STn8815A12 register descriptions

Introduction

This document describes the registers of the STn8815A12. For a more detailed description of each module, its features and functions, refer to the STn8815A12 datasheet.

The registers of the STn8815A12 are based on 32 bits. The descriptions provide the address (as an offset from the module base address), the reset value and a detailed bit field description. The software read, write or hardware access of each bit is printed below each table line, with following meanings:

- R: readable by software.
- W: writable by software.
- H: modified by hardware.

Bit fields may be referred to as Register_name.bit_field_name.

An example of the presentation is provided below.

Example

Registername **Register function**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BITA	BITB	0	0	0	0	BITFIELDA		0	0	0	BITFIELDB				
RW	RH	R	R	R	R	RW		R	R	R	R	R	R	R	R

Address: Module base address + register offset

Reset: 0x0000 0300

Description: Provides an example of a register description.

BITA Can be read and written by software, reset value is 0.

BITB Can be read by software and modified by hardware, reset value is 0.

BITFIELDA Can be read and written by software, reset value is 11.

BITFIELDB Can be read by software only.

Hexadecimal numbers are provided with the prefix 0x. Bit field values are provided in binary unless otherwise indicated.

Contents

1 STn8815 registers

1.1 STn8815 register addressing

Register addresses are provided as a base address plus the register offset.

1.2 STn8815 register summary

Table 1. STn8815 memory map

Base address	Register	Peripheral
0xA006 0000	Embedded SRAM bank 3 (128 Kbytes)	
0xA004 0000	Embedded SRAM bank 2 (128 Kbytes)	
0xA002 0000	Embedded SRAM bank 1 (128 Kbytes)	
0xA000 0000	Embedded SRAM bank 0 (128 Kbytes)	
0x8001 0000	Embedded backup RAM, 1 Kbytes remains supplied in deep-sleep mode	
0x5000 0000	CF/CF+/NAND Flash chip-select 1 (SMPS1n low)	
0x4000 0000	CF/CF+/NAND Flash chip-select 0 (SMPS0n low)	
0x3800 0000	NOR-Flash chip-select 2 (SMCS2n low) (64 Mbytes)	
0x3400 0000	NOR-Flash chip-select 1 (SMCS1n low) (64 Mbytes)	
0x3000 0000	NOR-Flash chip-select 0 (SMCS0n low) (64 Mbytes)	
0x1021 1000	L2 cache controller event monitor registers	
0x1021 0000	L2 cache controller registers	
0x101F F000	Hardware semaphores (HSEM)	DMA APB
0x101F D000	UART0 interface	DMA APB
0x101F C000	SSP interface	DMA APB
0x101F B000	UART1 interface	DMA APB
0x101F A000	FIrDA interface	DMA APB
0x101F 9000	MSP0 interface	DMA APB
0x101F 8000	I2C0 interface	DMA APB
0x101F 7000	I2C1 interface	DMA APB
0x101F 6000	SD-Card/MM-card interface (SDI)	DMA APB
0x101F 2000	UART2 interface	DMA APB
0x101F 1000	MSP1 interface	DMA APB
0x101F 0000	MSP2 interface	DMA APB
0x101E B000	Scroll key and keypad encoders (SKE)	Core APB
0x101E A000	One wire master (OWM)	Core APB
0x101E 9000	Power management unit (PMU	Core APB

Table 1. STn8815 memory map (continued)

Base address	Register	Peripheral
0x101E 8000	Real time clock, real time timer, PWL (RTC	Core APB
0x101E 7000	General purpose I/Os 96 to 103 (GPIO3)	Core APB
0x101E 6000	General purpose I/Os 64 to 95 (GPIO2)	Core APB
0x101E 5000	General purpose I/Os 32 to 63 (GPIO1)	Core APB
0x101E 4000	General purpose I/Os 0 to 31 (GPIO0)	Core APB
0x101E 3000	Multi timer unit (timers 4 to 7) (MTU1)	Core APB
0x101E 2000	Multi timer unit (timers 0 to 3) (MTU0)	Core APB
0x101E 1000	Watchdog timer (WDT)	Core APB
0x101E 0000	System and reset controller (SRC)	Core APB
0x1017 0000	USB OTG configuration registers and data	
0x1015 0000	DMAC1 control configuration registers	
0x1014 0000	VIC configuration registers	
0x1013 0000	DMAC0 control configuration registers	
0x1012 0000	CLCD/MDIF control configuration registers	
0x1011 0000	SDRAM control configuration registers	
0x1010 0000	Static memory control configuration registers	
0x0800 0000	SDRAM chip-select 1 (128 Mbytes)	
0x0000 0000	SDRAM chip-select 0 (128 Mbytes)	

Two consecutive 128 MByte regions starting at 0x7000 0000 are reserved for SDRAM chip-select 2 and 3 which are not currently delivered outside the chip due to pin-count limitation.

The flexible static memory controller (FSMC) supports up to 4 chip-select NOR-Flash memories (synchronous burst or standard, with muxed address/data bus) and 2 CompactFlash/CF+ card slots or SmartMedia/NAND-Flash chip-selects or a mix.

The remaining peripheral interfaces are contained in a single 1 MByte region, with the AHB peripherals at the bottom of the region and the APB peripherals at the top. This enables either type of peripheral to be added to the region. The core APB peripherals (0x101E XXXX) do not request DMA transfers and most are OS resources (timers, watchdog, real time clock). The DMA APB peripherals (0x101F XXXX) can perform DMA transfers. The hardware semaphores are also located in this address area, so they can be addressed by all the bus masters.

2 System and reset controller (SRC)

2.1 SRC register addressing

Register addresses are provided as the SRC base address, SRCBaseAddress, plus the register offset.

The SRCBaseAddress is 0x101E 0000.

2.2 SRC register summary

The internal system reset function initializes the SRC into a defined default state and is invoked either by asserting one the four sources of reset, which are:

- Power On Reset signal on pin PORn. This reset is intended to be used during power-up only. All DRAM and RTC contents will be lost during such a power on reset
- Software Reset, invoqued by writing to the SCRSTSR register, see section 7•4•7: Reset Status Register SRC_RSTSR on page 42). This reset performs a system reset identical to the Warm Hardware Reset.
- Watchdog Reset, invoqued when the watchdog reset generation is enable and the watchdog timer is not serviced by software. This reset performs a system reset identical to the Warm Hardware Reset.

On coming out of reset state, the device enters DOZE mode. Software can examine the reset status register, SCRSTSR, to determine which types of reset caused the last reset condition. Bit are set when reset command is released. Two concurrent reset requests can be memorized only if they are released at the same clock cycle (on HCLK). A status bit will be erased only by the arrival of an other reset source. Read of this register does not affect its content.

Table 2. SRC register list

Offset	Register	Description	Page
0x000	SRC_CR	SRC system control	9
0x004	SRC_IMCR	SRC interrupt mode control	10
0x008	SRC_IMSR	SRC interrupt mode status	11
0x00C	SRC_XTALCR(1)	SRC crystal control	12
0x010	SRC_PLLCR(1)	SRC PLL control	13
0x014	SRC_PLLFR(1)	SRC PLL frequency	14
0x018	SRC_RSTSR	SRC reset status	15
0x024	SRC_PCKEN0	SRC peripheral clock enable register 0	16
0x028	SRC_PCKDIS0	SRC peripheral clock disable register 0	16
0x02C	SRC_PCKENSR0	SRC peripheral clock enable status register 0	17
0x030	SRC_PCKSR0(1)	SRC peripheral clock status register 0	17
0x034	SRC_PCKEN1	SRC peripheral clock enable register 1	18

Table 2. SRC register list

Offset	Register	Description	Page
0x038	SRC_PCKDIS1	SRC peripheral clock disable register 1	18
0x03C	SRC_PCKENSR1	SRC peripheral clock enable status register 1	19
0x040	SRC_PCKSR1(1)	SRC peripheral clock status register 1	19
0x044	SRC_CLKOCR	SRC clock output configuration	20
0xFE0	SRCPeriphID0	SRC peripheral identification register 0 (bits 7:0)	22
0xFE4	SRCPeriphID1	SRC peripheral identification register 1 (bits 15:8)	22
0xFE8	SRCPeriphID2	SRC peripheral identification register 2 (bits 23:16)	22
0xFEC	SRCPeriphID3	SRC peripheral identification register 3 (bits 31:24)	23
0xFF0	SRCPCellID0	SRC PCell identification register 0 (bits 7:0)	23
0xFF4	SRCPCellID1	SRC PCell identification register 1 (bits 15:8)	23
0xFF8	SRCPCellID2	SRC PCell identification register 2 (bits 23:16)	24
0xFFC	SRCPCellID3	SRC PCell identification register 3 (bits 31:24)	24

1. These registers are only reset when the PORnot input is active, The other registers are reset when a software or watchdog reset occurs or when the WRSTnot input is active.

2.3 SRC register descriptions

SRC_CR System control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	T7EN SEL	0	T6EN SEL	0	T5EN SEL	0	T4EN SEL	0	T3EN SEL	0	T2EN SEL	0	T1EN SEL	0
R	R	RW	R	RW	R	RW	R	RW	R	RW	R	RW	R	RW	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T0EN SEL	HCLKDIVSEL		0	0	0	REMAP STAT	REMAP CLR	0	MODE STATUS				MODECR		
RW	RW		R	R	R	R	RW	R	R				RW		

Address: SRCBaseAddress + 0000

Reset: 0x0000 0289

Description: Defines the required operation of the STn8815 system.

T[0:7]ENSEL Timer 0 to 7 timing reference clock select. Defines the clock used to feed the timer.
0: REFCLK (32.768 kHz) 1: MXTALCLK divided by 8 (2.4 MHz)

HCLKDIVSEL HCLK divide ratio. Sets the divide ratio from CLK to HCLK when system is in normal mode (in other system modes, HCLK = CLK).
00: HCLK = CLK 01: HCLK = CLK / 2
10: HCLK = CLK / 3 11: HCLK = CLK / 4

REMAPSTAT Remap status request. Returns the REMAPSTAT input value.
0: memory map is the default (SDRAM banks 0-1 located at address 0x0000 0000 - 0x0FFF FFFF).
1: internal ROM or static memory banks remapped at address range 0x0000 0000 - 0x0FFF FFFF, according to REMAP[1:0] value latched at the end of power-on reset (default after reset).

REMAPCLR Remap clear request. Controls the REMAPCLR output.
1: memory remap is cleared

MODESTATUS Mode status. Returns the current operating mode as defined by the system controller state machine.
0000: sleep 0001: doze
0011: XTAL CTL 1011: SW to XTAL
1001: SW from XTAL 0010: slow
0110: PLL CTL 1110: SW to PLL
1010: SW from PLL 0100: normal

MODECR Mode control. Defines the required operating mode. The three bits are defined as:
bit 0: doze (set by default after reset)
bit 1: slow (clear by default after reset)
bit 2: normal (clear by default after reset)

0 Reserved for future use. Reading returns 0. Must be written with 0.

SRC_IMCR System controller interrupt mode control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PLL2 IM	PLL1 IM	ITMD PRI	ITMD TYPE	ITMDCR			ITMD EN
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW			RW

Address: SRCBaseAddress + 0004

Reset: 0x0000 0000

Description: Enables and controls STn8815 system operation when the VIC generates an interrupt.

PLL2IM PLL2 interrupt mask. This bit can mask the interrupt generated by PLL2 unlocking. If PLL2 unlocks:
0: interrupt status bit PLL2MIS of *SRC_IMSR* is not set (masked interrupt).
1: interrupt status bit PLL2MIS of SRC_IMSR is set and an interrupt signal is sent to the VIC.

PLL1IM PLL1 interrupt mask. This bit can mask the interrupt generated by PLL1 unlocking when the system is in normal mode. If PLL1 unlocks:
0: interrupt status bit PLL1MIS of SRC_IMSR is not set (masked interrupt). System remains in normal mode.
1: interrupt status bit PLL1MIS of SRC_IMSR is set and an interrupt signal is sent to the VIC. The system automatically switches to slow mode and returns to normal mode when bit PLL1MIS is cleared by software and PLL1 gets locked again.

ITMDPRI Interrupt mode priority enable. Enables the highest priority for the ARM instruction and data AHB through the STn8815 slave multiplexers when in interrupt mode.
0: no high priority 1: enables higher bus priority

ITMDTYPE Interrupt mode type. Defines which type of interrupt can cause the system to enter interrupt mode.
0: FIQ causes system to enter interrupt mode
1: FIQ or IRQ causes system to enter interrupt mode

ITMDCR Interrupt mode control. Defines the slowest operating mode that can be requested when in interrupt mode. These bits are ORed with the mode control bits in the system control register.

ITMDEN Interrupt mode enable. Interrupt mode is entered when an interrupt becomes active.
0: disabled 1: interrupt mode entered

0 Reserved for future use. Reading returns 0. Must be written with 0.

SRC_IMSR System controller interrupt mode status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	PLL2 MIS	PLL1 MIS	ITMD STAT
R	R	R	R	R	R	R	R	R	R	R	R	R	RWH	RWH	RWH

Address: SRCBaseAddress + 0008

Reset: 0x0000 0000

Description: Monitors and controls the STn8815 system interrupt mode.

PLL2MIS PLL2 unlock masked interrupt status. Set when PLL2 gets unlocked. Must be cleared by software.
0: interrupt is not active 1: interrupt is active

PLL1MIS PLL1 unlock masked interrupt status. Set when PLL1 is unlocked while the system is in normal mode. When set, the system automatically enters slow mode. It must be cleared by software, and once cleared, the system automatically returns to normal mode when PLL1 is locked again.
0: interrupt is not active 1: interrupt is active

ITMDSTAT Interrupt mode status. Can be directly written to enable software control of the interrupt mode logic. The interrupt mode must be cleared manually when the interrupt service routine has finished.
0: interrupt mode is not active 1: interrupt mode is active

0 Reserved for future use. Reading returns 0. Must be written with 0.

SRC_XTALCR Crystal control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	XTAL TIMEN	SXTAL DIS	MXTALTIME		
R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MXTALTIME													MXTAL STAT	MXTAL EN	MXTAL OVER
RW													R	RW	RW

Address: SRCBaseAddress + 000C

Reset: 0x0000 0000

Description: Controls the crystal oscillator that generates the system clock in slow and normal modes.

XTALTIMEN Crystal lock timer enable. Enables the timer to evaluate the stabilization (lock) of the external crystal.
0: XTAL lock enables the transition of the state machine
1: XTAL lock and timeout enables the transition of the state machine

SXTALDIS 32 kHz crystal disable bit. Disables the 32 kHz crystal oscillator by switching off the inverter that drives the SXTALO pin, to save power when the SXTALI pin is driven by an external clock generator.
0: 32 kHz oscillator is running. A crystal oscillates if connected between SXTALI and SXTALO pins.
1: 32 kHz oscillator is stopped. SXTALI pin must receive an external square wave signal.

MXTALTIME Main crystal timeout count. Defines the number of slow-speed oscillator cycles permitted for the crystal oscillator output to settle after being enabled. Timeout value is given by: 65536 - MXTALTIME.

MXTALSTAT Main crystal status. Returns the value on the MXTALON input signal.

MXTALEN Main crystal enable. Directly controls the MXTALEN signal when crystal control override is enabled.

MXTALOVER Crystal control override. Enables the crystal control signals to be placed under direct software control, rather than being controlled by the system mode control state machine.
0: disable crystal control override 1: enable crystal control override

0 Reserved for future use. Reading returns 0. Must be written with 0.

SRC_PLLCR PLL control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	PLL TIMEN	PLL2 STAT	PLL2 EN	PLL1TIME											
	RW	RH	RW	RW											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL1TIME													PLL1 STAT	PLL1 EN	PLL1 OVER
RW													RH	RW	RW

Address: SRCBaseAddress + 0010

Reset: 0x0000 0000

Description: Directly controls the PLL.

PLLTIMEN PLL lock timer enable bit. Enables the timer to evaluate the lock time of the PLL.
0: PLL lock enables the transition of the state machine.
1: PLL lock and timeout enables the transition of the state machine.

PLL2STAT PLL2 status. Returns the lock state of the PLL2.
0: PLL2 output frequency is not stable 1: PLL2 output frequency is stable

PLL2EN PLL2 enable. It takes some time for the PLL2 to deliver a stable output frequency. The peripheral clocks remain inactive until the PLL2 output frequency is stabilized. PLL2 is not automatically enabled or stopped when the system mode is changed (to normal, slow, doze, or sleep), thus it is the responsibility of software to disable the PLL2 when not used for reducing power consumption.
0: PLL2 stopped 1: PLL2 running

PLL1TIME PLL1 timeout count. Defines the number of crystal oscillator cycles permitted for the PLL output to settle after being enabled. The timeout value is given by: 33554432 - PLL1TIME.

PLL1STAT PLL1 status bit. Returns the lock state of PLL1. It reflects the PLLON input signal level, used in the system mode control state machine. It is used to switch from slow mode to normal mode. When the software programs normal mode, the signal PLL1ON must go high for state-machine to go to normal mode.
0: PLL1 output frequency is not stable 1: PLL1 output frequency is stable

PLL1EN PLL1 enable. Controls the PLL1EN output when the PLL control override is enabled.

PLL1OVER PLL1 control override. Enables the PLL control signals to be placed under software control, rather than being controlled by the system mode control state machine.
0: Disable PLL1 control override 1: Enable PLL1 control override

0 Reserved for future use. Reading returns 0. Must be written with 0.

SRC_PLLFR PLL frequency register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	PLL2NMUL						0	0	0	0	0	0	0	PLL2P DIV
R	R	RW						R	R	R	R	R	R	R	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	PLL1NMUL						0	0	0	0	0	PLL1PDIV		
R	R	RW						R	R	R	R	R	RW		

Address: SRCBaseAddress + 0014

Reset: 0x0000 0000

Description: Controls the PLL frequency.

There are some PLL1 output frequency restrictions. The PLL1 VCO operates best in the frequency range 600 MHz - 1280 MHz. The CPU operates best in the frequency range 0 to 270 MHz (can reach 264 MHz including the PLL jitter of 100 ps).

The PLL1NMUL multiply factor must be programmed so that the VCO output clock and the CPU clock operate in their characterized frequency range, so the following formula must be respected at all times:

600 MHz <= Freq (VCOCLK) = Freq (MXTAL) x 2 x (PLL1NMUL+2) <=

1280 MHz, and,
0 MHz <= Freq (CLK) = Freq (MXTAL) x (PLL1NMUL + 2) / ($2^{PLL1PDIV}$) <= 264 MHz.
For example:
PLL1PDIV= 2 = 010,
PLL1NMUL = 53 = 0x35 = 11 0101,
Freq (OSCIN) = 19.2 MHz.
Gives: Freq (VCOCLK) = 19.2 x (53+2) = 1056 MHz,
Freq (CLK) = Freq (VCO) / 2^2 = 264.0 MHz

PLL2NMUL PLL2 N multiply factor. PLL2EN must be cleared before changing this bit-field. When PLL2 is enabled, these bits set the multiply factor of PLL2 to achieve the nominal frequencies requested by the peripherals. $F_{PLL2} = F_{MXTAL}$ x (PLL2NUM +1) / 2

The following values must be loaded according to the MXTAL input frequency:

0x2E: MXTAL frequency is 13.0 MHz (F_{PLL2} = 624 MHz)

0x34: MXTAL frequency is 16.0 MHz (F_{PLL2} = 864 MHz)

0x2B: MXTAL frequency is 19.2 MHz (F_{PLL2} = 864 MHz)

0x22: MXTAL frequency is 24.0 MHz (F_{PLL2} = 864 MHz)

0x16: MXTAL frequency is 26.0 MHz (F_{PLL2} = 624 MHz)

PLL2PDIV PLL2 post-divisor factor. Sets the division factors for programmable dividers to achieve correct peripheral clock frequency according to the F_{PLL2} value. PLL2EN must be cleared before changing this bit-field.

0: use when F_{PLL2} = 624 MHz. Peripheral clock values: 208 MHz/ 104 MHz/ 78 MHz/ 48 MHz.

1: use when F_{PLL2} = 864 MHz. Peripheral clock values: 216 MHz/ 108 MHz/ 72 MHz/ 48 MHz.

PLL1NMUL PLL1 N multiply factor. When PLL is enabled, these bits together with PLL1PDIV bits set the multiply factor between the core clock (CLK) and the oscillator clock (OSCIN).

Freq(CLK) = Freq(OSCIN) x (PLL1NMUL + 2) / ($2^{PLL1PDIV}$)

PLL1PDIV PLL1 P post-divisor factor. See previous bit field description.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SRC_RSTSR Reset status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	REMAP1	REMAP0	SLEEPRST	SOFTRST	WDTRST	RESERVED	PORRST
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: SRCBaseAddress + 0018

Reset: 0x0000 000X

Description: Writing any value causes the SoftResReq output to pulse high for a single clock cycle. Reading gives the status of the last reset to be generated.

REMAP1 REMAP1 latched value. Reveals the value of REMAP1 (GPIO31) pin latched at power-on reset.

REMAP0 REMAP0 latched value. Reveals the value of REMAP0 (GPIO30) pin latched at power-on reset.

