EMC_CKEOUT0 —** SDRAM clock enable 0.
							-	**R —** Function reserved.
							O	**T2_MAT3 —** Match output 3 of timer 2.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
P6_12	G15	-	103	145	[2]	N; PU	I/O	**GPIO2[8] —** General purpose digital input/output pin.
							O	**CTOUT_7 —** SCT output 7. Match output 3 of timer 1.
							-	**R —** Function reserved.
							O	**EMC_DQMOUT0 —** Data mask 0 used with SDRAM and static devices.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
P7_0	B16	-	110	158	[2]	N; PU	I/O	**GPIO3[8] —** General purpose digital input/output pin.
							O	**CTOUT_14 —** SCT output 14. Match output 2 of timer 3.
							-	**R —** Function reserved.
							O	**LCD_LE —** Line end signal.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
P7_1	C14	-	113	162	[2]	N; PU	I/O	**GPIO3[9] —** General purpose digital input/output pin.
							O	**CTOUT_15 —** SCT output 15. Match output 3 of timer 3.
							I/O	**I2S0_TX_WS —** Transmit Word Select. It is driven by the master and received by the slave. Corresponds to the signal WS in the *I^2S-bus specification*.
							O	**LCD_VD19 —** LCD data.
							O	**LCD_VD7 —** LCD data.
							-	**R —** Function reserved.
							O	**U2_TXD —** Transmitter output for USART2.
							-	**R —** Function reserved.

Table 170. LPC18xx Pin description (flash-based parts) *…continued*

Pin name	LBGA256	TFBGA100	LQFP144	LQFP208		Reset state [1]	Type	Description
PD_8	P8	-	-	74	[2]	N; PU	-	**R —** Function reserved.
							I	**CTIN_6 —** SCT input 6. Capture input 1 of timer 3.
							I/O	**EMC_D22 —** External memory data line 22.
							-	**R —** Function reserved.
							I/O	**GPIO6[22] —** General purpose digital input/output pin.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
PD_9	T11	-	-	84	[2]	N; PU	-	**R —** Function reserved.
							O	**CTOUT_13 —** SCT output 13. Match output 3 of timer 3.
							I/O	**EMC_D23 —** External memory data line 23.
							-	**R —** Function reserved.
							I/O	**GPIO6[23] —** General purpose digital input/output pin.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
PD_10	P11	-	-	86	[2]	N; PU	-	**R —** Function reserved.
							I	**CTIN_1 —** SCT input 1. Capture input 1 of timer 0. Capture input 1 of timer 2.
							O	**$\overline{\text{EMC\_BLS3}}$ —** LOW active Byte Lane select signal 3.
							-	**R —** Function reserved.
							I/O	**GPIO6[24] —** General purpose digital input/output pin.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
							-	**R —** Function reserved.
PD_11	N9	-	-	88	[2]	N; PU	-	**R —** Function reserved.
							-	**R —** Function reserved.
							O	**$\overline{\text{EMC\_CS3}}$ —** LOW active Chip Select 3 signal.
							-	**R —** Function reserved.
							I/O	**GPIO6[25] —** General purpose digital input/output pin.
							I/O	**USB1_ULPI_D0 —** ULPI link bidirectional data line 0.
							O	**CTOUT_14 —** SCT output 14. Match output 2 of timer 3.
							-	**R —** Function reserved.

18.4.3 GPIO port

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

18.5 Register description

The GPIO consists of the following blocks:

- The GPIO pin interrupts block at address 0x4008 7000. Registers in this block enable the up to 8 pin interrupts selected in the PINTSELn registers (see Table 187 or Table 188) and configure the level and edge sensitivity for each selected pin interrupt. The GPIO interrupt registers are listed in Table 228 to Table 237.
- The GPIO GROUP0 interrupt block at address 0x4008 8000. Registers in this block allow to configure any pin on port 0 and 1 to contribute to a combined interrupt. The GPIO GROUP0 registers are listed in Table 225 and Section 18.5.2.
- The GPIO GROUP1 interrupt block at address 0x4008 9000. Registers in this block allow to configure any pin on port 0 and 1 to contribute to a combined interrupt. The GPIO GROUP1 registers are listed in Table 226 and Section 18.5.2.
- The GPIO port block at address 0x400F 4000. Registers in this block allow to read and write to port pins and configure port pins as inputs or outputs.The GPIO port registers are listed in Table 227 and Section 18.5.3.