SLEEPRST Sleep reset status. Indicates if the last reset was generated by an exit from deep-sleep state.
0: not generated by deep-sleep state exit 1: generated by deep-sleep state exit

SOFTRST Soft reset status. Indicates if the last reset was generated by a write to the reset status register.
0: not generated by reset status register write 1: generated by reset status register write

WDTRST Watchdog reset status. Indicates if the last reset was generated by a watchdog reset.
0: not generated by watchdog reset 1: generated by watchdog reset

PORRST Power-on reset status. Indicates if the last reset was generated by a POR.
0: not generated by a POR 1: generated by a POR

0 Reserved for future use. Reading returns 0. Must be written with 0.

SRC_PCKEN0 Peripheral clock enable register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PerCk En31	PerCk En30	PerCk En29	PerCk En28	PerCk En27	PerCk En26	PerCk En25	PerCk En24	PerCk En23	PerCk En22	PerCk En21	PerCk En20	PerCk En19	PerCk En18	PerCk En17	PerCk En16
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PerCk En15	PerCk En14	PerCk En13	PerCk En12	PerCk En11	PerCk En10	PerCk En9	PerCk En8	PerCk En7	PerCk En6	PerCk En5	PerCk En4	PerCk En3	PerCk En2	PerCk En1	PerCk En0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Address: SRCBaseAddress + 0024

Reset: -

Description: This write-only register initiates an enable in clock generation for peripheral clocks. A bit written as 1 sets the corresponding PERIPHCLKEN0[x] signal thus requests an enable for the peripheral clock. See *Table 3 on page 19* for clock details.

PerCkEn[0:31] Peripheral clock enable.
0: no effect 1: enable the clock of the attached peripheral

SRC_PCKDIS0 Peripheral clock disable register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PerCk Dis31	PerCk Dis30	PerCk Dis29	PerCk Dis28	PerCk Dis27	PerCk Dis26	PerCk Dis25	PerCk Dis24	PerCk Dis23	PerCk Dis22	PerCk Dis21	PerCk Dis20	PerCk Dis19	PerCk Dis18	PerCk Dis17	PerCk Dis16
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PerCk Dis15	PerCk Dis14	PerCk Dis13	PerCk Dis12	PerCk Dis11	PerCk Dis10	PerCk Dis9	PerCk Dis8	PerCk Dis7	PerCk Dis6	PerCk Dis5	PerCk Dis4	PerCk Dis3	PerCk Dis2	PerCk Dis1	PerCk Dis0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Address: SRCBaseAddress + 0028

Reset: -

Description: This write-only register initiates a disable in clock generation for peripheral clocks. A bit written as 1 clears the corresponding PERIPHCLKEN0[x] signal and thus requests a disable for the corresponding peripheral clock. See *Table 3 on page 19* for clock details.

PerCkDis [0:31] Peripheral clock disable.
0: no effect on PERIPHCLKEN0[0:31] signal 1: disable the clock of the attached peripheral

SRC_PCKENSR0 Peripheral clock enable status register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PerCk EnSt31	PerCk EnSt30	PerCk EnSt29	PerCk EnSt28	PerCk EnSt27	PerCk EnSt26	PerCk EnSt25	PerCk EnSt24	PerCk EnSt23	PerCk EnSt22	PerCk EnSt21	PerCk EnSt0	PerCk EnSt19	PerCk EnSt18	PerCk EnSt17	PerCk EnSt16
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PerCk EnSt15	PerCk EnSt14	PerCk EnSt13	PerCk EnSt12	PerCk EnSt11	PerCk EnSt10	PerCk EnSt9	PerCk EnSt8	PerCk EnSt7	PerCk EnSt6	PerCk EnSt5	PerCk EnSt4	PerCk EnSt3	PerCk EnSt2	PerCk EnSt1	PerCk EnSt0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: SRCBaseAddress + 002C

Reset: 0xFFFF FFFF

Description: Verifies the writes to *SRC_PCKEN0* and *SRC_PCKDIS0*.

PerCkEnSt [0:31] Peripheral clock enable status.
0: disable of clock[0:31] has been requested 1: enable of clock[0:31] has been requested

SRC_PCKSR0 Peripheral clock status register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Per Ck31	Per Ck30	Per Ck29	Per Ck28	Per Ck27	Per Ck26	Per Ck25	Per Ck24	Per Ck23	Per Ck22	Per Ck21	Per Ck20	Per Ck19	Per Ck18	Per Ck17	Per Ck16
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Per Ck15	Per Ck14	Per Ck13	Per Ck12	Per Ck11	Per Ck10	Per Ck9	Per Ck8	Per Ck7	Per Ck6	Per Ck5	Per Ck4	Per Ck3	Per Ck2	Per Ck1	Per Ck0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: SRCBaseAddress + 0030

Reset: 0xFFFF FFFF

Description: Monitors the status of the peripheral clocks. Synchronization delays mean clock enables or disables are not immediately acted on.

PerCk[0:31] Peripheral clock status. Reflects the current value of peripheral clock enable signal.
0: clock of attached peripheral is stopped 1:clock of attached peripheral is running

SRC_PCKEN1 Peripheral clock enable register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PerCk En63	PerCk En62	PerCk En61	PerCk En60	PerCk En59	PerCk En58	PerCk En57	PerCk En56	PerCk En55	PerCk En54	PerCk En53	PerCk En52	PerCk En51	PerCk En50	PerCk En49	PerCk En48
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PerCk En47	PerCk En46	PerCk En45	PerCk En44	PerCk En43	PerCk En42	PerCk En41	PerCk En40	PerCk En39	PerCk En38	PerCk En37	PerCk En36	PerCk En35	PerCk En34	PerCk En33	PerCk En32
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Address: SRCBaseAddress + 0034

Reset: -

Description: This write-only register initiates an enable in clock generation for the clock of the attached peripheral by setting the corresponding PERIPHCLKEN1[x] signal. See *Table 3 on page 19* for clock details.

PerCkEn [32:63] Peripheral clock enable.
0: no effect on PERIPHCLKEN1[32:63] signal 1: enable the clock of the peripheral

SRC_PCKDIS1 Peripheral clock disable register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PerCk Dis63	PerCk Dis62	PerCk Dis61	PerCk Dis60	PerCk Dis59	PerCk Dis58	PerCk Dis57	PerCk Dis56	PerCk Dis55	PerCk Dis54	PerCk Dis53	PerCk Dis52	PerCk Dis51	PerCk Dis50	PerCk Dis49	PerCk Dis48
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PerCk Dis47	PerCk Dis46	PerCk Dis45	PerCk Dis44	PerCk Dis43	PerCk Dis42	PerCk Dis41	PerCk Dis40	PerCk Dis39	PerCk Dis38	PerCk Dis37	PerCk Dis36	PerCk Dis35	PerCk Dis34	PerCk Dis33	PerCk Dis32
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Address: SRCBaseAddress + 0038

Reset: -

Description: This write-only register disables the clock of an attached peripheral by clearing the corresponding PERIPHCLKEN1[x] signal. See *Table 3 on page 19* for clock details.

PerCkDis [32:63] Peripheral clock disable.
0: no effect on PERIPHCLKEN1[32:63] signal 1: disable the clock[32:63] of the peripheral

SRC_PCKENSR1 Peripheral clock enable status register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PerCk EnSt63	PerCk EnSt62	PerCk EnSt61	PerCk EnSt60	PerCk EnSt59	PerCk EnSt58	PerCk EnSt57	PerCk EnSt56	PerCk EnSt55	PerCk EnSt54	PerCk EnSt53	PerCk EnSt52	PerCk EnSt51	PerCk EnSt50	PerCk EnSt49	PerCk EnSt48
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PerCk EnSt47	PerCk EnSt46	PerCk EnSt45	PerCk EnSt44	PerCk EnSt43	PerCk EnSt42	PerCk EnSt41	PerCk EnSt40	PerCk EnSt39	PerCk EnSt38	PerCk EnSt37	PerCk EnSt36	PerCk EnSt35	PerCk EnSt34	PerCk EnSt33	PerCk EnSt32
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: SRCBaseAddress + 003C

Reset: 0xFFFF FFFF

Description: Verifies writes to registers *SRC_PCKEN1* and *SRC_PCKDIS1*.

PerCkEnSt [32:63] Peripheral clock enable status register.
0: clock disable has been requested 1: clock enable has been requested

SRC_PCKSR1 Peripheral clock status register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Per Ck63	Per Ck62	Per Ck61	Per Ck60	Per Ck59	Per Ck58	Per Ck57	Per Ck56	Per Ck55	Per Ck54	Per Ck53	Per Ck52	Per Ck51	Per Ck50	Per Ck49	Per Ck48
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Per Ck47	Per Ck46	Per Ck45	Per Ck44	Per Ck43	Per Ck42	Per Ck41	Per Ck40	Per Ck39	Per Ck38	Per Ck37	Per Ck36	Per Ck35	Per Ck34	Per Ck33	Per Ck32
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: SRCBaseAddress + 0040

Reset: 0xFFFF FFFF

Description: Monitors the status of the peripheral clocks. Synchronization delays mean clock enables or disables are not immediately acted on.

PerCk[32:63] Peripheral clock status reflects the current value of the PERIPHCLKEN1[32:63] signal.
0: clock of attached peripheral is stopped 1: clock of attached peripheral is running

Table 3. PCKEN0/1 assignation

Bit	PCKEN0	PCKEN1
26	HCLK3D: 3D Graphics AMBA clock	3DCLK: 3D Graphics core clock
25	PCLKHSEM: HSEM AMBA clock	Reserved
24	PCLKSKE: SKE AMBA clock	SKECLK: SKE kernel clock
23	Reserved	Reserved
22	PCLKOWM: OWM AMBA clock	OWMCLK: OWM kernel clock
21	PCLKMSP2: MSP2 AMBA clock	MSPCLK2: MSP2 kernel clock
20	PCLKMSP1: MSP1 AMBA clock	MSPCLK1: MSP1 kernel clock

Table 3. PCKEN0/1 assignation

Bit	PCKEN0	PCKEN1
19	PCLKUART2: UART2 AMBA clock	UART2CLK: UART2 kernel clock
18	RESERVED	RESERVED
17	RESERVED	RESERVED
16	RESERVED	IPBMCCLK: BMC clock for VPIP
15	RESERVED	IPI2CCLK: VPIP I2C kernel clock
14	HCLKDIF: Display interface AMBA clock	DIFCLK: Display interface kernel clock
13	HCLKUSB: USB AMBA clock	USBCLK: USB kernel clock (48MHz, not ULPI clock)
12	PCLKMSP0: MSP0 AMBA clock	MSPCLK0: MSP0 kernel clock
11	PCLKUART1: UART1 AMBA clock	UART1CLK: UART1 kernel clock
10	PCLKI2C1: I2C1 AMBA clock	I2C1CLK: I2C1 kernel clock
9	PCLKI2C0: I2C0 AMBA clock	I2C0CLK: I2C0 kernel clock
8	PCLKSDI: SD/MM card interface AMBA clock	SDICLK: SD/MM card interface kernel clock
7	PCLKUART0: UART0 AMBA clock	UART0CLK: UART0 kernel clock
6	PCLKSSP: SSP AMBA clock	SSPICLK: SSP kernel clock
5	PCLKIRDA: IRDA AMBA clock	IRDACLK: FIRDA kernel clock
4	HCLKCLCD: LCD controller AMBA clock	CLCDCLK: LCD controller kernel clock
3	HCLKDMA1: DMA1 AMBA clock	Reserved
2	HCLKSDRAM: SDRAM controller AMBA clock	Reserved
1	HCLKSMC: Static memory controller AMBA clock	Reserved
0	HCLKDMA0: DMA0 AMBA clock	Reserved

SRC_CLKOCR Clock output configuration register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	CLKOSEL1			CLKODIV1					
R	R	R	R	R	R	R	RW			RW					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	CLKOSEL0			CLKODIV0					
R	R	R	R	R	R	R	RW			RW					

Address: SRCBaseAddress + 0044

Reset: 0x0000 0000

Description: There are two programmable clock output signals delivered by the system and reset controller, CLKOUT0 and CLKOUT1. Each can be individually programmed to deliver a sub-multiple of one of the following clocks:

PLL1 output clock for CLKOUT0, MXTALI clock for CLKOUT1,
PLL2 CLK216 clock output (216 MHz: MXTALI = 19.2 MHz, 208 MHz:XTALI = 13 MHz),
PLL2 CLK108 clock output (108 MHz: MXTALI = 19.2MHz, 104 MHz:XTALI = 13MHz),
PLL2 CLK72 clock output (72 MHz: MXTALI = 19.2 MHz, 78 MHz:XTALI = 13 MHz),
PLL2 CLK48 (48 MHz) clock output,
PLL2 CLK27 (27 MHz) clock output,
PLL2 CLK2 (2.4 MHz) clock output,
PLL2 CLK009 (90.056 kHz) clock output.
This general-purpose 32-bit register directly controls the PLL frequency generated.

CLKOSEL1 Off-chip clock CLKOUT1 selection. When the alternate function of GPIO55 is enabled, the pin delivers the CLKOUT1 signal, derived from one of the MXTAL or PLL2 output frequencies, divided by a programmable factor. CLKOSEL selects which PLL output is used.

000: MXTALI frequency divided by (CLKODIV1+1	001: PLL2 CLK216 output freq. / (CLKODIV1+1)
010: PLL2 CLK108 output freq. / (CLKODIV1 + 1)	011: PLL2 CLK72 output freq. / (CLKODIV1 + 1)
100: PLL2 CLK48 output freq. / (CLKODIV1 + 1)	101: PLL2 CLK27 output freq. / (CLKODIV1 + 1)
110: PLL2 CLK2 output freq. / (CLKODIV1 + 1)	111: PLL2 CLK009 output freq. / (CLKODIV1 + 1)

CLKODIV1 Off-chip clock CLKOUT1 frequency divisor. When the alternate function of GPIO55 is enabled, the pin delivers the CLKOUT1 signal, derived from one of the MXTAL or PLL2 output frequencies, divided by this programmable factor CLKODIV.

00000: MXTAL/PLL2 frequency divided by 1	00001: MXTAL/PLL2 frequency divided by 2
00010: MXTAL/PLL2 freq. divided by 3, through	11111: MXTAL/PLL2 frequency divided by 32

CLKOSEL0 Off-chip clock CLKOUT0 selection. When the alternate function of GPIO25 is enabled, the pin delivers the CLKOUT0 signal, derived from one of the PLL1 or PLL2 output frequencies, divided by a programmable factor. CLKOSEL0 selects which PLL output is used.

000: PLL1 output freq. / (CLKODIV0 + 1)	001: PLL2 CLK216 output freq. / (CLKODIV1 + 1)
010: PLL2 CLK108 output freq. / (CLKODIV1 + 1)	011: PLL2 CLK72 output freq. / (CLKODIV1 + 1)
100: PLL2 CLK48 output freq. / (CLKODIV1 + 1)	101: PLL2 CLK27 output freq. / (CLKODIV1 + 1)
110: PLL2 CLK2 output freq. / (CLKODIV1 + 1)	111: PLL2 CLK009 output freq. / (CLKODIV1 + 1)

CLKODIV0 Off-chip clock CLKOUT0 frequency divisor. When the alternate function of GPIO25 is enabled, the pin delivers the CLKOUT0 signal, derived from one of the PLL1 or PLL2 output frequencies, divided by this programmable factor CLKODIV0.

00000: selected PLL frequency divided by 1	00001: selected PLL frequency divided by 2
00010: selected PLL frequency divided by 3	11111: selected PLL frequency divided by 32

0 Reserved for future use. Reading returns 0. Must be written with 0.

SRCPeriphID0 SRC peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: SRCBaseAddress + 0FE0

Reset: 0x0000 0003

Description: PartNumber0 reads back as 0x03.

SRCPeriphID1 SRC peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: SRCBaseAddress + 0FE4

Reset: 0x0000 0018

Description: Designer0 reads back as 0x1. PartNumber1 reads back as 0x8.

SRCPeriphID2 SRC peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: SRCBaseAddress + 0FE8

Reset: 0x0000 0004

Description: Revision reads back as 0x0. Designer1 reads back as 0x4.

SRCPeriphID3 SRC peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: SRCBaseAddress + 0FEC

Reset: 0x0000 0000

Description: Configuration reads back as 0x00.

SRCPCellID0 SRC PCell identification registers 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SRCPCellID0							
R	R	R	R	R	R	R	R	R							

Address: SRCBaseAddress + 0FF0

Reset: 0x0000 000D

Description: SRCPCellID0 reads back as 0x0D.

SRCPCellID1 SRC PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SRCPCellID1							
R	R	R	R	R	R	R	R	R							

Address: SRCBaseAddress + 0FF4

Reset: 0x0000 00F0

Description: SRCPCellID1 reads back as 0xF0.

SRCPCellID2 SRC PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SRCPCellID2							
R	R	R	R	R	R	R	R	R							

Address: SRCBaseAddress + 0FF8

Reset: 0x0000 0005

Description: SRCPCellID2 reads back as 0x05.

SRCPCellID3 SRC PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SRCPCellID3							
R	R	R	R	R	R	R	R	R							

Address: SRCBaseAddress + 0FFC

Reset: 0x0000 00B1

Description: SRCPCellID3 reads back as 0xB1.

3 Power management unit (PMU)

3.1 PMU register addressing

Register addresses are provided as the PMU base address, PMUBaseAddress, plus the register offset.

The PMUBaseAddress is 0x101E 9000.

3.2 PMU register summary

The device communicates with the system via 32-bit wide control registers accessible via the AMBA rev. 2.0 peripheral bus (APB).

Table 4. PMU register list

Offset	Register	Description	Page
0x000	PMU_CTRL	PMU control register	*26*
0x004	PMU_STA	PMU status register	*27*
0x008	PMU_CTRL2	PMU control register 2	*28*
0x00C	PMU_STA2	PMU status register 2	*29*
0x010	PMU_SPR0	PMU scratch pad register 0	*30*
0x028	PMU_RIS	PMU raw interrupt status register	*30*
0x02C	PMU_ICR	PMU interrupt clear register	*30*
0xFE0	PMUPeriphID0	PMU peripheral identification register 0 (bits [7:0])	*31*
0xFE4	PMUPeriphID1	PMU peripheral identification register 1 (bits [15:8])	*31*
0xFE8	PMUPeriphID2	PMU peripheral identification register 2 (bits [23:16])	*31*
0xFEC	PMUPeriphID3	PMU peripheral identification register 3 (bits [31:24])	*32*
0xFF0	PMUPCellID0	PMU PCell identification register 0 (bits [7:0])	*32*
0xFF4	PMUPCellID1	PMU PCell identification register 1 (bits [15:8])	*32*
0xFF8	PMUPCellID2	PMU PCell identification register 2 (bits [23:16])	*33*
0xFFC	PMUPCellID3	PMU PCell identification register 3 (bits [31:24])	*33*

3.3 PMU register descriptions

PMU_CTRL PMU control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FSMC PDEN	0	0	EXTBR KEN	RESER VED	LCDn DIF	SDMC PDEN	SDMCSTR		DIFCLC DHiZ	RESER VED	0	XTIEN	IRPDB G	0	0
RW	R	R	RW	RW	RW	RW	RW		RW	R	R	RW	RW	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	TVOEN	DEEP DIS	SDMC HLD	IO FORCE	VDDO KMOD	BATOK MOD
R	R	R	R	R	R	R	R	R	R	RW	RW	RWH	RWH	RW	RW

Address: PMU base address + 0x0000

Reset: 0x0000 0000

Description: Controls the power manager.

FSMCPDEN FSMC pull-down enable on data bus (SMADQ[15:0]).

0: disable 1: enable

DSPDBGI2C Smart audio and video accelerator DSP I2C debug port enable.

0: SAA and SVA debug port is JTAG-based (common for the two DSPs): when GPIO27 is latched low at PORnot, GPIO37 = HTCK, GPIO36 = HTMS, GPIO35 = HTDO, GPIO34 = HTDI. When the DBGCFG pin (GPIO27) is latched high at PORnot, they are JTAG port pins.

1: SAA and SVA debug ports are I2C based (one port per DSP): GPIO37 = HASCL, GPIO36 = HASDA, GPIO35 = HVSCL, GPIO34 = HVSDA.

EXTBRKEN External debugger break (BRKIN/BRKOUT pins) enable.

0: disabled: GPIO[39:38] are user-defined (GPIO or alt function, depending on GPIO register settings).

1: enabled: GPIO39 = BRKOUT, GPIO38 = BRKIN. BRKOUT is a logical OR of the ARM ICE DBGACK signal, video accelerator HVBRKO and audio accelerator. BRKIN is an external condition that can be used to force the ARM, video accelerator and audio accelerator to enter debug state.

LCDnDIF LCD/display interface selection. During power-on reset, this takes the value of LCDnDIF. After reset, the default display selection can be changed by programming this bit.

0: LCD is selected 1: master display interface (MDIF) is selected

SDMCPDEN SDMC pull-down enable on data bus (SDRDQ[15:0]).

0: disable 1: enable

SDMCSTR SDMC I/O strength. Programs the output impedance of the SDMC IOs: SDMC IO output impedance is:

00: : 40 ohms 01: 60 ohms

10: 80 ohms 11: 80 ohms

DIFCLCDHiZ DIF/CLCD I/O high impedance enable.

0: outputs enabled (default after reset).

1: outputs (and bi-dir) are forced to high impedance. This allows an external bus master to get ownership of the smart panel or CLCD panel bus in place of the STn8815.

TVOEN TVO enable. Directly enables the TVO outputs on the FSMC I/Os.

0: TVO to FSMC IOs disabled (default after reset) 1: TVO to FSMC IOs enabled.

DEEPDIS Deep-sleep mode disable. Disables deep-sleep mode.

0: automatic transition from sleep to deep-sleep mode, switching off VDD1 intra-chip domain supply.

1: no transition from sleep to deep-sleep mode. The whole chip remains supplied with power.

SDMCHLD SDRAM controller hold in self-refresh. Set to 1 by hardware when the system enters sleep or deep-sleep mode. Writing 1 releases the SDRAM forced state during sleep mode. Writing 0 has no effect.

0: SDCSnot[1:0] and SDCKEN[3:0] are controlled by the SDRAM controller, SDRAMs are no longer in self-refresh mode.

1: SDCSnot[1:0] and SDCKEN[3:0] are inactive and SDRAMs are in self-refresh mode.

IOFORCE I/Os forced during sleep mode. Set to 1 by hardware when the system enters sleep or deep-sleep mode. Must be written by software with 1 to release the I/Os' forced state during sleep mode. Writing 0 has no effect. FSMC I/Os are released immediately after a wake-up from deep-sleep is detected, thus allowing the CPU to restart execution from the external Flash memory before clearing this bit.

VDDOKMOD VDDOK pin mode. Selects the operating mode of the VDDOK pin on a high-to-low transition.

0: interrupt mode: VDDOK falling edge triggers an interrupt to ask the CPU to enter sleep mode through software.

1: automatic mode: VDDOK falling edge triggers a hardware process that forces the device into sleep mode and the SDRAM into self-refresh mode.

In automatic sleep mode entry, the CPU data cache content is not written back to the external memory, nor the SDRAM write buffers, so data may be lost when this automatic sleep entry mode is selected.

BATOKMOD BATOK pin mode. Selects the operating mode of the BATOK pin on a high-to-low transition.

0: interrupt mode: BATOK falling edge triggers an interrupt to ask the CPU to enter sleep or deep-sleep mode through software.

1: automatic mode: BATOK falling edge triggers a hardware process that forces the device into sleep or deep-sleep mode and the SDRAM into self-refresh mode.

In automatic sleep mode entry, the CPU data cache content is not written back to the external memory, nor the SDRAM write buffers, so data may be lost when this automatic sleep entry mode is selected.

0 Reserved for future use. Reading returns 0. Must be written with 0.

PMU_STA PMU status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NASRC[6:0]							0	0	0	0	0	0	0	0	0
R							R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOFT SLEEP	VDD SLEEP	BAT SLEEP	0	0	0	0	0	0	0	0	0	0	GPWK STA	VDD STA	BAT STA
RH	RH	RH	R	R	R	R	R	R	R	R	R	R	RH	R	R

Address: PMU base address + 0x0004

Reset: 0xX000 X00X

Description: Provides status appropriate to power management.

NASRC[6:0] PVT code. Binary encoded value of process voltage temperature internally delivered by the compensation cell for the I/O active slew rate control.

SOFTSLEEP Software sleep status. Indicates if the chip is placed in sleep mode by software.
0: not in sleep mode 1: in sleep mode

VDDSLEEP VDD failure sleep status. A VDDOK input pin fall causes the chip to enter sleep mode.This status bit is not updated when the chip is in sleep mode.
0: no VDDOK fall since sleep mode last exited 1: VDDOK fall caused chip to enter sleep mode

BATSLEEP Battery fault sleep status.
0: no BATOK fall since last exit of sleep mode 1: BATOK fall caused chip to enter sleep mode

GPWKSTA Wake-up from GPIO status. Indicates if the last wake-up event was triggered by an enabled GPIO, or from a reset or an RTC wake-up event. This bit is cleared by hardware when read and when an entry into sleep mode is requested.
0: triggered by reset or RTC wake-up event 1: triggered by an enabled GPIO

VDDSTA VDDOK status. Reflects the level on the VDDOK pin.
0: low 1: high

BATSTA BATOK status. Reflects the level on the BATOK pin.
0: low 1: high

0 Reserved for future use. Reading returns 0. Must be written with 0.

PMU_CTRL2 PMU control register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP FRZ	COMP TYP	DRAM COMP FRZ	DRAM COMP TYP	BYP DIV2	BYP DIV1	BYP PLL2	0	0	0	IOLVL F	IOLVL E	IOLVL D	IOLVL C	IOLVL B	IOLVL A
RW	RW	RW	RW	RW	RW	RW	R	R	R	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WKUP_TEMPO								TEMP OEN	MSP0 WR	U2MSP 1	TCKDI R	TFSDI R	MSP BYPEN	0	0
RW								RW	RW	RW	RW	RWH	RW	R	R

Address: PMU base address + 0x0008

Reset: 0x0000 0000

Description: Controls the power manager.

COMPFRZ Compensation freeze. Freezes the compensation code value, regardless of the state of the compensation process. To ensure it freezes after the compensation process terminates, software must wait for *PMU_STA2*.COMPOK to be high.
0: not frozen, self-updating (default) 1: frozen to its current code value

COMPTYP Compensation typical (not implemented). Forces the compensation code value to its typical value 0001111, regardless of the state of the compensation process.
0: compensation code self-updating (default) 1: compensation code forced to its typical value

DRAMCOMPFRZ DRAM I/Os compensation freeze. Freezes the compensation code value, regardless of the state of the compensation process. To ensure it freezes after the compensation process terminates, software must wait for *PMU_STA2*.COMPOK to be high.
0: not frozen, self-updating (default) 1: frozen to its current code value

DRAMCOMP TYP DRAM I/Os compensation typical. Forces the compensation code value to its typical value 0001111, regardless of the state of the compensation process.

0: compensation code self-updating (default) 1: compensation code forced to its typical value

BYPDIV1 Bypass DIV1. Enables bypassing of the 8/12 divisor, making the 72-78 MHz peripheral clock equal to the divider input. This bit must only be used in conjunction with BYPPLL2, as the logic is not designed to run at a higher speed than 72-78 MHz.

0: disabled, 72-78 MHz clock is 72-78 MHz 1: enabled

BYPDIV2 Bypass DIV2. Enables bypassing of the 18/13 divisor, making the 48 MHz peripheral clock equal to the divider input. This bit must only be used in conjunction with BYPPLL2, as the logic is not designed to run at a higher speed than 48 MHz.

0: disabled, 48 MHz clock is 48 MHz 1: enabled

BYPPLL2 Bypass PLL2. Enables bypassing of PLL2, so that all clocks derived from PLL2 output are derived from MXCLK (typically 19.2 MHz).

0: disabled, clocks derived from PLL2 (default). 1: enabled, clocks are derived from MXCLK.

IOLVL F-A I/O power level, domain F - A. Controls the I/O power level for domains A to F.

0: I/O power supply is 1.8 V (default) 1: I/O power supply is 2.5 V

TFSDIR TFS direction. Defines frame sync signal direction when in MSP bypass mode and MSPBYPEN = 1.

0: MSPTFS1 is an input, MSPTFS2 an output, signal is routed from MSPTFS1 to MSPTFS2.

1: MSPTFS2 is an input, MSPTFS1 an output, signal is routed from MSPTFS2 to MSPTFS1.

MSP0WR MSP0 signal wiring. Defines MSP0 module connection to the signals from the pins.

0: all MSP0 signals are available on pins (default).

1: MSP0 connected as MSP2 (MSPTFS_in connected to MSPRFS_in, MSPTCK_in to MSPRCK_in.

U2MSP1 UART2 or MSP1 muxing. The default value (0) keeps full pinout compatibility with STn8810.

0: UART2 is mapped on GPIO[36:39] alt A pins 1: MSP1 is mapped on GPIO[36:39] alt A pins

TCKDIR TCK direction. Defines the clock signal direction when in MSP bypass mode and MSPBYPEN = 1.

0: MSPTCK1 is an input and MSPTCK2 an output, signal is routed from MSPTCK1 to MSPTCK2.

1: MSPTCK2 is an input and MSPTCK1 an output, signal is routed from MSPTCK2 to MSPTCK1.

MSPBYPEN MSP bypass enable. Enables MSP1 I/O to MSP2 I/O redirection (bypass mode) in deep-sleep mode.

0: MSP1 to MSP2 I/O bypass mode disabled (default)1: MSP1 to MSP2 I/O bypass mode enabled

WKUP_TEMPO Wake-up tempo. Gives number of 32 kHz clock cycles to wait before closing switches after PWREN is sent to power management companion chip. Useful if the chip used does not provide VDDOK signal.

x00: tempo is equal to one 32 kHz cycle (31 us), x01: tempo is equal to 2 * 32 kHz cycles (62 us),

xXX: tempo is equal to (XX+1)*32 kHz cycles until xFF: tempo is equal to 256 32 kHz cycles (8 ms).

TEMPOEN Tempo enable. Enables the tempo feature for wake-up without VDDOK.

0: tempo feature is disabled (default) 1: tempo feature is enabled

PMU_STA2 PMU status register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRAMNASRC[6:0]							0	0	0	0	0	0	0	0	0
R							R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	DRAMCOMPOK	COMPOK
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: PMU base address + 0x000C

Reset: 0xXX00 0000

Description:

DRAMNASRC [6:0] PVT code for SDRAM I/Os. Binary encoded value of process voltage temperature internally delivered by the compensation cell for the I/O active slew rate control of the DDR-SDRAM I/Os.

DRAMCOMPOK Compensation OK for SDRAM I/Os. When high, shows that the compensation code for the active slew rate control of the SDRAM IO is stable.
0: unstable 1: stable

COMPOK Compensation OK for other I/Os. When high, shows that the compensation code for the active slew rate control of the other I/Os is stable.
0: unstable 1: stable

0 Reserved for future use. Reading returns 0. Must be written with 0.

PMU_RIS PMU raw interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PMURIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: PMU base address + 0x0028

Reset: 0x0000 0000

Description:

PMURIS Raw interrupt status bit. Gives the raw interrupt state from VDDOK or BATOK (the BATVDDINTR interrupt).
0: BAT/VDD failure detected during automatic request (no VIC interrupt request).

0 Reserved for future use. Reading returns 0. Must be written with 0.

PMU_ICR PMU interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PMUIC
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Address: PMU base address + 0x002C

Reset: 0x0000 0000

Description:

PMUIC Interrupt clear.
0: no effect 1: clears the BATVDDINTR interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

PMU_PeriphID0 PMU peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: PMU base address + 0xFE0

Reset: 0x0000 0004

Description: PartNumber0 reads back as 0x04.

PMU_PeriphID1 PMU peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: PMU base address + 0xFE4

Reset: 0x0000 0008

Description: Designer0 reads back as 0x0, PartNumber1 reads back as 0x8.

PMU_PeriphID2 PMU peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: PMU base address + 0xFE8

Reset: 0x0000 0028

Description: Revision reads back as 0x2, Designer1 reads back as 0x8.

PMU_PeriphID3 PMU peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: PMU base address + 0xFEC

Reset: 0x0000 0000

Description: Configuration reads back as 0x00.

PMU_PCellID0 PMU PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PMUPCellID0							
R	R	R	R	R	R	R	R	R							

Address: PMU base address + 0xFF0

Reset: 0x0000 000D

Description: PMUPCellID0 reads back as 0x0D.

PMU_PCellID1 PMU PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PMUPCellID1							
R	R	R	R	R	R	R	R	R							

Address: PMU base address + 0xFF4

Reset: 0x0000 00F0

Description: PMUPCellID1 reads back as 0xF0.

PMU_PCellID2 PMU PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PMUPCellID2							
R	R	R	R	R	R	R	R	R							

Address: PMU base address + 0xFF8

Reset: 0x0000 0005

Description: PMUPCellID2 reads back as 0x05.

PMU_PCellID3 PMU PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PMUPCellID3							
R	R	R	R	R	R	R	R	R							

Address: PMU base address + 0xFFC

Reset: 0x0000 00B1

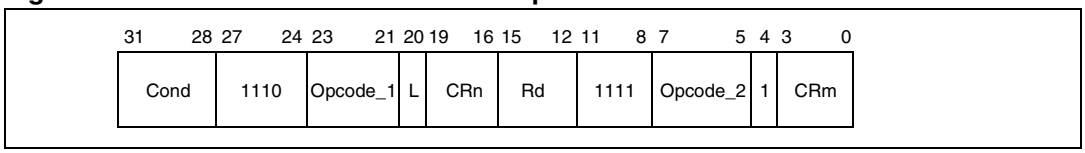
Description: PMUPCellID3 reads back as 0xB1.

4 ARM926EJ

4.1 ARM926EJ register addressing

CP15 registers can only be accessed with MRC and MCR instructions in a privileged mode. CDP, LDC, STC, MCRR and MRRC instructions, and unprivileged MRC or MCR instructions to CP15, cause an undefined instruction exception.

Figure 1. MRC and MCR instruction bit pattern



The CRn field specifies the coprocessor register to access. The CRm and opcode_2 fields specify a particular action when addressing registers or shadow registers.

The assembler for these instructions is:

```
MCR{cond} p15,Opcode_1,Rd,CRn,CRm,Opcode_2
MRC{cond} p15,Opcode_1,Rd,CRn,CRm,Opcode_2
```

4.2 ARM926EJ register summary

The system control coprocessor (CP15) configures and controls the ARM926EJ processor, caches, memory management units (MMU) and other system options.

Table 5. ARM926EJ register list

Register	Description		Page
	Reads	Writes	
CP15 r0(1)	ID code	Unpredictable	35
	Cache type		36
	TCM status		36
CP15 r1	Control	Control	37
CP15 r2	Translation table base (TTB)	Translation table base	39
CP15 r3	Domain access control	Domain access control	39
CP15 r4	Unpredictable	Ignored	N/A
CP15 r5(1)	Fault status (FSR)	Data or instruction fault status	40
CP15 r6	Fault address (FAR)	Fault address	40
CP15 r7	Cache operation	Cache operations	41
CP15 r8	TLB operations	TLB operations	43
CP15 r9(2)	Cache lockdown	Cache lockdown	44
	TCM region	TCM region	46
CP15 r10	TLB lockdown	TLB lockdown	47

Table 5. ARM926EJ register list (continued)

Register	Description		Page
	Reads	Writes	
r11	Unpredictable	Ignored	N/A
r12	Unpredictable	Ignored	N/A
CP15 r13[1)]	Fast context switch extension (FCSE) PID	FCSE PID or context ID	48
	Context ID		49
r14	Unpredictable	Ignored	N/A
r15	Test configuration	Test configuration	N/A

1. Registers 0, 5, and 13 provide access to more than one register, depending on the Opcode_2 field value.
2. Register 9 provides access to more than one register, depending on the CRm field value.

Abbreviations

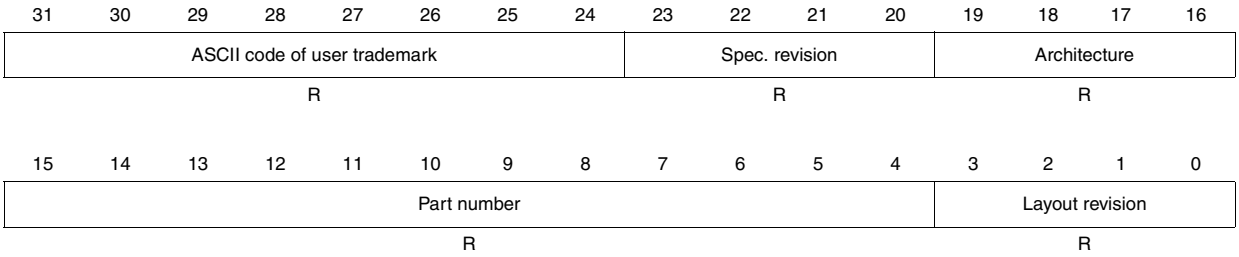
The following terms and abbreviations are used throughout the CP15 register descriptions.

- Unpredictable (UNP): reading from such a location returns data of unpredictable value. Writing to this location causes unpredictable behavior or an unpredictable change in device configuration.
- Undefined (UND): any access to such registers makes ARM926EJ take the undefined instruction trap.
- Should be zero (SBZ): all bits written to this field must be 0.
- Should be one (SBO): all bits written to this field must be 1.
- Virtual address (VA): data or instruction.
- Ignored: writing to such a location does not affect the system behavior.

Reading or writing CP15 register data, including unpredictable or SBZ fields, causes no permanent damage to the ARM926EJ processor. All CP15 register bits that are defined and contain a state are set to 0 by a reset.

4.3 ARM926EJ register descriptions

CP15 r0 ID code register



Reset: 0x4106 9260

Description: Returns the 32-bit device ID code. It is accessed by reading CP15 r0 with the Opcode_2 field set to any value other than 1 or 2. For example:

```
MRC p15, 0, Rd, c0, c0, {0, 3-7}; returns ID
```

ASCII code of user trademark returns 0x41.

Spec. revision returns 0x0.

Architecture returns 0x6 (ARMv5TEJ).

Part number returns 0x926.

Layout revision returns 0x0.

CP15 r0 Cache type register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15
0	0	0	Cache Type				S	0	0	DCacheSize				DCacheAssoc		
R	R	R	R				R	R	R	R						

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DM	DCacheLine		0	0	ICacheSize				ICacheAssoc			IM	ICacheLine	
R	R		R	R	R							R	R	

Reset: 0x1D15 2152

Description: Returns the cache size and architecture. It is accessed by reading CP15 r0 with the Opcode_2 field set to 1. For example:

```
MRC p15, 0, Rd, c0, c0, 1; returns Cache Type
```

Cache Type returns 0xE.

S returns 1.

DCacheSize returns 0x5 (16 Kbyte).

DCacheAssoc returns 010 (4-way).

DM returns 0.

DCacheLine returns 10 (8 words).

ICacheSize returns 0x05 (16 Kbyte).

ICacheAssoc returns 010 (4-way).

IM returns 0.

ICacheLine returns 10 (8 words).

CP15 r0 TCM status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DTCM Present
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ITCM Present
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Description: Returns the physical size and organization of the tightly coupled memories (TCM). It is accessed by reading CP15 r0 with the Opcode_2 field set to 2. For example:

```
MRC p15, 0, Rd, c0, c0, 2; returns TCM Status
```

DTCM present returns 0.

ITCM present returns 1.

CP15 r1 Control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBO	SBZ	SBO
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L4	RR	V	I	SBZ	SBZ	R	S	B	SBO	SBO	SBO	SBO	C	A	M
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Description: Specifies the ARM926EJ processor configuration used to enable and disable the caches and MMU. It is recommended that the register be written using a read-modify-write routine to provide the greatest future compatibility. The CRm and opcode_2 fields are not used and must be zero.

Reading from CP15 r1 reads the control bits, writing sets the control bits. For example:

```
MRC p15, 0, Rd, c1, c0, 0; read control register
MCR p15, 0, Rd, c1, c0, 0; write control register
```

SBZ Reserved. Returns an unpredictable value. When written, must be zero, or a value read from bits [31:19] on the same processor.

SBO Reserved. Returns 1. When written, must be one, or a value read from bits [31:19] on the same processor.

L4 Configures if load instruction to the PC sets the T bit. For details, see the *ARM Architecture Reference Manual*.

0: sets the T bit 1: does not set the T bit

RR Replacement strategy for ICache and DCache.

0: random replacement 1: round-robin replacement

V Location of exception vectors.

0: 0x0000 0000 to 0x0000 001C 1: 0xFFFF_0000 to 0xFFFF_001C

I ICache enable or disable.

0: ICache disabled 1: ICache enabled

R ROM protection. Modifies the MMU protection system (see domain access control).

S System protection. Modifies the MMU protection system.

0: MMU protection disabled 1: MMU protection enabled

B Endianness.

0: little-endian operation 1: big-endian operation (reserved, do not use)

C DCache enable or disable.

0: DCache disabled 1: DCache enabled

A Alignment fault enable or disable.
0: data address alignment fault checks disabled 1: data address alignment fault checking enabled

M MMU enable or disable. Care must be taken if the translated address differs from the non-translated address, because the instructions following MMU enabling are fetched using no address translation. Enabling the MMU has delayed execution. A similar situation occurs when the MMU is disabled.
0: MMU disabled 1: MMU enabled

Effects of the control register on caches

The control register bits M, C, I and RR directly affect the ICache and DCache.

Table 6. Effect of control register on caches

Cache	MMU	Behavior
ICache disabled	Enabled or disabled	All instruction fetches are from external memory (AHB).
ICache enabled	Disabled	All instruction fetches are cachable, with no protection checks. All addresses are flat mapped, that is VA = MVA = PA.
ICache enabled	Enabled	Instruction fetches are cachable or non-cachable, and protection checks are performed. All addresses are remapped from VA to PA, depending on the MMU page table entry, that is VA translated to MVA, MVA remapped to PA.
DCache disabled	Enabled or disabled	All instruction accesses are from external memory (AHB).
DCache enabled	Disabled	All data accesses are non-cachable non-bufferable. All addresses are flat mapped, that is VA = MVA = PA.
DCache enabled	Enabled	All data accesses are cachable or non-cachable, and protection checks are performed. All addresses are remapped from VA to PA, depending on the MMU page table entry. That is, VA is translated to MVA and MVA is remapped to PA.

If either the DCache or the ICache is disabled, its contents are not accessed. If the cache is subsequently re-enabled, the content will not have changed.
To guarantee that memory coherency is maintained, the DCache must be cleaned of dirty data before it is disabled.