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

Table 224. Register overview: GPIO pin interrupts (base address: 0x4008 7000)

Name	Access	Address offset	Description	Reset value	Reference
ISEL	R/W	0x000	Pin Interrupt Mode register	0	Table 228
IENR	R/W	0x004	Pin interrupt level (rising edge) interrupt enable register	0	Table 229
SIENR	WO	0x008	Pin interrupt level (rising edge) interrupt set register	NA	Table 230
CIENR	WO	0x00C	Pin interrupt level (rising edge interrupt) clear register	NA	Table 231
IENF	R/W	0x010	Pin interrupt active level (falling edge) interrupt enable register	0	Table 232
SIENF	WO	0x014	Pin interrupt active level (falling edge) interrupt set register	NA	Table 233
CIENF	WO	0x018	Pin interrupt active level (falling edge) interrupt clear register	NA	Table 234
RISE	R/W	0x01C	Pin interrupt rising edge register	0	Table 235
FALL	R/W	0x020	Pin interrupt falling edge register	0	Table 236
IST	R/W	0x024	Pin interrupt status register	0	Table 237

19.6.3 DMA Interrupt Terminal Count Request Clear Register

The INTTCCLEAR Register is write-only and clears one or more terminal count interrupt requests. When writing to this register, each data bit that is set HIGH causes the corresponding bit in the status register (IntTCStat) to be cleared. Data bits that are LOW have no effect.

Table 256. DMA Interrupt Terminal Count Request Clear Register (INTTCCLEAR, address 0x4000 2008) bit description

Bit	Symbol	Description	Reset value	Access
7:0	INTTCCLEAR	Allows clearing the Terminal count interrupt request (IntTCStat) for DMA channels. Each bit represents one channel: 0 - writing 0 has no effect. 1 - clears the corresponding channel terminal count interrupt.	0x00	WO
31:8	-	Reserved. Read undefined. Write reserved bits as zero.	-	-

19.6.4 DMA Interrupt Error Status Register

The INTERRSTAT Register is read-only and indicates the status of the error request after masking.

Table 257. DMA Interrupt Error Status Register (INTERRSTAT, address 0x4000 200C) bit description

Bit	Symbol	Description	Reset value	Access
7:0	INTERRSTAT	Interrupt error status for DMA channels. Each bit represents one channel: 0 - the corresponding channel has no active error interrupt request. 1 - the corresponding channel does have an active error interrupt request.	0x00	RO
31:8	-	Reserved. Read undefined.	-	-

19.6.5 DMA Interrupt Error Clear Register

The INTERRCLR Register is write-only and clears the error interrupt requests. When writing to this register, each data bit that is HIGH causes the corresponding bit in the status register to be cleared. Data bits that are LOW have no effect on the corresponding bit in the register.

- The DMA Controller responds with a DMA acknowledge to the destination peripheral.
- The terminal count interrupt is generated (this interrupt can be masked).
- If the CLLI Register is not 0, then reload the CSRCADDR, CDESTADDR, CLLI, and CCONTROL Registers and go to back to step 2. However, if CLLI is 0, the DMA stream is disabled and the flow sequence ends.