CP15 r2 Translation table base (TTB) register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14
Translation table base																	
RW																	

13	12	11	10	9	8	7	6	5	4	3	2	1	0
SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ	SBZ
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Description: Register r2 is the translation table base of the first-level translation table base address.

Reading from CP15 r2 returns the pointer to the currently active first-level translation table in bits [31:14] and an unpredictable value in bits [13:0]. The CRm and opcode_2 fields should be zero when CP15 r2.

Writing to CP15 r2 updates the pointer to the first-level translation table from the value in bits [31:14] of the written value. Bits [13:0] must be zero. The CRm and opcode_2 fields should be zero when writing CP15 r2. For example:

```
MRC p15, 0, Rd, c2, c0, 0; read TTB register
MCR p15, 0, Rd, c2, c0, 0; write TTB register
```

CP15 r3 Domain access control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D15		D14		D13		D12		D11		D10		D9		D8	
RW		RW		RW		RW		RW		RW		RW		RW	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D7		D6		D5		D4		D3		D2		D1		D0	
RW		RW		RW		RW		RW		RW		RW		RW	

Description: Reading CP15 r3 returns the value of the domain access control register.

Writing to CP15 r3 writes the value of domain access control register. For example:

```
MRC p15, 0, Rd, c3, c0, 0; read domain access permissions
MCR p15, 0, Rd, c3, c0, 0; write domain access permissions
```

D[0:15] Domain access control.

00: No access. Any access generates a domain fault.

01: Client. Accesses are checked against the access permission in the section or page descriptor.

10: Reserved. Same as the no access mode.

11: Manager. Accesses are not checked against the access permission bits so a permission fault cannot be generated.

CP15 r5 Fault status register (FSR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SBZP	SBZP	SBZP	SBZP	SBZP	SBZP	SBZP	SBZP	SBZP	SBZP	SBZP	SBZP	SBZP	SBZP	SBZP	SBZP
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SBZP	SBZP	SBZP	SBZP	SBZP	SBZP	SBZP	0	Domain				Status			
RW	RW	RW	RW	RW	RW	RW	RW	RW				RW			

Description: Contains the source of the last instruction or data fault. It is updated for alignment faults and external aborts that occur while the MMU is disabled.
If Opcode_2 = 0, FSR = data fault status register (DFSR),
If Opcode_2 = 1, FSR = instruction fault status register (IFSR).

The FSR is accessed using the following instructions:

```
MRC p15, 0, Rd, c5, c0, 0; read DFSR
MCR p15, 0, Rd, c5, c0, 0; write DFSR
MRC p15, 0, Rd, c5, c0, 1; read IFSR
MCR p15, 0, Rd, c5, c0, 1; write IFSR
```

SBZP Reserved. When read, it returns an unpredictable value. When written, it must be zero or preserved.

Domain Specifies which of the 16 domains (D15-D0) was being accessed during a instruction or data fault.

Status Type of fault generated. Refer to *Table 7* for the encoding used for the status field.

Table 7. Status values

Priority	Source	Size	Status	Domain
Highest	Alignment	-	00X1	Invalid
	External abort on translation	First level Second level	1100 1110	Invalid Valid
	Translation	Section Page	0101 0111	Invalid Valid
	Domain	Section Page	1001 1011	Valid Valid
	Permission	Section Page	1101 1111	Valid Valid
Lowest	External abort	Section Page	1000 1010	Valid Valid

CP15 r6 Fault address register (FAR)

Description: The FAR contains the modified virtual address of the access being attempted when a data abort occurred. It is only updated for data aborts, not for prefetch aborts. It is updated for alignment faults and external aborts that occur while the MMU is disabled.

You can access the FAR using the following instructions:

```
MRC p15, 0, Rd, c6, c0, 0; read FAR
MCR p15, 0, Rd, c6, c0, 0; write FAR
```

The CRm and Opcode_2 fields must be zero when reading or writing CP15 r6.

CP15 r7 Cache operation register

Description: Register r7 controls the caches and the write buffer. The function of each cache operation is selected by the Opcode_2 and CRm fields in the MCR instruction used to write to CP15 r7. Writing other Opcode_2 or CRm values is unpredictable. Reading from CP15 r7 is unpredictable, with the exception of the two test and clean operations.

You can use the following instruction to write to r7:

```
MCR p15, 0, Rd, c7, CRm, Opcode_2; write CP15 r7
```

Table 8. Cache functions provided by CP15 r7

Function	Description
Invalidate cache	Invalidates all cache data, including any dirty data.
Invalidate single entry using index or modified virtual address	Invalidates a single cache line, discarding any dirty data.
Clean single data entry using index or modified virtual address	Writes the specified DCache line to main memory if the line is marked valid and dirty. The line is marked as not dirty. The valid bit is unchanged.
Clean and invalidate single data entry using index or modified virtual address	Writes the specified DCache line to main memory if the line is marked valid and dirty. The line is marked not valid.
Test and clean DCache	Tests a number of cache lines, and cleans one of them if any are dirty. Returns the overall dirty state of the cache in bit 30. See *Clean operations on page 42*.
Test, clean, and invalidate DCache	As for test and clean, except that when the entire cache has been tested and cleaned, it is invalidated. See *Clean operations*.
Pre-fetch ICache line	Performs an ICache lookup of the specified modified virtual address. If the cache misses, and region is cachable, a line-fill is performed.
Drain write buffer	This instruction acts as an explicit memory barrier. It drains the content of the write buffers of all memory stores occurring in program order before this instruction is completed. No instructions occurring in program order after this instruction are executed until it completes. This can be used when timing of specific stores to the level two memory system has to be controlled (for example, when a store to an interrupt acknowledge location has to complete before interrupts are enabled).
Wait for interrupt	This instruction drains the contents of the write buffers, puts the processor into a low-power state, and stops it executing further instructions until an interrupt (or debug request) occurs. When an interrupt does occur, the MCR instruction completes and the IRQ or FIQ handler is entered as normal. The return link in R14_irq or R14_fiq contains the address of the MCR instruction plus eight, so that the normal instruction used for interrupt return (`SUBS PC,R14,#4`) returns to the instruction following the MCR.

Table 9. Cache operation functions and associated data and instruction formats for r7

Function or operation	Data format	Instruction
Invalidate ICache and DCache	SBZ	MCR p15, 0, Rd, c7, c7, 0
Invalidate ICache	SBZ	MCR p15, 0, Rd, c7, c5, 0
Invalidate ICache single entry (MVA)	MVA	MCR p15, 0, Rd, c7, c5, 1
Invalidate ICache single entry (Set or Way)	Set or Way	MCR p15, 0, Rd, c7, c5, 2
Pre-fetch ICache line (MVA)	MVA	MCR p15, 0, Rd, c7, c13, 1
Invalidate DCache	SBZ	MCR p15, 0, Rd, c7, c6, 0
Invalidate DCache single entry (MVA)	MVA	MCR p15, 0, Rd, c7, c6, 1
Invalidate DCache single entry (Set or Way)	Set or Way	MCR p15, 0, Rd, c7, c6, 2
Clean DCache single entry (MVA)	MVA	MCR p15, 0, Rd, c7, c10, 1
Clean DCache single entry (Set or Way)	Set or Way	MCR p15, 0, Rd, c7, c10, 2
Test and clean DCache	-	MCR p15, 0, Rd, c7, c10, 3
Clean and invalidate DCache entry (MVA)	MVA	MCR p15, 0, Rd, c7, c14, 1
Clean and invalidate DCache entry (Set or Way)	Set or Way	MCR p15, 0, Rd, c7, c14, 2
Test, clean, and invalidate DCache	-	MCR p15, 0, Rd, c7, c14, 3
Drain write buffer	SBZ	MCR p15, 0, Rd, c7, c10, 4
Wait for interrupt	SBZ	MCR p15, 0, Rd, c7, c0, 4

Figure 2. MVA format for Rd for CP15 r7 MCR operations

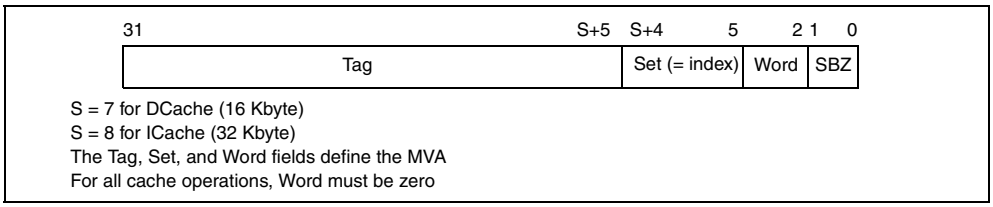
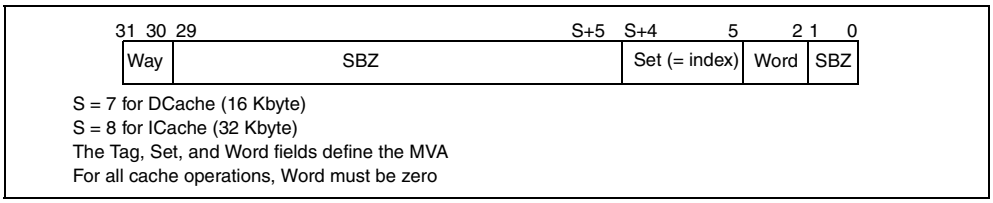


Figure 3. Set and Way format for Rd for CP15 r7 MCR operations



Clean operations

The clean DCache instruction provides an efficient way to clean the entire DCache using a simple loop. See the MRC instruction entry in the *ARM Architecture Reference Manual* for details of how the condition code flag bits are updated. If the cache contains any dirty lines, bit 30 is set to 0. If the cache contains no dirty lines, bit 30 is set to 1. This means that the following loop can clean the entire DCache:

```
tc_loop:MRC p15, 0, r15, c7, c10, 3; test and clean
  BNE tc_loop
```

The following loop cleans and invalidates the entire DCache:

```
tci_loop:MRC p15, 0, r15, c7, c14, 3; test, clean and
invalidate
   BNE tci_loop
```

CP15 r8 TLB operations register

Description: Controls the translation look aside buffer (TLB). A single TLB holds both data and instruction entries. It is divided into two parts:

- a set associative part,
- a fully associative part.

The fully-associative part (also referred to as the lockdown part of the TLB) is used to store entries to be locked down. Entries held in the lockdown part of the TLB are preserved during an invalidate TLB operation. Entries can be removed from the lockdown TLB using an invalidate TLB single entry operation.

Six TLB operations are defined, and the function to be performed is selected by the Opcode_2 and CRm field in the MCR instruction used to write CP15 r8. Writing other Opcode_2 or CRm values is unpredictable. Reading from CP15 r8 is unpredictable.

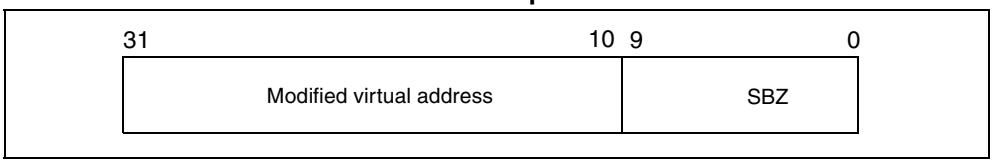
Table 10. TLB operation functions and associated data and instruction formats for r8

Function or operation	Data format	Instruction
Invalidate set-associative TLB	SBZ	MCR p15, 0, Rd, c8, c7, 0
Invalidate TLB single entry	SBZ	MCR p15, 0, Rd, c8, c7, 1
Invalidate set-associative TLB	SBZ	MCR p15, 0, Rd, c8, c5, 0
Invalidate TLB single entry (MVA)	MVA	MCR p15, 0, Rd, c8, c5, 1
Invalidate set-associative TLB	SBZ	MCR p15, 0, Rd, c8, c6, 0
Clean TLB single entry	MVA	MCR p15, 0, Rd, c8, c6, 1

The invalidate TLB operations invalidate all non-preserved entries in the TLB. The invalidate TLB single entry operations invalidate any TLB entry corresponding to the modified virtual address given in the Rd, regardless of its preserved state.

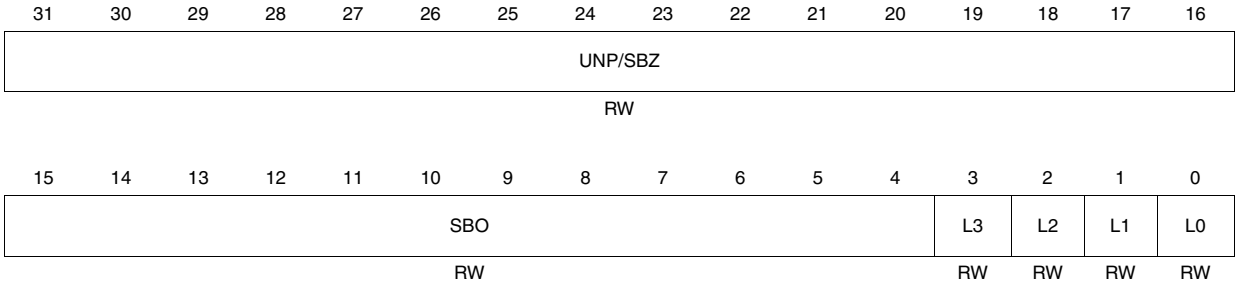
The Tag, Set, and Word fields define the MVA. For all cache operations, Word must be zero.

Figure 4. MVA Rd format for the CP15 r8 MCR operations



Note: If either small or large pages are used, and these pages contain sub-page access permissions that are different, then you must use four invalidate TLB single entry operations with the MVA set to each sub-page. This invalidates all information related to that page held in a TLB.

CP15 r9 Lockdown cache register



Reset: 0000 FFF0

Description: Register r9 accesses the lockdown cache when CRm = c0.

The register uses a cache way based locking scheme (format C) that enables control of each cache way independently. It controls which way, of the four-way cache, is used for the allocation on a line-fill.

When the registers are defined, subsequent line-fills are only placed in the specified way. This gives control over cache pollution caused by particular applications and provides a traditional lockdown operation for locking critical code into the cache. A locking bit for each way of the four-way associative cache can be locked, ensuring that normal cache line replacement is performed.

The first four bits of this register determine the L bit for the associated cache way.

If no ways have L bits set to 0, then way 3 is used for all line-fills.

- Opcode_2 = 0: selects the DCache lockdown register,
- Opcode_2 = 1: selects the ICache lockdown register.

Table 11. TLB operation functions and associated data and instruction formats for r9

Function or operation	Data format	Instruction
Read DCache lockdown register	L bit	MRC p15, 0, Rd, c9, c0, 0
Write DCache lockdown register	L bit	MCR p15, 0, Rd, c9, c0, 0
Read ICache lockdown register	L bit	MRC p15, 0, Rd, c9, c0, 1
Write ICache lockdown register	L bit	MCR p15, 0, Rd, c9, c0, 1

The cache lockdown register can only be modified using a read-modify-write sequence. For example, this sequence sets the L bit to 1 for way 0 of the ICache:

```
MRC p15, 0, Rn, c9, c0, 1;
ORR Rn, Rn, 0x01;
```

```
MCR p15, 0, Rn, c9, c0, 1;
```

UNP/SBZ Unpredictable/should be zero. Reserved.

SBO Should be one. 0xFFF.

L3 L bit for way 3 indicates which cache way is used for allocation on a line-fill.
0: standard replacement algorithm (default after reset) 1: no allocation performed

L2 L bit for way 2 indicates which cache way is used for allocation on a line-fill.
0: standard replacement algorithm (default after reset) 1: no allocation performed

L1 L bit for way 1 indicates which cache way is used for allocation on a line-fill.
0: standard replacement algorithm (default after reset) 1: no allocation performed

L0 L bit for way 0 indicates which cache way is used for allocation on a line-fill.
0: standard replacement algorithm (default after reset) 1: no allocation performed

Procedure for locking down code and data into way i of a cache

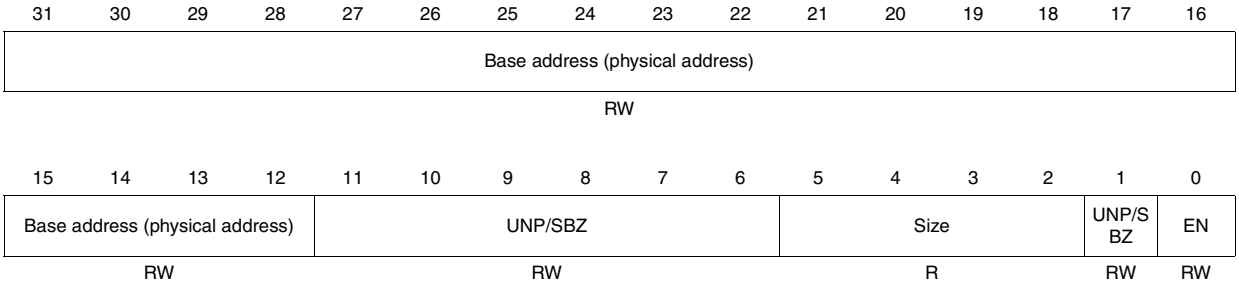
1. Ensure that no processor exceptions can occur during the execution of this procedure, for example by disabling interrupts. If this is not possible, all code and data used by any exception handlers must be treated as code and data as in steps 2 and 3.
2. If an ICache way is being locked down, ensure that all code executed by the lockdown procedure is in a non-cachable area of memory or in a locked cache way.
3. If a DCache way is being locked down, ensure that all data used by the lockdown procedure is in an non-cachable area of memory or is in a locked cache way.
4. Ensure that the data or instructions that are locked down are in a cachable area of memory.
5. Ensure that the data or instructions that are to be locked down are not already in the cache. Use the register r7 clean or invalidate operations to ensure this.
6. Write to register r9, with CRm = 0, setting Li = 0 and Lj = 1 for all other ways (j <> i). This enables allocation to the target cache way.
7. For each of the cache lines to be locked down in cache way i:
   - If a DCache is being locked down, use an LDR instruction to load a word from the memory cache line to ensure that the memory cache line is loaded into the cache.
   - If an ICache is being locked down, use the register r7 MCR prefetch ICache line (CRm = c13, Opcode_2 = 1) to fetch the memory cache line into the cache.
8. Write to register r9, CRm = 0 setting Li = 1 and restoring all the other bits to the values they had before the lockdown routine was started.

Procedure for unlocking cache

To unlock the locked down portion of the cache, write to register r9 setting L = 0 for the appropriate bit. For example, the following sequence sets L to 0 for way 0 of the ICache, unlocking way 0:

```
MRC p15, 0, Rn, c9, c0, 1;
BIC Rn, Rn, 0x01;
MCR p15, 0, Rn, c9, c0, 1;
```

CP15 r9 TCM region register



Description: Register r9 accesses the TCM region when CRm = c1.

The ARM926EJ processor supports physically-indexed, physically-tagged tightly coupled memories (TCM). The TCM status register indicates if TCM memories are attached. The TCM region minimum size is 4 Kbyte. The size of each TCM region is hardwired (ITCM size = 128 Kbyte, DTCM size = 0 Kbyte). There is one region of instruction TCM and one region of data TCM. Both the DTCM and ITCM are disabled at reset in this implementation.

Instruction fetches from the DTCM are not possible. An attempt to fetch an instruction from an address in the DTCM space does not result in an access to the DTCM; the instruction is fetched from main memory. These accesses can result in external aborts, because the address range might not be supported in main memory.

Baseaddress TCM base address is the physical address for TCM mapping. It must be aligned with the TCM size.

UNP/SBZ Unpredictable/should be zero. Reserved.

Size TCM size.

0x0: TCM absent (0 Kbyte)
0x1: reserved
0x2: reserved
0x3: 4 Kbyte
0x4: 8 Kbyte
0x5: 16 Kbyte (data TCM size)
0x6: 32 Kbyte
0x7: 64 Kbyte (instruction TCM size)
0x8: 128 Kbyte
0x9: 256 Kbyte
0xA: 512 Kbyte
0xB: 1 Mbyte
0xC to 0xF: reserved.

EN TCM enable. If either the data or instruction TCM is disabled, its contents are not accessed. If the TCM is re-enabled, the contents will not have been changed by the ARM926EJ processor.

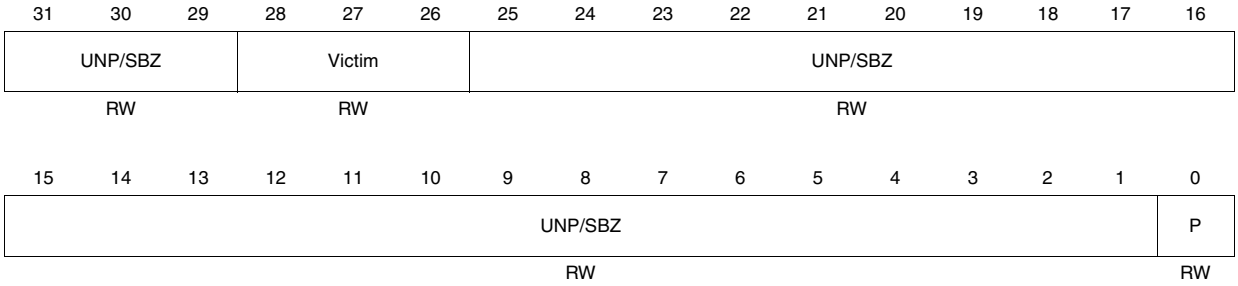
0: TCM disabled
1: TCM enabled

Table 12. TCM operation functions and associated data and instruction formats for r9

Function or operation	Data	Instruction
Read data TCM region register	Base address	MRC p15, 0, Rd, c9, c1, 0
Write data TCM region register	Base address	MCR p15, 0, Rd, c9, c1, 0
Read Instruction TCM region register	Base address	MRC p15, 0, Rd, c9, c1, 1
Write Instruction TCM region register	Base address	MCR p15, 0, Rd, c9, c1, 1

The ITCM must not be programmed to the same address as the DTCM.
If the two TCMs are of different sizes, the regions in physical memory must not overlap. If they do overlap, it is unpredictable which memory is accessed.

CP15 r10 TLB lockdown register



Description: Controls where hardware page table walks place the TLB entry. That is, in the set associative region, or the lockdown region of the TLB, and if in the lockdown region, which entry is written. The lockdown region of the TLB contains eight entries which are preserved so that invalid TLB operations only invalidate the unpreserved entries in the TLB (those in the set-associative region). Invalid TLB single entry operations invalidate any TLB entry corresponding to the modified virtual address given in Rd, regardless of their preserved state. That is, if they are in the lockdown or set-associative regions of the TLB. See register *CP15 r8* for a description of the structure of the TLB and invalid TLB operations.

It is not possible for a lockdown entry to entirely map either small or large pages, unless all the subpage access permissions are identical. Entries can still be written into the lockdown region, but the mapped address range only covers the subpage corresponding to the address that was used to perform the page table walk.

The following code sample locks down an entry to the current victim:

```
MRC p15, 0, R0, c10, c0, 0;read the lockdown register
ORR R0, R0, #1;set the preserved bit
MCR p15, 0, R0, c10, c0, 0;write to the lockdown register
ADR r1, LockAddr;set r1 to the value of the
     ;address to be locked down
MCR p15, 0, r1, c8, c7, 1;invalidate TLB single entry to
   ;ensure that LockAddr is not
   ;already in the TLB
LDR r1, [r1];TLB will miss, and entry will
   ;be loaded
MRC p15, 0, r0, c10, c0, 0;read the lockdown register
   ;(victim will have incremented)
BIC r0, r0, #1;clear preserve bit
MCR p15, 0, r0, c10, c0, 0;write to the lockdown register
```

Table 13. TLB lockdown functions and the associated instruction formats for r10

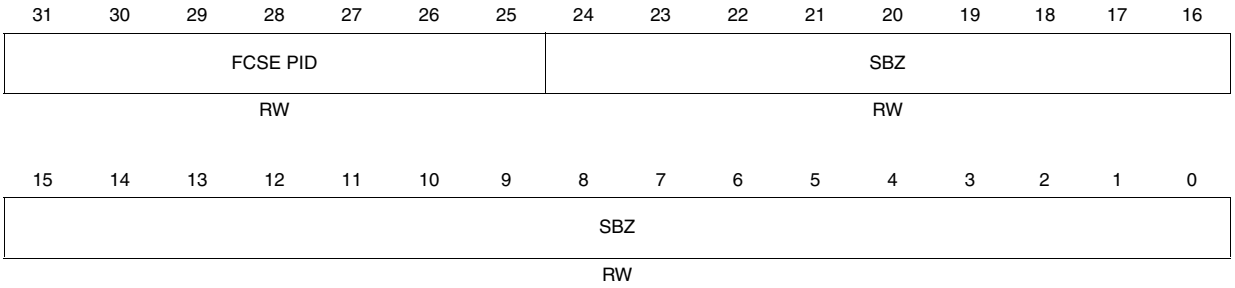
Function or operation	Instruction
Read data TLB lockdown register	MRC p15, 0, Rd, c10, c0, 0
Write data TLB lockdown register	MCR p15, 0, Rd, c10, c0, 0

UNP/SBZ Unpredictable/should be zero. Reserved.

Victim Victim automatically increments after each table walk resulting in an entry being written into the lockdown part of the TLB.

P Preserved bit indicates where subsequent hardware page walks place the TLB entry. In the:
0: set associative region of the TLB 1: lockdown region at the entry specified by victim

CP15 r13 Fast context switch extension (FCSE) PID register



Description: The FCSE PID register is selected when Opcode_2 = 0.

Addresses issued by the ARM9EJ core in the range 0 to 32 Mbyte are translated in accordance with the value contained in this register. Address A becomes A + (FCSE PID x 32 Mbyte). It is this modified address that is seen by the caches and MMU. Addresses above 32 Mbyte are not modified. The FCSE PID is a seven-bit field, enabling 128 x 32 Mbyte processes to be mapped.

If the FCSE PID is 0, there is a flat-mapping between the virtual addresses output by the ARM9EJ core and the modified virtual addresses used by the caches and MMU. The FCSE PID is set to 0 at system reset. If the MMU is disabled no FCSE address translation occurs. FCSE translation is not applied for addresses used for entry based cache of TLB maintenance operations. For these operations, VA = MVA.

A fast context switch can be performed by writing to CP15 register 13 with Opcode_2 = 0. The content of the caches and the TLB do not have to be flushed after a fast context switch because they will hold valid address tags. The two instructions after the FCSE PID has been written have been fetched with the old FCSE PID, as the following code example shows:

```
{FCSE PID = 0}
  MOV r0, #1:SHL:25;fetched with FCSE PID = 0
  MCR p15, 0, r0, c13, c0, 0;fetched with FCSE PID = 0
  A1 ;fetched with FCSE PID = 0
  A2 ;fetched with FCSE PID = 0
  A3 ;fetched with FCSE PID = 1
```

Table 14. FCSE operation functions and instructions to access CP15 r13 FCSE PID

Function or operation	Data format	Instruction
Read FCSE PID	FCSE PID	MRC p15, 0, Rd, c13, c0, 0
Write FCSE PID	FCSE PID	MCR p15, 0, Rd, c13, c0, 0

CP15 r13 Context ID register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Context Identifier, bits [31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Context Identifier, bits [15:0]															
RW															

Description: The context ID register is selected when Opcode_2 = 1.
It provides a mechanism to allow real-time trace tools to identify the currently executing process in multi-tasking environments.

Table 15. Context ID register operations

Function or operation	Data format	Instruction
Read context ID	context ID	MRC p15, 0, Rd, c13, c0, 1
Write context ID	context ID	MCR p15, 0, Rd, c13, c0, 1

5 Level 2 cache controller (L2CC)

The L2CC is controlled by a set of 32-bit memory-mapped registers that occupy a relocatable 4KB memory space starting from address L2CC_Base = 0x1021_0000.

All registers except some Test and Debug registers accept 32-bits accesses only.

The Auxiliary Control register, L2CC_ACR, must only be written with a read-modify-write type access when the cache is turned off.

Data line and tag line registers, L2CC_LDAT and L2CC_LTAG, support 64-bit transfer to enable setting or reading registers with `LDM/STM` instructions.

You must disable the L2CC by writing to the L2CC Control Register, L2CC_CR, and perform a Cache Sync operation (by writing to L2CC_CSYNC register) before writing to any of the other internal registers.

Reads to an unmapped register return 0x0.

5.1 L2CC register summary

Table 16. L2CC register list

Offset	Register name	Description	Page
0x00	L2CC_ID	L2CC ID Register	52
0x04	L2CC_TYP	L2CC Type Register	52
0x100	L2CC_CR	L2CC Control Register	53
0x104	L2CC_ACR	L2CC Auxiliary Control Register	53
0x730	L2CC_CSYNC	L2CC Cache Sync Register	55
0x770	L2CC_ILIPA	L2CC Invalidate Line by PA Register	56
0x77C	L2CC_IWAY	L2CC Invalidate by Way Register	56
0x7B0	L2CC_CLIPA	L2CC Clean Line by PA Register	57
0x7B8	L2CC_CLIWI	L2CC Clean Line by Way/Index Register	57
0x7BC	L2CC_CWAY	L2CC Clean by Way Register	58
0x7F0	L2CC_CILIPA	L2CC Clean and Invalidate Line by PA Register	57
0x7F8	L2CC_CILIWI	L2CC Clean and Invalidate Line by Way/Index Register	59
0x7FC	L2CC_CIWAY	L2CC Clean and Invalidate by Way Register	59
0x900	L2CC_LCKWD	L2CC Lockdown by Way - D Side Register	60
0x904	L2CC_LCKWI	L2CC Lockdown by Way - I Side Register	60
0xF00	L2CC_TSTOP	L2CC Test Operation Register	61
0xF10 to 0xF2C	L2CC_LDAT0.7	L2CC Line Data 0 to 7 Registers	62
0xF30	L2CC_LTAG	L2CC Line Tag Register	62
0xF40	L2CC_DBGCR	L2CC Debug Control Register	63

5.2 L2CC register descriptions

L2CC_ID L2CC ID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designer								RESERVED														PartNumber				Revision					
R								R														R				R					

Address: L2CCBaseAddress + 0x00

Type: R

Reset: 0x41000041

Description: L2CC_ID is the read-only ID Register that returns the 32-bit device ID code of the L2CC.

[31:24] **Designer:** These bits read back as 0x41

[9:6] **PartNumber:** These bits read back as 0x1

[5:0] **Revision:** These bits read back as 0x03

L2CC_TYP L2CC Type Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									Way_size_D_side			RESERVED	Associativity_D_side	RESERVED							Way_size_I_side			RESERVED	Associativity_I_side				RESERVED		
R									R			R	R	R							R			R	R				R		

Address: L2CCBaseAddress + 0x04

Type: R

Reset: 0x1C100100

Description: L2CC_TYP is the read-only register that returns the 32-bit Cache Type, which makes the cache parameter a product of cache way size and the asociativity.

[22:20] **Way_size_D_side:** These bits reflects the bits [19:17] of L2CC_ACR register

[18] **Associativity_D_side:** These bits reflects the bits [16:13] of L2CC_ACR register

[10:8] **Way_size_I_side:** These bits reflects the bits [19:17] of L2CC_ACR register

[6:3] **Associativity_I_side:** These bits reflects the bits [16:13] of L2CC_ACR register

L2CC_CR L2CC Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															L2EN
R																															R/W

Address: L2CCBaseAddress + 0x100

Type: R/W

Reset: 0x00000000

Description: L2CC_CR is the L2CC control register, and must be accessed using a read-modify-write sequence.

Note: If the L2EN bit is change while a transaction is on going in the L2CC, that transaction completes as if the enable bit did not change, so an ongoing linefill, and possible subsequent line eviction, completes even if the cache is turned off.

[0] **L2EN:** Level 2 Cache Enable bit. This bit enables or disables the L2CC:
0: L2CC cache in bypass mode (disabled),
1: L2CC cache is enabled.

L2CC_ACR L2CC Auxiliary Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							EABDIS	WRAOVR	SHATOVEN	PAREN	EVMBEN	WAYSIZE			ASSO				WRAPDIS	DIRTLAT			TAGLAT			DATWRLAT			DATRDLAT		
R							R/W	R/W	R/W	R/W	R/W	R/W			R/W				R/W	R/W			R/W			R/W			R/W		

Address: L2CCBaseAddress + 0x104

Type: R/W

Reset: 0x00000000

Description: L2CC_ACR is the L2CC auxiliary control register.

[24] **EABDIS:** Exclusive Abort Disable bit
0: L2CC sends an ERROR response back to exclusive access in a cachable, shared memory region with shared override bit set (default after reset),
1: Abort generation for exclusive access disabled. Treated as cachable non-shared accesses.

[23] **WRAOVR:** Write Allocate Override
0: Use of HPROT attributes (default after reset),
1: Override HPROT attributes. All Write-Through and Writeback accesses are read-write-allocate.

[22] **SHATOVEN:** Shared Attribute Override Enable
0: Shared accesses treated as non-cachable (default after reset),
1: Shared attribute internally ignored but still forwarded to system memory.

[21] **PAREN:** Parity Enable
0: Disabled (default after reset),
1: Enabled.

[20] **EVMBEN:** Event Bus Enable
0: Disabled (default after reset),
1: Enabled.

[19:17] **WAYSIZE:** Way Size
000: Reserved,
001: 16 KBytes (default after reset),
010: 32 KBytes,
011: 64 KBytes,
100: 128 KBytes,
101: 256 KBytes,
110: to
111: Reserved.

[16:13] **ASSO:** Associativityto
0000: Cache absent (default after reset),
0001: Direct-mapped cache,
0010: 2-way cache,
0011: 3-way cache,
0100: ,4-way cache,
0101: 5-way cache,
00110: 6-way cache,
111: 7-way cache,
1000: 8-way cache,
1001:
1111: Reserved.

[12] **WRAPDIS:** Wrap Accesses Disabled
0: AHB Master ports can perform wrap accesses (default after reset),
1: Wrap accesses requested on AHB slave ports are converted to linear accesses on AHB master ports.

[11:9] **DIRTLAT:** Latency for dirty RAMDefines the number of CPU clock-cycles used for accessing the Dirty RAM:
000: 1 cycle, i.e. no additional latency: do not use this value,
001: 2 cycles of latency,
..
111: 8 cycles of latency (default after reset)
Note: The minimum value when CPU clock is 264 MHz at 1.2V and 332MHz at 1.4V is 2 cycles of latency, so set DIRTLAT to 001b.

[8:6] **TAGLAT:** Latency for tag RAMDefines the number of CPU clock-cycles used for accessing the Tag RAM:

000: 1 cycle, i.e. no additional latency: do not use this value,

001: 2 cycles of latency,

..

111: 8 cycles of latency (default after reset)

Note: The minimum value when CPU clock is 264 MHz at 1.2V and 332MHz at 1.4V is 2 cycles of latency, so set TAGLAT to 001b.

[5:3] **DATWRLAT:** Latency for data RAM writes. Defines the number of CPU clock-cycles used for accessing the data RAM write operations:

Note: The minimum value when CPU clock is 264 MHz at 1.2V and 332MHz at 1.4V is 2 cycles of latency, so set DATWRLAT to 001b.

000: 1 cycle, i.e. no additional latency: do not use this value,

001: 2 cycles of latency,

..

111: 8 cycles of latency (default after reset),

[2:0] **DATRDLAT:** Latency for data RAM reads. Defines the number of CPU clock-cycles used for accessing the data RAM read operations:

000: 1 cycle, i.e. no additional latency: do not use this value,

001: 2 cycles of latency,

..

111: 8 cycles of latency (default after reset)

Note: The minimum value when CPU clock is 264 MHz at 1.2V and 332MHz at 1.4V is 2 cycles of latency, so set DATRDLAT to 001b.

L2CC_CSYNC L2CC Cache Sync Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
R/W																															

Address: L2CCBaseAddress + 0x730

Type: R/W

Reset: 0x00000000

Description: L2CC_CSYNC performs a L2CC Cache Sync operation when it is written. Written data is don't care, and is not memorized. Read accesses from this register always return 0x0. A Cache Sync operation performs the following operations:

- Drain write buffer (WB). and eviction buffer (EB) to System (L3) memory
- Drain write-allocate buffer (WA) to the L2CC data RAM.

Note: The Cache Sync operation is considered to be complete when the write buffer (WB), eviction buffer (EB) and write-allocate buffer (WA) are empty, regardless of on-going linefills that could generate a new eviction.

L2CC_ILIPA L2CC Invalidate Line by PA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TAGADDR									RESERVED					DATAADDR										RESERVED				
R				R									R					R										R				

Address: L2CCBaseAddress + 0x770

Type: R

Reset: 0x00000000

Description: L2CC_ILIPA register allows to invalidate the L2CC line specified by its Physical Address (PA) (the line is marked as not valid). This register is always read as 0x0.

Note: Cache maintenance operations are atomic operations (AHB bus from which this register is written to is blocked until the operation completes), and writing to the register starts the operation, on the line specified here by {Tag, Index}.

[27:19] **TAGADDR:** Tag RAM address bits. TAGADDR[8:0] are the LSBs of Tag RAM address bus.

[13:5] **DATAADDR:** Data RAM address bits. DATAADDR[8:0] are the LSBs of Data RAM address bus for the line to be invalidated.

L2CC_IWAY L2CC Invalidate by Way Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								W 7	W 6	W 5	W 4	W 3	W 2	W 1	W 0
R																								R/ W	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W	R/ W

Address: L2CCBaseAddress + 0x77C

Type: R/W

Reset: 0x00000000

Description: L2CC_IWAY register allows to invalidate all data in the specified ways including dirty data. This register is always read as 0x0.

Note: Way-based cache maintenance operations are background operation, so writing to the register starts the operation, on the ways for which bit Wn is written to 1b in L2CC_IWAY register. The Way bits Wn are reset to 0b as the respective ways are invalidated, so the L2CC_IWAY register must be polled to see when the operation has completed.

[7:0] **W[7:0]:** Way bit n (n = 0..7)Writing 0b do not invalidate data in way n, writing 1b invalidate data in way n,

Reading 0b: no operation or maintenance operation is completed on way n,

Reading 1b: maintenance operation is on going/not completed on way n.

L2CC_CLIPA L2CC Clean Line by PA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TAGADDR									RESERVED					DATAADDR									RESERVED					
R				R									R					R									R					

Address: L2CCBaseAddress + 0x7B0

Type: R

Reset: 0x00000000

Description: L2CC_CLIPA register allows to clean the L2CC line specified by its Physical Address (PA). The clean operation:

- write the specified L2CC line to main memory if the line is marked as valid and dirty
- mark the line as not dirty,valid bit is unchanged.This register is always read as 0x0.

Note: Cache maintenance operations are atomic operations (AHB bus from which this register is written to is blocked until the operation completes), and writing to the register starts the operation, on the line specified here by {Tag, Index}.

[27:19] **TAGADDR:** Tag RAM address bits. TAGADDR[8:0] are the LSBs of Tag RAM address bus for the line to be cleaned.

[13:5] **DATAADDR:** Data RAM address bits. DATAADDR[8:0] are the LSBs of Data RAM address bus for the line to be cleaned.

L2CC_CLIWI L2CC Clean Line by Way/Index Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WAY				RESERVED														DATAADDR									RESERVED					
R				R														R									R					

Address: L2CCBaseAddress + 0x7B8

Type: R

Reset: 0x00000000

Description: L2CC_CLIWI register allows to clean the L2CC line specified by its Index/Way. The clean operation:

– write the specific L2CC line within the specified way to main memory if the line is marked as valid and dirty

– mark the line as not dirty,valid bit is unchanged.This register is always read as 0x0.

Note: Cache maintenance operations are atomic operations (AHB bus from which this register is written to is blocked until the operation completes), and writing to the register starts the operation, on the line specified here by {Way, Index}.

[31:29] **WAY:** Way bits. Way number for the line to be cleaned.

[13:5] **DATAADDR:** Data RAM address bits. DATAADDR[8:0] are the LSBs of Data RAM address bus for the line to be cleaned.

L2CC_CWAY L2CC Clean by Way Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								W7	W6	W5	W4	W3	W2	W1	W0
R																								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: L2CCBaseAddress + 0x7BC

Type: R/W

Reset: 0x00000000

Description: L2CC_CWAY register allows to clean all data in the specified ways. The clean operation:

- write each line of the specified L2CC ways to main memory if the line is marked as valid and dirty
- mark the lines as not dirty,valid bits are unchanged.This register is always read as 0x0.

Note: Way-based cache maintenance operations are background operation, so writing to the register starts the operation, on the ways for which bit Wn is written to 1b in L2CC_CWAY register. The Way bits Wn are reset to 0b as the respective ways are cleaned, so the L2CC_CWAY register must be polled to see when the operation has completed.

[7:0] **W[7:0]:** Way bit n (n = 0..7)Writing 0b do not clean data in way n,
Writing 1b clean data in way n,
Reading 0b: no operation or maintenance operation is completed on way n,
Reading 1b: maintenance operation is on going/not completed on way n.

L2CC_CILIPA L2CC Clean and Invalidate Line by PA Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				TAGADDR									RESERVED					DATAADDR												RESERVED		
R				R									R					R												R		

Address: L2CCBaseAddress + 0x7F0

Type: R

Reset: 0x00000000

Description: L2CC_CILIPA register allows to clean and mark as not valid the L2CC line specified by its Physical Address (PA). The clean and invalidate operation:

- write the specified L2CC line to main memory if the line is marked as valid and dirty
- mark the line as not valid.This register is always read as 0x0.

Note: Cache maintenance operations are atomic operations (AHB bus from which this register is written to is blocked until the operation completes), and writing to the register starts the operation, on the line specified here by {Tag, Index}.

[27:19] **TAGADDR:** Tag RAM address bits. TAGADDR[8:0] are the LSBs of Tag RAM address bus for the line to be cleaned and invalidated.

[13:5] **DATAADDR:** Data RAM address bits. DATAADDR[8:0] are the LSBs of Data RAM address bus for the line to be cleaned and invalidated.

L2CC_CILIWI L2CC Clean and Invalidate Line by Way/Index Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WAY			RESERVED															DATAADDR									RESERVED				
R			R															R									R				

Address: L2CCBaseAddress + 0x7F8

Type: R

Reset: 0x00000000

Description: L2CC_CLIWI register allows to clean and invalidate the L2CC line specified by its Index/Way. The clean and invalidate operation:

– write the specific L2CC line within the specified way to main memory if the line is marked as valid and dirty

– mark the line as not valid.This register is always read as 0x0.