19.8.2.3 Memory-to-memory DMA flow

For a memory-to-memory DMA flow the following sequence occurs:

1. Program and enable the DMA channel.
2. Transfer data whenever the DMA channel has the highest pending priority and the DMA Controller gains mastership of the AHB bus.
3. If an error occurs while transferring the data, generate an error interrupt and disable the DMA stream.
4. Decrement the transfer count.
5. If the count has reached zero:
   - Generate a terminal count interrupt (the interrupt can be masked).
   - If the CLLI Register is not 0, then reload the CSRCADDR, CDESTADDR, CLLI, and CCONTROL Registers and go to back to step 2. However, if CLLI is 0, the DMA stream is disabled and the flow sequence ends.

Note: Memory-to-memory transfers should be programmed with a low channel priority, otherwise other DMA channels cannot access the bus until the memory-to-memory transfer has finished, or other AHB masters cannot perform any transaction.

19.8.3 Interrupt requests

Interrupt requests can be generated when an AHB error is encountered or at the end of a transfer (terminal count), after all the data corresponding to the current LLI has been transferred to the destination. The interrupts can be masked by programming bits in the relevant CCONTROL and CCONFIG Channel Registers. Interrupt status registers are provided which group the interrupt requests from all the DMA channels prior to interrupt masking (RAWINTTCSTAT and RAWINTERRSTAT), and after interrupt masking (INTTCSTAT and INTERRSTAT). The INTSTAT Register combines both the INTTCSTAT and INTERRSTAT requests into a single register to enable the source of an interrupt to be quickly found. Writing to the INTTCCLEAR or the INTERRCLR Registers with a bit set HIGH enables selective clearing of interrupts.

19.8.3.1 Hardware interrupt sequence flow

When a DMA interrupt request occurs, the Interrupt Service Routine needs to:

1. Read the INTTCSTAT Register to determine whether the interrupt was generated due to the end of the transfer (terminal count). A HIGH bit indicates that the transfer completed. If more than one request is active, it is recommended that the highest priority channels be checked first.
2. Read the INTERRSTAT Register to determine whether the interrupt was generated due to an error occurring. A HIGH bit indicates that an error occurred.
3. Service the interrupt request.

4. For a terminal count interrupt, write a 1 to the relevant bit of the INTTCCLR Register. For an error interrupt write a 1 to the relevant bit of the INTERRCLR Register to clear the interrupt request.

19.8.4 Address generation

Address generation can be either incrementing or non-incrementing (address wrapping is not supported).

Some devices, especially memories, disallow burst accesses across certain address boundaries. The DMA controller assumes that this is the case with any source or destination area, which is configured for incrementing addressing. This boundary is assumed to be aligned with the specified burst size. For example, if the channel is set for 16-transfer burst to a 32-bit wide device then the boundary is 64-bytes aligned (that is address bits [5:0] equal 0). If a DMA burst is to cross one of these boundaries, then, instead of a burst, that transfer is split into separate AHB transactions.

Note: When transferring data to or from the SDRAM, the SDRAM access must always be programmed to 32 bit accesses. The SDRAM memory controller does not support AHB-INCR4 or INCR8 bursts using halfword or byte transfer-size. Start address in SDRAM should always be aligned to a burst boundary address.

19.8.4.1 Word-aligned transfers across a boundary

The channel is configured for 16-transfer bursts, each transfer 32-bits wide, to a destination for which address incrementing is enabled. The start address for the current burst is 0x0C000024, the next boundary (calculated from the burst size and transfer width) is 0x0C000040.

The transfer will be split into two AHB transactions:

- a 7-transfer burst starting at address 0x0C000024
- a 9-transfer burst starting at address 0x0C000040.

19.8.5 Scatter/gather

Scatter/gather is supported through the use of linked lists. This means that the source and destination areas do not have to occupy contiguous areas in memory. Where scatter/gather is not required, the CLLI Register must be set to 0.

The source and destination data areas are defined by a series of linked lists. Each Linked List Item (LLI) controls the transfer of one block of data, and then optionally loads another LLI to continue the DMA operation, or stops the DMA stream. The first LLI is programmed into the DMA Controller.

The data to be transferred described by a LLI (referred to as the packet of data) usually requires one or more DMA bursts (to each of the source and destination).