Note: Cache maintenance operations are atomic operations (AHB bus from which this register is written to is blocked until the operation completes), and writing to the register starts the operation, on the line specified here by {Way, Index}.

[31:29] **WAY:** Way bits. Way number for the line to be cleaned and invalidated.

[13:5] **DATAADDR:** Data RAM address bits. DATAADDR[8:0] are the LSBs of Data RAM address bus for the line to be cleaned and invalidated.

L2CC_CIWAY L2CC Clean and Invalidate by Way Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								W7	W6	W5	W4	W3	W2	W1	W0
R																								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: L2CCBaseAddress + 0x7FC

Type: R/W

Reset: 0x00000000

Description: L2CC_CIWAY register allows to clean and invalidate all data in the specified ways. The clean and invalidate operation:

– write each line of the specified L2CC ways to main memory if the line is marked as valid and dirty

– mark the lines as not valid.This register is always read as 0x0.

Note: Way-based cache maintenance operations are background operation, so writing to the register starts the operation, on the ways for which bit Wn is written to 1b in L2CC_CIWAY register. The Way bits Wn are reset to 0b as the respective ways are cleaned, so the L2CC_CIWAY register must be polled to see when the operation has completed.

[7:0] **W[7:0]:** Way bit n (n = 0..7).
Writing 0b does not clean and invalidate data in way n,
Writing 1b cleans and invalidates data in way n,
Reading 0b: no operation or maintenance operation is completed on way n,
Reading 1b: maintenance operation is on going/not completed on way n.

L2CC_LCKWD L2CC Lockdown by Way - D Side Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								W7	W6	W5	W4	W3	W2	W1	W0
R																								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: L2CCBaseAddress + 0x900

Type: R/W

Reset: 0x00000000

Description: If a cache lookup misses and a cache linefill is required, there are eight possible locations where the new line can be placed. Lockdown format C restricts the cache replacement algorithm to only use a subset of the eight possible locations. The choice of an 8-way L2CC increases hit rate and increases effectiveness of Lockdown Format C. With Lockdown Format C, a block of the L2CC can be used as a form of frame buffer.

If all ways are locked, a linefill is performed on a cache miss, reading eight words from external memory, but the cache is not updated with the linefill data. In the same way, Write-Through write allocate and Write-Back write allocate accesses are treated as normal Write-Through and Write-Back accesses respectively.

L2CC_LCKWD register allows to lock the specified ways for the Data side.

[7:0] **W[7:0]:** Way bit n (n = 0..7). Writing 0b do lock way n on data side,
Writing 1b way n on data side,
Reading 0b: way n is not locked on data side,
Reading 1b: way n is locked on data side.

L2CC_LCKWI L2CC Lockdown by Way - I Side Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								W7	W6	W5	W4	W3	W2	W1	W0
R																								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: L2CCBaseAddress + 0x904

Type: R/W

Reset: 0x00000000

Description: If a cache lookup misses and a cache linefill is required, there are eight possible locations where the new line can be placed. Lockdown format C restricts the cache replacement algorithm to only use a subset of the eight possible locations. The choice of an 8-way L2CC increases hit rate and increases effectiveness of Lockdown Format C. With Lockdown Format C, a block of the L2CC can be used as a form of frame buffer.

If all ways are locked, a linefill is performed on a cache miss, reading eight words from external memory, but the cache is not updated with the linefill data. In the same way, Write-Through write allocate and Write-Back write allocate accesses are treated as normal Write-Through and Write-Back accesses respectively.

L2CC_LCKWI register allows to lock the specified ways for the Instruction side.

[7:0] **W[7:0]:** Way bit n (n = 0..7)
Writing 0b do lock way n on instruction side,
Writing 1b way n on instruction side,
Reading 0b: way n is not locked on instruction side,
Reading 1b: way n is locked on instruction side.

L2CC_TSTOP L2CC Test Operation Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WAY			RESERVED															DATAADDR									RESERVED				nRW
R/W			R															R/W									R				R/W

Address: L2CCBaseAddress + 0xF00

Type: R/W

Reset: 0x00000000

Description: The L2CC_TSTOP Test Operation Register enables the contents of the L2CC to be read and written for test purpose.

For cache line read, L2CC_TSTOP register is written with the nRW bit cleared so that the content of the line designated by the Index/Way fields is put in the line data registers and its attributes put in the Line Tag Register. All information needed about the cache line can then be retrieved by reading via the Line Data Registers (L2CC_LDAT0..7) and Line Tag Registers (L2CC_LTAG).

For cache line writes, all line data and attributes must be written first in both the Line Data Registers (L2CC_LTAG) and Line Tag Registers. At this time, by writing to the

Test Operation Register L2CC_TSTOP, the cache line designated by the Index/Way fields is updated with register contents.

[31:29] **WAY:** Way bits. Way number for the line to be read or written in the Test Operation.

[13:5] **DATAADDR:** Data RAM address bits. DATAADDR[8:0] are the LSBs of Data RAM address bus for the line to be read or written in the Test Operation.

[0] **nRW:** Read or Write test operation. Defines the direction (read or write) of the Test Operation:
0: Read Test Operation
1: Write Test Operation

L2CC_LDAT0..7 L2CC Line Data 0 to 7 Registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINE_DATA_WORDx																															
R/W																															

Address: L2CC Base + 0xF10 + 0x4 * x, x = 0..7

Type: R/W

Reset: 0x00000000

Description: For cache line read, test operation is written with the nRW bit cleared so that the content of the line designated by the Index/Way fields is put in the line data registers and its attributes put in the Line Tag Register. All information needed about the cache line can then be retrieved by reading via the Line Data Registers (L2CC_LDAT0..7) and Line Tag Registers (L2CC_LTAG).

For cache line writes, all line data and attributes must be written first in both the Line Data Registers (L2CC_LTAG) and Line Tag Registers. At this time, by writing to the Test Operation Register, the cache line designated by the Index/Way fields is updated with register contents.

L2CC_LTAG L2CC Line Tag Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAG																		VALID	DIRT1	DIRT0	VICTIMP			RESERVED							
R																		R	R	R	R			R							

Address: L2CCBaseAddress + 0xF30

Type: R

Reset: 0x00000000

Description: For cache line read, test operation is written with the nRW bit cleared so that the content of the line designated by the Index/Way fields is put in the line data registers and its attributes put in the Line Tag Register.

All information needed about the cache line can then be retrieved by reading via the Line Data Registers (L2CC_LDAT0..7) and Line Tag Registers (L2CC_LTAG).For cache line writes, all line data and attributes must be written first in both the Line Data Registers (L2CC_LTAG) and Line Tag Registers. At this time, by writing to the Test

Operation Register, the cache line designated by the Index/Way fields is updated with register contents.

[31:14] **TAG:** Tag. An invalid line always has its tag set to zero until it has been written through a test operation.

[13] **VALID:** Valid. An invalid line always has its tag set to zero until it has been written through a test operation.

[12] **DIRT1:** Dirty 1. Defines state of the last four words in the cache line, words 4-8.

[11] **DIRT0:** Dirty 0. Defines state of the first four words in the cache line, words 0-3.

[10:8] **VICTIMP:** Victim Pointer. Defines last allocated way:..
000: way 0
111: way 7

L2CC_DBGCR L2CC Debug Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DWB	DCL
R																														R/W	R/W

Address: L2CCBaseAddress + 0xF40

Type: R/W

Reset: 0x00000000

Description: L2CC_DBGCR register forces specific cache behavior required for debug.

[1] **DWB:** Disable Write-Back
0: Enable Write-Back behavior (default after reset)
1: Force Write-Through and read-allocate only behavior.

[0] **DCL:** Disable Write-Back
0: Enable cache linefills (default after reset)
1: Disable cache linefills.

Forcing Write-Through behavior

Setting the DWB bit to 1 forces theL2CC to treat all cachable accesses as though they are in a Write-Through no write-allocate region of memory. Setting the DWB bit overrides access attributes. If the cache contains dirty cache lines, these remain dirty

while the DWB bit is set, unless they are written back because of a Write-Back eviction after a linefill, or because of an explicit clean operation. Lines that are cleaned are not marked as dirty if they are updated while the DWB is set. This functionality enables a debugger to download code or data to external memory, without the requirement to clean part or the entire cache to ensure that the code or data being downloaded has been written to external memory.

If the DWB is set, and a write is made to a cache line that is dirty, then both the cache line and the external memory are updated with the write data. Other entries in the cache line still have to be written back to main memory to achieve coherency.

Disabling cache linefills

Setting the DCL bit prevents the cache from being updated when performing a linefill on a miss. When set, a linefill is performed on a cache miss, reading eight words from external memory, but the cache is not updated with the linefill data. This mode of operation is required for debug so that the memory image, as seen by the CPU core, can be examined in a noninvasive manner. Cache hits read data words from the cache, and cache misses from a cachable region read words directly from memory.

Setting the DCL bit overrides the write-allocate attributes. Write-Through write allocate and Write-Back write allocate accesses are treated as normal Write-Through and Write-Back accesses respectively.

6 L2CC event monitor (L2EM)

The L2EM contains ten registers, shown in *Table 17*

6.1 L2EM addressing

The L2EM_Base address is 0x1021_1000.

6.2 L2EM register summary

Table 17. L2EM register list

Offset	Register name	Description	Page
0x00	L2EM_CR	L2EM Control Register	*66*
0x04	L2EM_CS	L2EM Counter Status Register	*66*
0x100 to 0x10C	L2EM_CC0..3	L2EM Counter Configuration Register 0..3	*67*
0x200 to 0x20C	L2EM_CT0..3	L2EM Counter x Register 0..3	*68*

6.3 L2EM register descriptions

L2EM_CR L2EM Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				RSTC3	RSTC2	RSTC1	RSTC0	RESERVED		ITDUR			ITPOL	ITTYPE	L2EMEN
R																				R/W	R/W	R/W	R/W	R		R/W			R/W	R/W	R/W

Address: L2EMBaseAddress + 0x00

Type: R/W

Reset: 0x00000000

Description: L2EM_CR is used to:

- enable the event monitor block,
- control interrupt generation,
- reset counters to zero.

Note: The interrupt polarity bit ITPOL must be set to allow the interrupt controler (VIC) to acknowledge the interrupt.

[11:8] **RSTC[3:0] :** Counter Reset (x = 0..3)Corresponding counter of the L2EM is reset when this bit is written with 1b. This bit is always read as 0b.

[5:3] **ITDUR :** Edge-sensitive interrupt pulse duration.
0: 1 CLK cycle
1: 2 CLK cycles
111: 128 CLK cycles

[2] **ITPOL :** Interrupt polarity
0: interrupt signal is active low (default)
1: interrupt signal is active high

[1] **ITTYPE :** Interrupt type
0: level sensitive (default). Remains active until all counter flags are cleared
1: edge sensitive

[0] **L2EMEN :** L2CC Event Monitor Enable bit
0: Event Monitor disabled (default after reset),
1: Event Monitor enabled.

L2EM_CS L2EM Counter Status Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												FLGC3	FLGC2	FLGC1	FLGC0
R																												R/W	R/W	R/W	R/W

Address: L2EMBaseAddress + 0x04

Type: R/W

Reset: 0x00000000

Description: L2EM_CS contains a set of flags that indicate if the corresponding counter flag set condition has occured. It is used to determine which counter or counters cause an interrupt if interrupt generation is enabled. The interrupt line remains active until all flags in L2EM_CS register are cleared.

[3:0] **FLGC[3:0] :** Counter Status flag (x = 0..3)When read, returns the status of corresponding Counter x status flag:Counter x Flag is cleared when written to 1b.

0: Counter Flag set condition has not occured,

1: Counter Flasg set condition has occured.

L2EM_CC0..3 L2EM Counter Configuration Register 0..3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									CxEVS					CFLGSC	CINTGEN
R																									R/W					R/W	R/W

Address: L2EM Base + 0x100 + 0x4 * x, x = 0..3

Type: R/W

Reset: 0x00000000

Description: Each counter has a configuration register, which can define:counter event source,counter Flag set condition,interrupt enable.Table 47 shows the bit assignments for L2EM_SR register.

[6:2] **CxEVS :** Counter x Event Source selection (cont.)
1100: Instruction read hit,
1101: Instruction read request,
1110: Parity error on L2CC cache data RAM read,
1111: Parity error on L2CC cache tag RAM read,
10000: Write allocate,
10001: Any L2CC cache RAM error (data or tag, read or write),
10010: Parity error on L2CC cache data or tag RAM read, ,
10011: EMC3 (Event Monitor Counter 3) overflow,
10100: EMC2 (Event Monitor Counter 2) overflow,
10101: EMC1 (Event Monitor Counter 1) overflow,
10110: EMC0 (Event Monitor Counter 0) overflow,
10111: CLK cycles (counter incremented on every CLK cycle),
11000: to
11111: Counter disabled.

[1] **CFLGSC :** Counter Flag Set Condition
0: Counter flag set on overflow (default0
1: Counter flag set on increment

[0] **CINTGEN :** Counter Interrupt Generation
0: No interrupt generated by the counter (default)
1: an interrupt is generated when counter flag set condition is met

L2EM_CT0..3 L2EM Counter x Register 0..3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMCx																															
R																															

Address: L2CC Base + 0x200 + 0x4 * x, x = 0..3

Type: R

Reset: 0x00000000

Description: L2EM Counter x Register 0..3

7 Hardware semaphore (HSEM)

7.1 HSEM register addressing

Register addresses are provided as the HSEM base address, HSEMBaseAddress, plus the register offset.

The HSEMBaseAddress is 0x101F F000.

7.2 HSEM register summary

The device communicates with the system via registers which are accessible via a 32-bit width AMBA rev. 2.0 peripheral bus (APB). The 16hardware semaphores (HSEM) perform a lock operation in a single write-access, avoiding the need of a read-modify-write bus transfer that is outside the capabilities of some programmable cores.

The HSEM manages four internal interrupt signals for each of the 16 semaphores, named HSEMINTRAx, HSEMINTRBx, HSEMINTRCx, HSEMINTRDx.

- All HSEMINTRA[0:15] internal signals are ORed together after masking to form the HSEMINTRA interrupt line usually connected to the VIC.
- All HSEMINTRB[0:15] internal signals are ORed together (after masking) to form the HSEMINTRB interrupt line usually connected to the HA.
- All HSEMINTRC[0:15] internal signals are ORed together (after masking) to form the HSEMINTRC interrupt line usually connected to the HV.

HSEMINTRA

- Each HSEMINTRA[0:15] signal can be set by the master that clears the semaphore, by writing 1 to HSEM_R[0:15].INTSA in the same write cycle.
- The interrupt signal status is stored in HSEM_RISA.
- Each HSEMINTRA[0:15] interrupt can be masked using HSEM_IMSCA.
- Each HSEMINTRA[0:15] interrupt can be cleared using HSEM_ICRA.
- The status of each HSEMINTRA[0:15] signal after masking is visible in HSEM_MISA.

The same scheme applies for interrupts B, C and D (except that the HSEMINTRB/C/ D signals are not connected to the VIC).

Table 18. HSEM register list

Address	Register	Description	Page
0x000	HSEM_R0	Hardware semaphore register 0	*71*
0x004	HSEM_R1	Hardware semaphore register 1	*71*
0x008	HSEM_R2	Hardware semaphore register 2	*71*
0x00C	HSEM_R3	Hardware semaphore register 3	*71*
0x010	HSEM_R4	Hardware semaphore register 4	*71*
0x014	HSEM_R5	Hardware semaphore register 5	*71*
0x018	HSEM_R6	Hardware semaphore register 6	*71*

Table 18. HSEM register list (continued)

Address	Register	Description	Page
0x01C	HSEM_R7	Hardware semaphore register 7	71
0x020	HSEM_R8	Hardware semaphore register 8	71
0x024	HSEM_R9	Hardware semaphore register 9	71
0x028	HSEM_R10	Hardware semaphore register 10	71
0x02C	HSEM_R11	Hardware semaphore register 11	71
0x030	HSEM_R12	Hardware semaphore register 12	71
0x034	HSEM_R13	Hardware semaphore register 13	71
0x038	HSEM_R14	Hardware semaphore register 14	71
0x03C	HSEM_R15	Hardware semaphore register 15	71
0x090	HSEM_ICRALL	HSEM interrupt clear all register	72
0x0A0	HSEM_IMSCA	HSEM interrupt mask register A	72
0x0A4	HSEM_RISA	HSEM raw interrupt status register A	73
0x0A8	HSEM_MISA	HSEM masked interrupt status register A	73
0x0AC	HSEM_ICRA	HSEM interrupt clear register A	74
0x0B0	HSEM_IMSCB	HSEM interrupt mask register B	74
0x0B4	HSEM_RISB	HSEM raw interrupt status register B	75
0x0B8	HSEM_MISB	HSEM masked interrupt status register B	75
0x0BC	HSEM_ICRB	HSEM interrupt clear register B	76
0x0C0	HSEM_IMSCC	HSEM interrupt mask register C	76
0x0C4	HSEM_RISC	HSEM raw interrupt status register C	77
0x0CA8	HSEM_MISC	HSEM masked interrupt status register C	77
0x0CC	HSEM_ICRC	HSEM interrupt clear register C	78
0x0D0	HSEM_IMSCD	HSEM interrupt mask register D	78
0x0D4	HSEM_RISD	HSEM raw interrupt status register D	79
0x0D8	HSEM_MISD	HSEM masked interrupt status register D	79
0x0DC	HSEM_ICRD	HSEM interrupt clear register D	80
0xFE0	HSEMPeriphID0	HSEM peripheral identification register 0 (bits 7:0)	80
0xFE4	HSEMPeriphID1	HSEM peripheral identification register 1 (bits 15:8)	80
0xFE8	HSEMPeriphID2	HSEM peripheral identification register 2 (bits 23:16)	81
0xFEC	HSEMPeriphID3	HSEM peripheral identification register 3 (bits 31:24)	81
0xFF0	HSEMPCellID0	HSEM PCell identification register 0 (bits 7:0)	81
0xFF4	HSEMPCellID1	HSEM PCell identification register 1 (bits 15:8)	82
0xFF8	HSEMPCellID2	HSEM PCell identification register 2 (bits 23:16)	82
0xFFC	HSEMPCellID3	HSEM PCell identification register 3 (bits 31:24)	82

HSEM register descriptionsHSEM_R[0:Hardware semaphore data registers 0 to

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	INTSD	INTSC	INTSB	INTSA	HSEMDAT			
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW			

Address: HSEMBaseAddress + 0x4*x, where x = 0 to 15

Reset: 0x0000 0000

Description: Contains the free (0 value) or locked (non-zero value) state of the semaphore and the interrupt set bits.

INTSD Set HSEM interrupt signal D for semaphore x. Sets the interrupt line HSEMINTRDx for semaphore x, and subsequently HSEMINTRD global interrupt line. A read returns zero.

A write has no effect if data bus bits [3:0] != 0x0, or data bus bits [3:0] = 0x0 while the hardware semaphore is free (HSEM_Rx.HSEMDAT = 0x0),

A write with data bus bits [3:0] = 0x0 while the hardware semaphore is locked (HSEM_Rx.HSEMDAT != 0x0) (that is, when freeing the hardware semaphore):

0: no effect, write is ignored, interrupt line level is not affected

1: raw interrupt signal HSEMINTRD[x] is raised

INTSC Set HSEM interrupt signal C for semaphore x. Sets the interrupt line HSEMINTRCx for semaphore x, and subsequently HSEMINTRC global interrupt line. A read returns zero.

A write has no effect if data bus bits [3:0] != 0x0, or data bus bits [3:0] = 0x0 while the hardware semaphore is free (HSEM_Rx.HSEMDAT = 0x0),

A write with data bus bits [3:0] = 0x0 while the hardware semaphore is locked (HSEM_Rx.HSEMDAT != 0x0) (that is, when freeing the hardware semaphore):

0: no effect, write is ignored, interrupt line level is not affected

1: raw interrupt signal HSEMINTRC[x] is raised

INTSB Set HSEM interrupt signal B for semaphore x. Sets the interrupt line HSEMINTRBx for semaphore x, and subsequently HSEMINTRB global interrupt line. A read returns zero.

A write has no effect if data bus bits [3:0] != 0x0, or data bus bits [3:0] = 0x0 while the hardware semaphore is free (HSEM_Rx.HSEMDAT = 0x0),

A write with data bus bits [3:0] = 0x0 while the hardware semaphore is locked (HSEM_Rx.HSEMDAT != 0x0) (that is, when freeing the hardware semaphore):

0: no effect, write is ignored, interrupt line level is not affected

1: raw interrupt signal HSEMINTRB[x] is raised

INTSA Set HSEM interrupt signal A for semaphore x. Sets the interrupt line HSEMINTRAx for semaphore x, and subsequently HSEMINTRA global interrupt line. A read returns zero.

A write has no effect if data bus bits [3:0] != 0x0, or data bus bits [3:0] = 0x0 while the hardware semaphore is free (HSEM_Rx.HSEMDAT = 0x0),

A write with data bus bits [3:0] = 0x0 while the hardware semaphore is locked (HSEM_Rx.HSEMDAT != 0x0) (that is, when freeing the hardware semaphore):

0: no effect, write is ignored, interrupt line level is not affected

1: raw interrupt signal HSEMINTRA[x] is raised

HSEMDAT Hardware semaphore x data. Holds the current semaphore value. A write takes the new value if current value is zero, else is not modified (write is ignored). A read returns:

0: (default after reset) the semaphore is free, it is then writable to any value.