19.8.5.1 Linked list items

A Linked List Item (LLI) consists of four words. These words are organized in the following order:

1. CSRCADDR
2. CDESTADDR

- Generic DMA mode - Simultaneously sets controller_reset, fifo_reset, and dma_reset; clears the RAWINTS register @0x44 by using another write in order to clear any resultant interrupt. If a "graceful" completion of the DMA is required, then it is recommended to poll the status register to see whether the dma request is 0 before resetting the DMA interface control and issuing an additional FIFO reset.
- In DMA transfer mode, even when the FIFO pointers are reset, if there is a DMA transfer in progress, it could push or pop data to or from the FIFO; the DMA itself completes correctly. In order to clear the FIFO, the software should issue an additional FIFO reset and clear any FIFO underrun or overrun errors in the RAWINTS register caused by the DMA transfers after the FIFO was reset.

20.7.5.4 Error Handling

The Module implements error checking; errors are reflected in the RAWINTS register @0x44 and can be communicated to the software through an interrupt, or the software can poll for these bits. Upon power-on, interrupts are disabled (int_enable in the CTRL register is 0), and all the interrupts are masked (bits 0-31 of the INTMASK register; default is 0). Error handling:

- Response and data time-out errors - For response time-out, software can retry the command. For data time-out, the Module has not received the data start bit - either for the first block or the intermediate block - within the time-out period, so software can either retry the whole data transfer again or retry from a specified block onwards. By reading the contents of the TCBCNT later, the software can decide how many bytes remain to be copied.
- Response errors - Set when an error is received during response reception. In this case, the response that copied in the response registers is invalid. Software can retry the command.
- Data errors - Set when error in data reception are observed; for example, data CRC, start bit not found, end bit not found, and so on. These errors could be set for any block-first block, intermediate block, or last block. On receipt of an error, the software can issue a STOP or ABORT command and retry the command for either whole data or partial data.
- Hardware locked error - Set when the Module cannot load a command issued by software. When software sets the start_cmd bit in the CMD register, the Module tries to load the command. If the command buffer is already filled with a command, this error is raised. The software then has to reload the command.
- FIFO underrun/overrun error - If the FIFO is full and software tries to write data in the FIFO, then an overrun error is set. Conversely, if the FIFO is empty and the software tries to read data from the FIFO, an underrun error is set. Before reading or writing data in the FIFO, the software should read
- the fifo_empty or fifo_full bits in the Status register.
- Data starvation by cpu time-out - Raised when the Module is waiting for software intervention to transfer the data to or from the FIFO, but the software does not transfer within the stipulated time-out period. Under this condition and when a read transfer is in process, the software
- Should read data from the FIFO and create space for further data reception. When a transmit operation is in process, the software should fill data in the FIFO in order to start transferring data to the card.

Table 496. _MSC_CBW class structure

Member	Description
dSignature	uint32_t _MSC_CBW::dSignature
dTag	uint32_t _MSC_CBW::dTag
dDataLength	uint32_t _MSC_CBW::dDataLength
bmFlags	uint8_t _MSC_CBW::bmFlags
bLUN	uint8_t _MSC_CBW::bLUN
bCBLength	uint8_t _MSC_CBW::bCBLength
CB	uint8_t _MSC_CBW::CB[16][16]

25.5.14 _MSC_CSW

Table 497. _MSC_CSW class structure

Member	Description
dSignature	uint32_t _MSC_CSW::dSignature
dTag	uint32_t _MSC_CSW::dTag
dDataResidue	uint32_t _MSC_CSW::dDataResidue
bStatus	uint8_t _MSC_CSW::bStatus

25.5.15 _REQUEST_TYPE

Table 498. _REQUEST_TYPE class structure

Member	Description
B	uint8_t _REQUEST_TYPE::B byte wide access member
BM	BM_TBM_T _REQUEST_TYPE::BM bitfield structure access member

25.5.16 _USB_COMMON_DESCRIPTOR

Table 499. _USB_COMMON_DESCRIPTOR class structure

Member	Description
bLength	uint8_t _USB_COMMON_DESCRIPTOR::bLength Size of this descriptor in bytes
bDescriptorType	uint8_t _USB_COMMON_DESCRIPTOR::bDescriptorType Descriptor Type

25.5.17 _USB_CORE_DESCS_T

USB descriptors data structure.