Non-zero: the semaphore is locked, it can then only be written with zero value.

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_ICRALL HSEM interrupt clear all register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSEM ICALL15	HSEM ICALL14	HSEM ICALL13	HSEM ICALL12	HSEM ICALL11	HSEM ICALL10	HSEM ICALL9	HSEM ICALL8	HSEM ICALL7	HSEM ICALL6	HSEM ICALL5	HSEM ICALL4	HSEM ICALL3	HSEM ICALL2	HSEM ICALL1	HSEM ICALL0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: HSEMBaseAddress + 0x090

Reset: 0x0000 0000

Description: Clears interrupt line x (where x = 0 to 15) for master A, B, C and D simultaneously.

HSEMICALL [0:15] Hardware semaphore x interrupt A, B, C and D clear. Clears the corresponding hardware semaphore x interrupt for all 4 masters simultaneously (this is equivalent to writing to the bits in the *HSEM_ICRA*, *HSEM_ICRB*, *HSEM_ICRC* and *HSEM_ICRD* registers).

0: no effect 1: clears the interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_IMSCA HSEM interrupt mask register A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSEM IMA15	HSEM IMA14	HSEM IMA13	HSEM IMA12	HSEM IMA11	HSEM IMA10	HSEM IMA9	HSEM IMA8	HSEM IMA7	HSEM IMA6	HSEM IMA5	HSEM IMA4	HSEM IMA3	HSEM IMA2	HSEM IMA1	HSEM IMA0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: HSEMBaseAddress + 0x0A0

Reset: 0x0000 0000

Description: Controls masking of the interrupt generated by the hardware semaphores 0 to 15 for the generation of HSEMINTRA signal. Reading this register returns the current value of the mask on the HSEM interrupt. This register is cleared upon a power-on reset only (PORnot)

HSEMIMA[0:15] Hardware semaphore x interrupt A. Enables hardware semaphore interrupt A generation for the VIC.
0: HSEMINTRAx internal interrupt signal is masked for HSEMINTRA signal generation
1: HSEMINTRAx internal interrupt signal is enabled for HSEMINTRA signal generation

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_RISA HSEM raw interrupt status register A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSEM RISA15	HSEM RISA14	HSEM RISA13	HSEM RISA12	HSEM RISA11	HSEM RISA10	HSEM RISA9	HSEM RISA8	HSEM RISA7	HSEM RISA6	HSEM RISA5	HSEM RISA4	HSEM RISA3	HSEM RISA2	HSEM RISA1	HSEM RISA0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: HSEMBaseAddress + 0x0A4

Reset: 0x0000 0000

Description: Gives the raw status of the HSEMINTRAx internal interrupt signals prior to masking. A write has no effect.

HSEM RISA[0:15] Hardware semaphore x interrupt A raw interrupt status. Gives the raw interrupt state (prior to masking) of HSEMINTRAx internal interrupt signal.

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_MISA HSEM masked interrupt status register A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSEM MISA7	HSEM MISA6	HSEM MISA5	HSEM MISA4	HSEM MISA3	HSEM MISA2	HSEM MISA1	HSEM MISA0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: HSEMBaseAddress + 0x0A8

Reset: 0x0000 0000

Description: Gives the masked status of the interrupt. A write has no effect.

HSEMMISA[0:15] Hardware semaphore x interrupt A masked interrupt status gives the corresponding masked interrupt state (after to masking) of HSEMINTRAx internal interrupt signal.

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_ICRA HSEM interrupt clear register A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSEM ICA15								HSEM ICA7	HSEM ICA6	HSEM ICA5	HSEM ICA4	HSEM ICA3	HSEM ICA2	HSEM ICA1	HSEM ICA0
RW	RW	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: HSEMBaseAddress + 0x0AC

Reset: 0x0000 0000

Description: Clears the HSEM x interrupt A.

HSEMICAx Hardware semaphore x interrupt A clear. Clears the hardware semaphore x interrupt A.
0: no effect 1: clears the corresponding interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_IMSCB HSEM interrupt mask register B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSEM IMB7	HSEM IMB6	HSEM IMB5	HSEM IMB4	HSEM IMB3	HSEM IMB2	HSEM IMB1	HSEM IMB0
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: HSEMBaseAddress + 0x0B0

Reset: 0x0000 0000

Description: Controls masking of interrupts generated by the hardware semaphores 0 to 15 for HSEMINTRB signal generation. Returns the current value of the mask on the HSEM interrupt. Cleared upon a power-on reset only (PORnot).

HSEMIMBx Hardware semaphore x interrupt B enable. Enables interrupt B generation for HSEM x.
0: HSEMINTRBx internal interrupt signal is masked for HSEMINTRB signal generation,
1: HSEMINTRBx internal interrupt signal is enabled for HSEMINTRB signal generation.

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_RISB HSEM raw interrupt status register B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSEM RISB7	HSEM RISB6	HSEM RISB5	HSEM RISB4	HSEM RISB3	HSEM RISB2	HSEM RISB1	HSEM RISB0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: HSEMBaseAddress + 0x0B4

Reset: 0x0000 0000

Description: Gives the raw status of the HSEMINTRBx internal interrupt signals prior to masking. A write has no effect.

HSEMRISB x Hardware semaphore x interrupt B raw interrupt status. Gives the raw interrupt state (prior to masking) of HSEMINTRBx internal interrupt signals.

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_MISB HSEM masked interrupt status register B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSEM MISB7	HSEM MISB6	HSEM MISB5	HSEM MISB4	HSEM MISB3	HSEM MISB2	HSEM MISB1	HSEM MISB0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: HSEMBaseAddress + 0x0B8

Reset: 0x0000 0000

Description: Gives the masked status of a corresponding interrupt. A write has no effect.

HSEMMISB x Hardware semaphore x interrupt B masked interrupt status gives the masked interrupt state (after to masking) of the corresponding HSEMINTRBx internal interrupt signal.

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_ICRB HSEM interrupt clear register B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSEM ICB7	HSEM ICB6	HSEM ICB5	HSEM ICB4	HSEM ICB3	HSEM ICB2	HSEM ICB1	HSEM ICB0
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: HSEMBaseAddress + 0x0BC

Reset: 0x0000 0000

Description: Clears the corresponding interrupt.

HSEMICBx Hardware semaphore x interrupt B clear. Clears the hardware semaphore x interrupt.
0: no effect 1: clears the interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_IMSCC HSEM interrupt mask register C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	H							HSEM IMC7	HSEM IMC6	HSEM IMC5	HSEM IMC4	HSEM IMC3	HSEM IMC2	HSEM IMC1	HSEM IMC0
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: HSEMBaseAddress + 0x0C0

Reset: 0x0000 0000

Description: Controls masking of the interrupt generated by the hardware semaphores 0 to 15 for the generation of HSEMINTRC signal. Returns the current value of the mask on the HSEM interrupt. Cleared upon a power-on reset only (PORnot)

HSEMIMCx Hardware semaphore x interrupt C enable. Enables the interrupt generation C for HSEM x.
0: HSEMINTRCx internal interrupt signal is masked for HSEMINTRC signal generation
1: HSEMINTRCx internal interrupt signal is enabled for HSEMINTRC signal generation

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_RISC HSEM raw interrupt status register C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSEM RISC7	HSEM RISC6	HSEM RISC5	HSEM RISC4	HSEM RISC3	HSEM RISC2	HSEM RISC1	HSEM RISC0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: HSEMBaseAddress + 0x0C4

Reset: 0x0000 0000

Description: Gives the HSEMINTRC internal interrupt signals raw status prior to masking. A write has no effect.

HSEMRISCx Hardware semaphore x interrupt C raw interrupt status. Gives the HSEMINTRC x internal interrupt signal raw interrupt state prior to masking.

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_MISC HSEM masked interrupt status register C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSEM MISC7	HSEM MISC6	HSEM MISC5	HSEM MISC4	HSEM MISC3	HSEM MISC2	HSEM MISC1	HSEM MISC0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: HSEMBaseAddress + 0x0C8

Reset: 0x0000 0000

Description: Gives the masked status of the corresponding interrupt. A write has no effect.

HSEMMISCx Hardware semaphore x interrupt C masked interrupt status. Gives the masked interrupt state of the HSEMINTRCx internal interrupt signal after masking.

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_ICRC HSEM interrupt clear register C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSEM ICC7	HSEM ICC6	HSEM ICC5	HSEM ICC4	HSEM ICC3	HSEM ICC2	HSEM ICC1	HSEM ICC0
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: HSEMBaseAddress + 0x0CC

Reset: 0x0000 0000

Description:

HSEMICCx Hardware semaphore x interrupt C clear. Clears the corresponding hardware semaphore interrupt.
0: no effect 1: clears the interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_IMSCD HSEM interrupt mask register D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSEM IMD7	HSEM IMD6	HSEM IMD5	HSEM IMD4	HSEM IMD3	HSEM IMD2	HSEM IMD1	HSEM IMD0
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: HSEMBaseAddress + 0x0D0

Reset: 0x0000 0000

Description: Controls masking of the interrupt generated by the hardware semaphores 0 to 15 for the generation of HSEMINTRD signal. Reading this register returns the current value of the mask on the HSEM interrupt. This register is cleared upon a power-on reset only (PORnot).

HSEMIMDx Hardware semaphore x interrupt D enable. Enables the interrupt generation D to VIC for hardware semaphore x:
0: HSEMINTRDx internal interrupt signal is masked for HSEMINTRD signal generation,
1: HSEMINTRDx internal interrupt signal is enabled for HSEMINTRD signal generation.

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_RISD HSEM raw interrupt status register D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSEM RISD7	HSEM RISD6	HSEM RISD5	HSEM RISD4	HSEM RISD3	HSEM RISD2	HSEM RISD1	HSEM RISD0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: HSEMBaseAddress + 0x0D4

Reset: 0x0000 0000

Description: Gives the raw status of the HSEMINTRDx internal interrupt signals prior to masking. A write has no effect.

HSEMRISD x Hardware semaphore x interrupt D raw interrupt status. Gives the raw interrupt state (prior to masking) of HSEMINTRDx internal interrupt signal.

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_MISD HSEM masked interrupt status register D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				H				HSEM MISD7	HSEM MISD6	HSEM MISD5	HSEM MISD4	HSEM MISD3	HSEM MISD2	HSEM MISD1	HSEM MISD0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: HSEMBaseAddress + 0x0D8

Reset: 0x0000 0000

Description: Gives the masked status of the corresponding interrupt. A write has no effect.

HSEMMISD x Hardware semaphore x interrupt D masked interrupt status gives the masked interrupt state (after masking) of HSEMINTRDx internal interrupt signal.

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEM_ICRD HSEM interrupt clear register D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSEM ICD7	HSEM ICD6	HSEM ICD5	HSEM ICD4	HSEM ICD3	HSEM ICD2	HSEM ICD1	HSEM ICD0
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: HSEMBaseAddress + 0x0DC

Reset: 0x0000 0000

Description: This register is an interrupt clear register.

HSEMICDx] Hardware semaphore x interrupt D clear. Clears the hardware semaphore x interrupt.
0: no effect 1: clears the interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

HSEMPeriphID0 HSEM peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: HSEMBaseAddress + 0xFE0

Reset: 0x0000 00AC

Description: PartNumber0 returns 0xAC.

HSEMPeriphID1 HSEM peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: HSEMBaseAddress + 0xFE4

Reset: 0x0000 000A

Description: Designer0 returns 0x00, PartNumber1 returns 0x0A.

HSEMPeriphID2 HSEM peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: HSEMBaseAddress + 0xFE8

Reset: 0x0000 0008

Description: Revision returns 0x00, Designer1 returns 0x08.

HSEMPeriphID3 HSEM peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: HSEMBaseAddress + 0xFEC

Description: 0x0000 000Configuration returns 0x0 (semaphores).

HSEMPCellID0 HSEM PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	HSEMPCellID0							
R	R	R	R	R	R	R	R	R							

Address: HSEMBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: HSEMPCellID0 returns 0x0D

HSEMPCellID1 HSEM PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	HSEMPCellID1							
R	R	R	R	R	R	R	R	R							

Address: HSEMBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: HSEMPCellID1 returns 0xF0

HSEMPCellID2 HSEM PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	HSEMPCellID2							
R	R	R	R	R	R	R	R	R							

Address: HSEMBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: HSEMPCellID2 returns 0x05

HSEMPCellID3 HSEM PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	HSEMPCellID3							
R	R	R	R	R	R	R	R	R							

Address: HSEMBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: HSEMPCellID3 returns 0xB1.

8 LCD controller (LCDC)

8.1 LCDC register addressing

Register addresses are provided as the LCD controller base address, LCDCBaseAddress, plus the register offset.

The LCDCBaseAddress is 0x1012 0000.

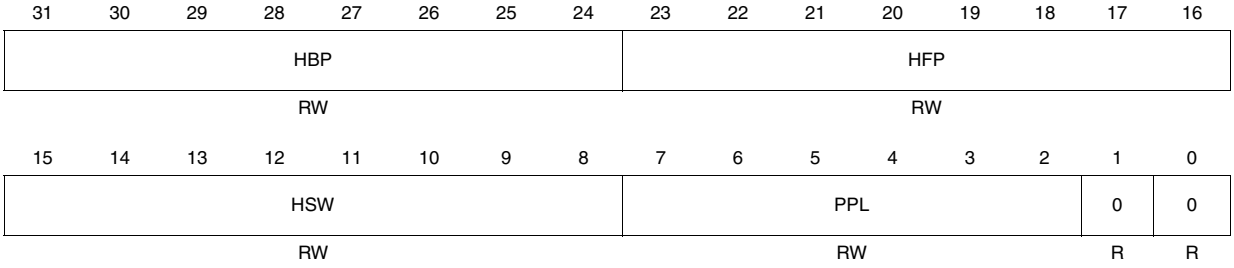
8.2 LCDC register summary

Table 19. LCDC register list

Offset	Register	Description	Page
0x000	LCDC_TIM0		84
0x004	LCDC_TIM1		85
0x008	LCDC_TIM2		85
0x00C	LCDC_TIM3		87
0x010	LCDC_UPBASE		88
0x014	LCDC_LPBASE		88
0x018	LCDC_IMSC		89
0x01C	LCDC_CR		89
0x020	LCDC_RIS		91
0x024	LCDC_MIS		92
0x028	LCDC_UPCURR		92
0x02C	LCDC_LPCURR		93
0x200 to 0x3FC	LCDC_PALx	LCDC 128 * 2 * 16-bit color palette registers 0 to 127	93
0xFE0	LCDCPeriphID0	LCDC peripheral identification register 0 (bits 7:0)	94
0xFE4	LCDCPeriphID1	LCDC peripheral identification register 1 (bits 15:8)	94
0xFE8	LCDCPeriphID2	LCDC peripheral identification register 2 (bits 23:16)	94
0xFEC	LCDCPeriphID3	LCDC peripheral identification register 3 (bits 31:24)	95
0xFF0	LCDCPCellID0	LCDC PCell identification register 0 (bits 7:0)	95
0xFF4	LCDCPCellID1	LCDC PCell identification register 1 (bits 15:8)	95
0xFF8	LCDCPCellID2	LCDC PCell identification register 2 (bits 23:16)	96
0xFFC	LCDCPCellID3	LCDC PCell identification register 3 (bits 31:24)	96

8.3 LCDC register descriptions

LCDC_TIM0 LCDC timing register 0



Address: LCDCBaseAddress + 0x000

Reset: 0x0000 0000

Description: Controls the horizontal timing parameters.

Horizontal timing restrictions: the DMA requests new data at the start of a horizontal display line. Some time must be allowed for the DMA transfer and for the data to propagate down the FIFO path in the LCDC interface. Data path latency forces restrictions on the minimum usable values (HSW: 3, HBP: 5, HFP: 5) for the horizontal porch width in STN mode.

HBP Horizontal back porch. Number of panel clock (CLPCK) periods inserted between the last line clock synchronization pulse and the start of valid data (on CLCD[23:0]) of the current line. Program with the desired value minus 1.

0x00: 1 pixel clock cycle (default after reset) through to 0xFF: 256 pixel clock cycles.

HFP Horizontal front porch. Number of panel clock (CLPCK) periods inserted between the end of valid data (on CLCD[23:0]) of the current line and the clock synchronization pulse of the next line. Program with the desired value minus 1.

0x00: 1 pixel clock cycle (default after reset) through to 0xFF: 256 pixel clock cycles.

HSW Horizontal synchronization pulse width. CLLPHS pulse width (line clock in passive mode, or horizontal synchronization pulse in active mode) expressed in CLPCK periods. Program with desired value minus 1.

0x00: 1 pixel clock cycle (default after reset) through to 0xFF: 256 pixel clock cycles.

PPL Pixels-per-line. Specifies the number of pixels in each line or row of the LCDC panel. PPL is a 16-bit value that represents between 16 and 1024 pixels per line. PPL is used to count the number of pixel clocks that occur before the HFP bit field is applied. The number of pixels per line is 16 x (PPL + 1).

0x00: 16 pixels per line (default after reset) through to 0xFF: 1024 pixels per line.

0 Reserved for future use. Reading returns 0. Must be written with 0.

LCDC_TIM1 LCDC timing register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VBP								VFP							
RW								RW							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VSW						LPP									
RW						RW									

Address: LCDCBaseAddress + 0x004

Reset: 0x0000 0000

Description: Controls the vertical timing parameters.

VBP Vertical back porch. Number of horizontal line clocks (CLLPHS)—inactive lines—at the beginning of each frame, after the vertical synchronization period (falling edge of CLFPVS signal/rising edge of SPS signal). In STN mode, program VBP with 0x00 to avoid reduced contrast.

0x00: No line clocks inserted (default after reset), up to 0xFF: 255 line clock cycles inserted.

VFP Vertical front porch. Number of line clocks inserted at the end of each frame. When a complete frame of pixels is transmitted to the LCDC display, VFP counts the number of line clock periods to wait. When the count is finished, either the vertical synchronization signal is asserted in active mode, or extra line clocks are inserted as specified by the VSW bit field in passive mode. VFP generates from 0 to 255 line clock cycles.

VSW Vertical synchronization pulse width field. Width of the vertical synchronization specified in number of line clocks. Program to the number of lines required minus 1. To avoid reduced contrast in STN LCDs, this value must be small (0 for example).

LPP Lines per panel. Number of active lines per screen. For dual panel displays this register is programmed with the number of lines on each of the upper and lower panels. Program to number of lines minus 1. LPP allows between 1 and 1024 lines.

LCDC_TIM2 LCDC timing register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCD_HI					BCD	CPL									
RW					RW	RW									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	IOE	IPC	IHS	IVS	ACB					CLKSEL	PCD_LO				
R	RW	RW	RW	RW	RW					RW	RW				

Address: LCDCBaseAddress + 0x008

Reset: 0x0000 0000

Description: Controls the LCDC timing.

PCD_HI Panel clock divisor, upper 5 bits. PCD_HI plus PCD_LO enable the LCDC panel clock frequency CLPCK to be derived from the CLCDCLK frequency. CLPCK= CLCDCLK/(PCD+2).

Mono STN modes: mono STN displays receive multiple pixels during each panel clock cycle. Those with a 4- or 8-bit interface should have CLPCK programmed to be 1/4 or 1/8 respectively of the desired individual pixel clock rate.

Color STN modes: color STN displays receive multiple pixels during each panel clock cycle. Their pixel data is stored and transferred in packed format, with each pixel represented by 3 bits (R, G and B). Consequently, one byte contains the pixel data for 2+2/3 pixels (R0G0B0, R1G1B1, R2G2), and 3 bytes contain the data of exactly 8 pixels. Each CLPCK clock cycle transfers one byte to the panel. CLPCK should be programmed as close as possible to 3/8 (the desired individual pixel rate).

TFT mode: displays the pixel clock divider; can be bypassed by setting the BCD bit.

BCD Bypass pixel clock divider. Allows the panel clock divider logic to be bypassed. Used mainly for TFT displays, as it is best to send the highest clock rate in the LCDC panel.

0: enable. Clock signal toggles only when valid data is transferred.

1: bypass. Clock is not permanently provided on the CLPCK signal in STN mode.

CPL Clock per line. Specifies the number of actual CLPCK pulses sent to the LCDC panel on each horizontal line. It must be programmed as the number of pixels per line (PPL value + 1) divided by either 1 (TFT), 4 or 8 (for mono passive), or 8/3 (for color passive), minus 1. CPL and PLL must be correctly programmed for the LCDC controller to work correctly.

CPL = PPL for TFT panels — CPL = (PPL+1)/4 - 1 for 4-bit mono STN panels

CPL = 3 x (PPL+1)/8 -1 for color STN panels — CPL = (PPL+1)/8 - 1 for 8-bit mono STN panels

IOE **TFT mode**: the invert output enable selects polarity of the output enable signal (CLACDE). The CLACDE pin is an enable that indicates to the LCDC panel when valid display data is available.

Active display mode: data is driven onto the LCDC data lines at the programmed edge of CLPCK when CLACDE is in its active state. IOE must be programmed to 0 for STN mode.

0: CLACDE active high in TFT mode — 1: CLACDE active low in TFT mode

IPC Invert panel clock. Selects the pixel clock edge on which pixel data goes to LCDC lines CLCD[23:0].