Table 543. System time nanoseconds register (NANOSECONDS, address 0x4001 070C) bit description

Bit	Symbol	Description	Reset value	Access
30:0	TSSS	Time stamp sub seconds The value in this field has the sub second representation of time, with an accuracy of 0.46 nano-second. (When TSCTRLSSR in the MAC_TIMESTAMP_CTRL register is set, each bit represents 1 ns and the maximum value will be 0x3B9A_C9FF, after which it rolls-over to zero).	0	RO
31	PSNT	Positive or negative time This bit indicates positive or negative time value. If the bit is reset, it indicates that the time representation is positive, and if it is set, it indicates negative time value. (This bit represents the 32nd bit of the nanoseconds value when the Advance Time Stamp feature is enabled).	0	RO

26.6.20 System time seconds update register

This register contains the lower 32 bits of the seconds field to be written to, added to, or subtracted from the System Time value.

The System Time - Seconds Update register, along with the System Time - Nanoseconds Update register, initialize or update the system time maintained by the core. You must write both of these registers before setting the TSINIT or TSUPDT bits in the Time Stamp Control register.

Table 544. System time seconds update register (SECONDSUPDATE, address 0x4001 0710) bit description

Bit	Symbol	Description	Reset value	Access
31:0	TSS	Time stamp second The value in this field indicates the time, in seconds, to be initialized or added to the system time.	0	R/W

26.6.21 System time nanoseconds update register

This register contains 32 bits of the nano-seconds field to be written to, added to, or subtracted from the System Time value.

Table 631. Color display driven with 2 2/3 pixel data

Byte	CLD[7]	CLD[6]	CLD[5]	CLD[4]	CLD[3]	CLD[2]	CLD[1]	CLD[0]
0	P2[Green]	P2[Red]	P1[Blue]	P1[Green]	P1[Red]	P0[Blue]	P0[Green]	P0[Red]
1	P5[Red]	P4q[Blue]	P4[Green]	P4[Red]	P3[Blue]	P3[Green]	P3[Red]	P2[Blue]
2	P7[Blue]	P7[Green]	P7[Red]	P6[Blue]	P6[Green]	P6[Red]	P5[Blue]	P5[Green]

Each formatter consists of three 3-bit (RGB) shift left registers. RGB pixel data bit values from the gray scaler are concurrently shifted into the respective registers. When enough data is available, a byte is constructed by multiplexing the registered data to the correct bit position to satisfy the RGB data pattern of LCD panel. The byte is transferred to the 3-byte FIFO, which has enough space to store eight color pixels.

27.7.8 Panel clock generator

The output of the panel clock generator block is the panel clock, pin LCDDCLK. The panel clock can be based on either the peripheral clock for the LCD block or the external clock input for the LCD, pin LCDCLKIN. Whichever source is selected can be divided down in order to produce the internal LCD clock, LCDCLK.

The panel clock generator can be programmed to output the LCD panel clock in the range of LCDCLK/2 to LCDCLK/1025 to match the bpp data rate of the LCD panel being used.

The CLKSEL bit in the POL register determines whether the base clock used is CCLK or the LCDCLKIN pin.

27.7.9 Timing controller

The primary function of the timing controller block is to generate the horizontal and vertical timing panel signals. It also provides the panel bias and enable signals. These timings are all register-programmable.

27.7.10 STN and TFT data select

Support is provided for passive Super Twisted Nematic (STN) and active Thin Film Transistor (TFT) LCD display types:

27.7.10.1 STN displays

STN display panels require algorithmic pixel pattern generation to provide pseudo gray scaling on monochrome displays, or color creation on color displays.

27.7.10.2 TFT displays

TFT display panels require the digital color value of each pixel to be applied to the display data inputs.

27.7.11 Interrupt generation

Four interrupts are generated by the LCD controller, and a single combined interrupt. The four interrupts are:

- Master bus error interrupt.
- Vertical compare interrupt.

The bits in this register select which events, if any, clear the STOP bit in the Control register. (Since no events can occur when HALT is 1, only software can clear the HALT bit by writing the Control register.)

Table 673. SCT start condition register (START - address 0x4000 0014) bit description

Bit	Symbol	Description	Reset value
15:0	STARTMSK_L	If bit n is one, event n clears the STOP_L bit in the CTRL register (event 0 = bit 0, event 1 = bit 1, event 15 = bit 15).	0
31:16	STARTMSK_H	If bit n is one, event n clears the STOP_H bit in the CTRL register (event 0 = bit 16, event 1 = bit 17, event 15 = bit 31).	0

29.3.7 SCT dither condition register

If UNIFY = 1 in the CONFIG register, only the L bits are used.

If UNIFY = 0 in the CONFIG register, this register can be written to as two registers DITHER_L and DITHER_H. Both the L and H registers can be read or written individually or in a single 32-bit read or write operation.

When the Dither Condition register contains all zeroes (the default value), the dither engine advances to the next count in the dither pattern **every time** the SCT counter reaches zero (i.e. at the start of every new SCT counter cycle).

It is possible, using this register, to alter that behavior by qualifying the advancement through the dither pattern with designated events. As with the other condition/mask registers (HALT, STOP, LIMIT, etc.) each bit in this register corresponds to an event.

Setting one or more of the bits in this register to ones will cause the dither engine to advance to the next element in the dither pattern (i.e. increment the 16-state cycle counter) only following SCT counter cycles during which one or more of the designated dither events have occurred.

There is one, global Dither Condition register per 16-bit SCT. This register controls advancement through the dither patterns for all of the match registers associated with that half of the SCT.

For details on the dither engine and the dither pattern, see Section 29.4.1.1.

Table 705. Timer2 inputs and outputs *…continued*

Input/output	From/to multiplexed pin function	From/to internal signal	Default (see GIMA, Table 190)	CTOUTCTRL bit (see Table 96)
CAP3	T2_CAP2	-	no	-
	-	USART2 RX active	no	-
	-	I2S1_TX_MWS	no	-
	-	SCT output 7 OR T1 match channel 3	yes	0
	-	SCT output 7	yes	1
	-	T1 match channel 3	no	-
	T2_CAP3	-	no	-
Timer2 outputs				
MAT0	T2_MAT0	-	-	-
	CTOUT_8 if match ORed with SCT output	-	-	0
	-	ADC start1 input (ADC CR register bit START = 0x3)	no	-
MAT1	T2_MAT1	-	-	-
MAT2	T2_MAT2	-	-	-
MAT3	T3_MAT3	-	-	-
	CTOUT_11 if match ORed with SCT output	-	-	0

Table 706. Timer3 inputs and outputs

Input/output	From/to multiplexed pin function	From/to internal signal	Default (see GIMA, Table 190)	CTOUTCTRL bit (see Table 96)
Timer3 inputs				
CAP0	CTIN_0		yes	-
	-	I2S0_RX_MWS	no	-
	T3_CAP0	-	no	-
CAP1	CTIN_6	-	yes	-
	T3_CAP1	-	no	-
	-	USART3 TX active	no	-
	-	I2S0_TX_MWS	no	-
CAP2	CTIN_7	-	yes	-
	T3_CAP2	-	no	-
	-	USART3 RX active	no	-
	-	SOF0	no	-

Table 945. Register overview: I²C1 (base address 0x400E 0000) *…continued*

Name	Access	Address offset	Description	Reset value[1]	Reference
MASK1	R/W	0x034	**I2C Slave address mask register 1**. This mask register is associated with ADR1 to determine an address match. The mask register has no effect when comparing to the General Call address (‘0000000’).	0x00	Table 957
MASK2	R/W	0x038	**I2C Slave address mask register 2**. This mask register is associated with ADR2 to determine an address match. The mask register has no effect when comparing to the General Call address (‘0000000’).	0x00	Table 957
MASK3	R/W	0x03C	**I2C Slave address mask register 3**. This mask register is associated with ADR3 to determine an address match. The mask register has no effect when comparing to the General Call address (‘0000000’).	0x00	Table 957

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

42.7.1 I²C Control Set register

The CONSET registers control setting of bits in the CON 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 946. I²C Control Set register (CONSET - address 0x400A 1000 (I2C0) and 0x400E 0000 (I2C1)) 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.	-
2	AA	Assert acknowledge flag.	
3	SI	I²C interrupt flag.	0
4	STO	STOP flag.	0
5	STA	START flag.	0
6	I2EN	I²C interface enable.	0
31:7	-	Reserved. The value read from a reserved bit is not defined.	-

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 CONCLR 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.

Table 1050.Parallel Trace pin description

Pin Name	Type	Description
TRACECLK	Input	**Trace Clock.** This pin provides the sample clock for trace data on the TRACEDATA pins when tracing is enabled by an external debug tool.
TRACEDATA[3:0]	Output	**Trace Data bits 3 to 0.** These pins provide ETM trace data when tracing is enabled by an external debug tool. The debug tool can then interpret the compressed information and make it available to the user.

47.6 Debug connections

The LPC18xx supplies dedicated pins for JTAG and Serial Wire Debug (SWD). When a debug session is started, the part will be in JTAG debug mode. Once in debug mode, the debugger can switch the device to SWD mode.

Connections from a target board to the debugger can vary. Selecting a debug connector to add to a new board design depends on the debug tools that will be used.

47.6.1 ARM Standard JTAG connector (20-pin)

Figure 174 shows a standard JTAG connector. The ARM Standard JTAG Connector provides support for Serial Wire and JTAG interface modes in a 20-pin (0.1") connector. It can be used to access all SWD, SWV, and JTAG signals.

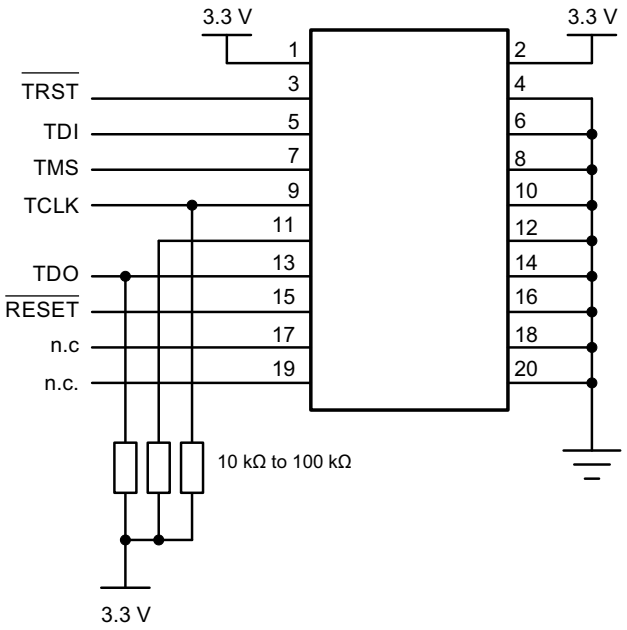

Fig 174. ARM Standard JTAG Connector

Chapter 47: LPC18xx JTAG, Serial Wire Debug (SWD), and trace functions

Chapter 48: LPC18xx API General error codes

Chapter 49: Supplementary information

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 26 July 2017
Document identifier: UM10430