0: rising edge of CLPCK — 1: falling edge of CLPCK

IHS Invert horizontal synchronization. Inverts polarity of the CLLPHS/LP signal.

0: active high and inactive low — 1: active low and inactive high

IVS Invert vertical synchronization. Used to invert polarity of the CLFPVS/SPS signal.

0: active high and inactive low — 1: active low and inactive high

ACB AC bias pin frequency. Only applicable to STN displays, which require pixel voltage polarity to be reversed periodically to prevent damage due to DC charge accumulation. Program this field minus 1 to apply the number of line clocks between each toggle of the AC bias pin (CLACDE). It is irrelevant if the LCDC is in TFT mode as the CLACDE pin is used as a data enable signal.

CLKSEL Clock selection. Selects the clock source for the color LCDC. 0: CLCDCLK = 48 MHz.

1: CLCDCLK = 72 MHz, when MXTALI = 16 MHz, 19.2 MHz or 24 MHz.

CLCDCLK = 78 MHz, when MXTALI = 13 MHz or 26 MHz.

PCD_LO Panel clock divisor, lower five bits. See the PCD_HI bit field description for details.

0 Reserved for future use. Reading returns 0. Must be written with 0.

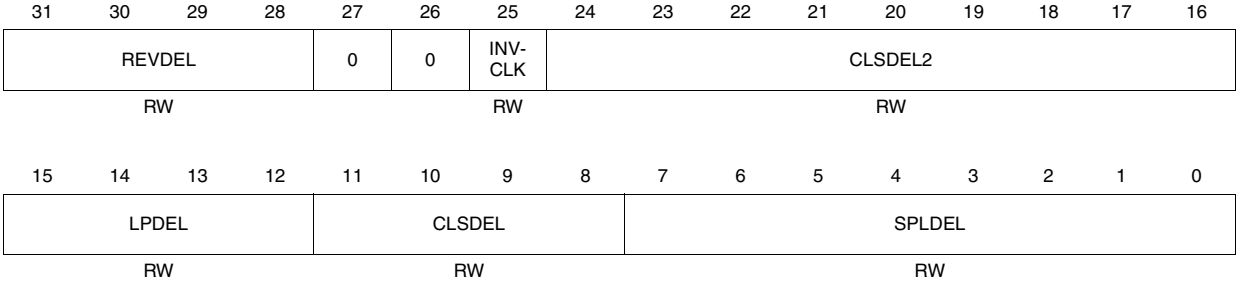
Table 20. PCD restrictions in STN modes caused by data path latency

Panel	Type	Interface	PCD minimum usable value
Single	Color	(all)	PCD = 1 (CLPCK = CLCDCLK/3)
Dual	Color	(all)	PCD = 4 (CLPCK = CLCDCLK/6)

Table 20. PCD restrictions in STN modes caused by data path latency

Single	Mono	4-bit	PCD = 2 (CLPCK = CLCDCLK/4)
Dual	Mono	4-bit	PCD = 6 (CLPCK = CLCDCLK/8)
Single	Mono	8-bit	PCD = 6 (CLPCK = CLCDCLK/8)
Dual	Mono	8-bit	PCD = 14 (CLPCK = CLCDCLK/16)

LCDC_TIM3 LCDC timing register 3



Address: LCDCBaseAddress + 0x00C

Reset: 0x0000 0000

Description: Controls timing parameters specific to HR-TFT LCDs.

REVDEL REV delay. Controls the delay (in CLPCK periods) of the polarity reversal signal from the rising edge of CLLPHS. Program with desired value minus 1.
0x0: 1 pixel clock cycle (default after reset) up to 0xF: 16 pixel clock cycles.

INVCLK Invert clock signal output to HR-TFT panel with respect to internal CPCLK signal. This inversion results in all signals being referenced by the display on the clock rising edge.
0: clock is inverted 1: clock is not inverted

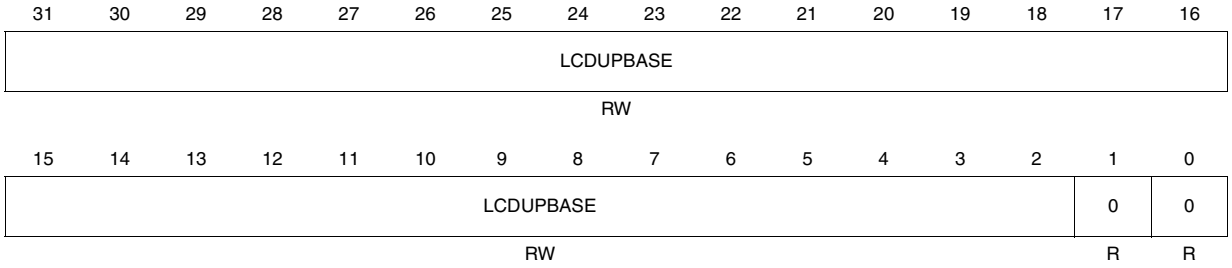
CLSDEL2 CLS delay 2. Controls the delay (in CLPCK periods) from the rising edge of the SPL/SPR signal to the falling edge of the CLS signal.
0x000: 1 pixel clock cycle (default after reset) up to 0x1FF: 512 pixel clock cycles.

LPDEL LP delay. Controls the delay (in CLPCK periods) of the line pulse LP signal from the rising edge of CLLPHS.
0x0: No delay (default after reset) up to 0xF:16 CLPCK periods inserted.

CLSDEL CLS delay. Controls the delay (in CLPCK periods) of the CLS and PS signals from the rising edge of CLLPHS.
0x0: no delay (default after reset) up to 0xF: 16 CLPCK periods inserted.

SPLDEL SPL delay. Controls the delay (in CLPCK periods) of the SPL signal during the vertical front and back porch. This must be greater than or equal to HSW + HBP. Program with the desired value minus 1.
0x00: 1 pixel clock cycle (default after reset) up to 0x7F: 128 pixel clock cycles.

LCDC_UPBASE LCDC upper panel base address register



Address: LCDCBaseAddress + 0x10

Reset: 0x0000 0000

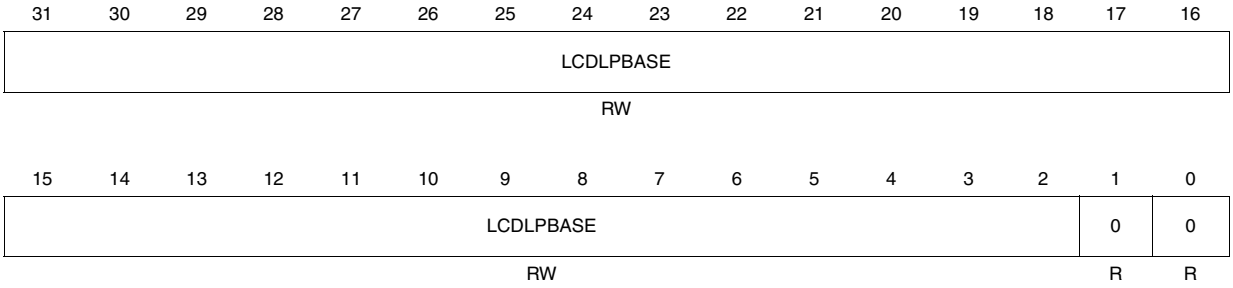
Description: This color LCDC DMA base address register programs the base addresses of the frame buffers. It is used for TFT displays, single-panel STN displays and the upper panel of dual-panel STN displays. LCDC_UPBASE (and *LCDC_LPBASE* for dual panels) must be initialized before enabling the LCDC controller. The value may be changed mid-frame to allow double-buffered video displays to be created.

This register is copied to the corresponding current register at each LCDC vertical synchronization. This generates the *LCDC_RIS*.LNBURIS bit as well as an optional interrupt. This interrupt can be used to reprogram the base address when generating double-buffered video.

LCDUPBASE LCDC upper panel base address. Word-aligned start address of the upper panel frame data in the memory.

0 Reserved for future use. Reading returns 0. Must be written with 0.

LCDC_LPBASE LCDC lower panel base address register



Address: LCDCBaseAddress + 0x014

Reset: 0x0000 0000

Description: This color LCDC DMA base address register is used for the lower panel of dual-panel STN displays. LCDC_LPBASE for dual panels must be initialized before enabling the LCDC. The value may be changed mid-frame to allow double-buffered video displays to be created. This register is copied to the corresponding current register at each LCDC vertical synchronization. This generates the *LCDC_RIS*.LNBURIS bit as well

as an optional interrupt. This interrupt can be used to reprogram the base address when generating double-buffered video.

LCDLPBASE LCDC lower panel base address. Word-aligned start address of the lower panel frame data in the memory.

0 Reserved for future use. Reading returns 0. Must be written with 0.

LCDC_IMSC LCDC interrupt mask set/clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	MBERIM	VCMPIM	LNBUIM	FUIM	0
R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	R

Address: LCDCBaseAddress + 0x018

Reset: 0x0000 0000

Description: Setting the bits in this interrupt enable register enables the corresponding raw interrupt *LCDC_RIS* bit values to be passed to the register *LCDC_MIS*.

MBERIM AHB master bus error interrupt mask.
0: masked 1: not masked

VCMPIM Vertical compare interrupt mask.
0: masked 1: not masked

LNBUIM Next base update interrupt mask.
0: masked 1: not masked

FUIM FIFO underflow interrupt mask.
0: masked 1: not masked

0 Reserved for future use. Reading returns 0. Must be written with 0.

LCDC_CR LCDC control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	CEAEN	CDWID		1XBPP		WMLVL
R	R	R	R	R	R	R	R	R	R	RW	RW		RW		RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HRTFT	FRMEN	LCDVCOMP		LCDPWR	BEPO	BEBO	BGR	LCDDUAL	STNMWD	STNTFT	STNBW	LCDBPP			LCDEN
RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW			RW

Address: LCDCBaseAddress + 0x01C

Reset: 0x0000 0000

Description: Controls the LCDC operating mode.

CEAEN Color enhancement algorithm enable for the TFT panel (from 16 bpp RGB 5:6:5 to 18 bpp RGB 6:6:6 color conversion). This bit must only be set when CDWID = 10, 1XBPP = 10, STNTFT = 1 and LCDBPP = 100, otherwise erroneous results could be generated.
The horizontal resolution of the panel must be less than or equal to 800 pixels when CEAEN = 1 (PPL < 0x32).
0: no color enhancement: only 64K colors on 18-bit TFT panel.
1: color enhancement enabled: 256K colors on 18-bit TFT panel from a 16-bpp frame buffer.

CDWID CLCD data bus width selection. Defines the number of CLCD data signals used for interfacing to the LCD panel. Only used in TFT modes (when STNTFT = 1).
00: CLCD[11:0] deliver data, CLCD[23:12] low 01: CLCD[15:0] deliver data, CLCD[23:16] low
10: CLCD[17:0] deliver data, CLCD[23:18] low 11: CLCD[23:0] deliver data

1XBPP 12, 15, or 16 bits per pixel selection. When LCDBPP = 100 (12, 15, or 16 bpp), this bit field selects the frame data representation.
00: 4:4:4: 4 upper bits unused + 4 bits per component.
01: I:5:5:5: intensity bit + 5 bits per component.
10: 5:6:5: red and blue on 5 bits, green on 6 bits.
11: reserved.

WMLVL LCD DMA FIFO watermark level. Defines the number of empty locations required (in either of the two DMA FIFOs) to cause data or frame descriptor words to be displayed.
0: four or more empty locations 1: eight or more empty locations

HRTFT HR-TFT panel selection. If STNTFT= 1, the TFT panel is selected.
0: LCD is normal TFT 1: LCD is HR-TFT or AD-TFT

FRMEN Frame modulation enable. When re-written to 1, the frame rate counter is cleared and starts counting. This allows the firmware to protect against artefacts when changing the display image (the bit does not need to be cleared first). The software must ensure consistency with the LCDBPP field.
0: frame modulation not used 1: frame modulation algorithm enabled

LCDVCOMP Generate event interrupt at:
00: start of vertical synchronization 01: start of back porch
10: start of active video 11: start of front porch

LCDPWR LCD power enable. The CLDC[23:0] signals are enabled and power-gated through to LCD panel.
0: disabled and power not gated to LCD panel 1: enabled (active)

BEPO Big-endian pixel ordering within a byte. Selects amongst little- and big-endian pixel packing for 1, 2 and 4 bpp display modes. It has no effect on 8, 16 or 24 bpp formats.
0: little-endian 1: big-endian

BEBO Big-endian byte order.
0: little-endian 1: big-endian

BGR RGB/BGR format selection.
0: RGB normal output 1: BGR (red and blue swapped)

LCDDUAL STN single/dual LCD panel selection.
0: single 1: dual

STNMWD STN monochrome interface width. Controls whether the monochrome STN LCD panel uses a 4- or 8-bit parallel interface. It has no meaning in other modes and must be programmed to zero.
0: 4-bit 1: 8-bit

STNTFT STN or TFT LCD panel selection.
0: STN, use gray scaler 1: TFT, do not use gray scaler

STNBW STN LCD panel monochrome selection (black and white). Irrelevant in TFT mode.
0: color 1: monochrome

LCDBPP LCD bits per pixel selection.
000: 1 bpp, palettized 001: 2 bpp, palettized
010: 4 bpp, palettized 011: 8 bpp, palettized
100: 12, 15 or 16 bpp (see 1XBPP bit-field), true-color (non-palettized).
101: 24 bpp (TFT panel only), non-packed data, true-color (non-palettized).
110: 24 bpp (TFT panel only), packed data, true-color (non-palettized).
111: reserved.
Note: non-packed data means a 32-bit data word contains one pixel plus one unused byte. Packed data means all four bytes of a 32-bit data word contain useful data.

LCDEN LCD controller enable of LCD signals CLLPHS, CLPCK, CLFPVS, CLACDE and CLLE.
0: disabled (held low) 1: enabled (active)

0 Reserved for future use. Reading returns 0. Must be written with 0.

LCDC_RIS LCDC raw interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	MERRIS	VCMPRIS	LNBURIS	FURIS	0
R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	R

Address: LCDCBaseAddress + 0x020

Reset: 0x0000 0000

Description: When read, this register returns five bits that can generate interrupts when set. When written, a bit value of 1 clears the interrupt corresponding to that bit. Writing 0 has no effect.

MERRIS AMBA AHB master bus error status. 1: AMBA AHB master received a bus error response from the slave.

VCMPRIS Vertical compare status. 1: one of four vertical regions, selected via register LCDControl, is reached.

LNBURIS LCD next base address update, mode-independent. Set when the current base address registers have been successfully updated by the next address registers. Signifies that a new address can be loaded if double buffering is in use.

FURIS FIFO underflow. 1: upper or lower DMA FIFOs have been read when empty, causing an underflow condition to occur.

0 Reserved for future use. Reading returns 0. Must be written with 0.

LCDC_MIS LCDC masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	MERMIS	VCMPMIS	LNBUMIS	FUMIS	0
R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	R

Address: LCDCBaseAddress + 0x024

Reset: 0x0000 0000

Description: This is a bit-by-bit logical AND of registers *LCDC_RIS* and *LCDC_IMSC*. The interrupt lines correspond to each interrupt. A logical OR of all interrupts is provided to the system interrupt controller. A write has no effect.

MERMIS AHB master error interrupt status bit. Reading returns the masked interrupt state (after masking) of the AHB master error interrupt.

VCMPMIS Vertical compare interrupt status bit. Reading returns the masked interrupt state (after masking) of the vertical compare interrupt.

LNBUMIS LCD next base update interrupt status bit. Reading returns the masked interrupt state (after masking) of the LCD next base address update interrupt. A write has no effect.

FUMIS FIFO underflow interrupt status bit. Reading returns the masked interrupt state (after masking) of the FIFO underflow interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

LCDC_UPCURR LCDC upper panel current address register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCDUPCURR															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDUPCURR															
RW															

Address: LCDCBaseAddress + 0x028

Reset: 0x0000 0000

Description: Contains the most recent address of upper panel frame data when read. It can change at any time and therefore can only be used as a mechanism for coarse delay.

LCDUPCURR LCD upper panel current address. Contains the approximate upper panel data address.

LCDC_LPCURR LCDC lower panel current address register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCDLPCURR															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCDLPCURR															
RW															

Address: LCDCBaseAddress + 0x02C

Reset: 0x0000 0000

Description: Contains the most recent address of lower panel frame data when read. It can change at any time and therefore can only be used as a mechanism for coarse delay.

LCDLPCURR LCD lower panel current address. Contains the most recent lower panel data address.

LCDC_PALx LCDC palette register x (x = 0:127)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R1					G1						B1				
RW					RW						RW				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0					G0						B0				
RW					RW						RW				

Address: LCDCBaseAddress + 0x200 + 0x4 * x, where x = 0 to 127

Reset: 0x0000 0000

Description: These registers contain 256 palette entries organized as 128 locations of two palette entries per word. Only TFT displays use all of the palette entry bits. When configured for little-endian byte ordering, bits [15:0] are the lower numbered palette entry and [31:16] are the higher numbered palette entry. Big-endian byte order configuration is the reverse, bits [31:16] being the lower numbered palette entry and [15:0] the higher numbered entry.

R[1:0] Red palette data. For color STN displays, only the four MSBs (bits Rx[4:1]) are used. For monochrome displays, this red bit field of the palette data is not used.

G[1:0] Green palette data. For color STN displays, only the four MSBs (bits Gx[5:2]) are used. For monochrome displays this green bit field of the palette data is not used.

B[1:0] Blue palette data. For color STN displays, only the four MSBs (bits Bx[4:1]) are used. For monochrome displays only this blue bit field of the palette data is used.

LCDCPeriphID0 LCDC peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: LCDCBaseAddress + 0xFE0

Reset: 0x0000 0010

Description: PartNumber0 returns 0x10.

LCDCPeriphID1 LCDC peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: LCDCBaseAddress + 0xFE4

Reset: 0x0000 0001

Description: Designer0 returns 0x0. PartNumber1 returns 0x1.

LCDCPeriphID2 LCDC peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: LCDCBaseAddress + 0xFE8

Reset: 0x0000 0018

Description: Revision returns the peripheral revision (0x1: first revision). Designer1 returns 0x8.

LCDCPeriphID3 LCDC peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: LCDCBaseAddress + 0xFEC

Reset: 0x0000 0000

Description: Configuration returns 0x00.

LCDCPCellID0 LCDC PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	LCDPCellID0							
R	R	R	R	R	R	R	R	R							

Address: LCDCBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: LCDPCellID0 returns 0x0D.

LCDCPCellID1 LCDC PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	LCDPCellID1							
R	R	R	R	R	R	R	R	R							

Address: LCDCBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: LCDPCellID1 returns 0xF0.

LCDCPCellID2 LCDC PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	LCDPCellID2							
R	R	R	R	R	R	R	R	R							

Address: LCDCBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: LCDPCellID2 returns 0x05.

LCDCPCellID3 LCDC PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	LCDPCellID3							
R	R	R	R	R	R	R	R	R							

Address: LCDCBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: LCDPCellID3 returns 0xB1.

9 Master display interface (MDIF)

9.1 MDIF register addressing

Register addresses are provided as the MDIF base address, MDIFBaseAddress, plus the register offset.

The MDIFBaseAddress is 0x1012 0000.

9.2 MDIF register summary

The device communicates to the system via a 32-bit width AMBA rev. 2.0 AHB slave bus.

Table 21. MDIF register list

Offset	Name	Description	Page
0x000	MDIF_CR	Main control register with display control signals	*99*
0x004	MDIF_SR	MDIF status register	*100*
0x008	MDIF_IMSC	Interrupt mask register	*101*
0x00C	MDIF_RIS	Raw interrupt status register	*101*
0x010	MDIF_MIS	Masked interrupt status register	*102*
0x014	MDIF_ICR	Interrupt clear register	*102*
0x018	MDIF_RDATA	Read data register	*103*
0x01C	MDIF_PBCCR	PBC duplexer, bit segmentation, dynamic control, polarity config.	*104*
0x020	MDIF_PBCBMR0	PBC bit multiplexer configuration register 0	*105*
0x024	MDIF_PBCBMR1	PBC bit multiplexer configuration register 1	*105*
0x028	MDIF_PBCBMR2	PBC bit multiplexer configuration register 2	*105*
0x02C	MDIF_PBCBMR3	PBC bit multiplexer configuration register 3	*105*
0x030	MDIF_PBCBMR4	PBC bit multiplexer configuration register 4	*105*
0x034	MDIF_PBCBCR0	PBC bit control configuration register 0	*106*
0x038	MDIF_PBCBCR1	PBC bit control configuration register 1	*106*
0x040	MDIF_BCNR	Number of cycles for defining a single bus access	*106*
0x044	MDIF_CSCDTR	Chip select valid area position in the bus access	*107*
0x048	MDIF_RDWRTR	Read or write transitions positions in the bus access	*108*
0x04C	MDIF_DOTR	DATAOUT valid area position for the bus access	*108*
0x050	MDIF_VSCR	Vertical sync capture configuration register	*109*
0x054	MDIF_HSCR	Horizontal sync capture configuration register	*110*
0x058	MDIF_SCTR	Sync capture trigger event configuration register	*110*
0x05C	MDIF_SCSR	Sync capture status register	*111*

Table 21. MDIF register list (continued)

Offset	Name	Description	Page
0x060	MDIF_NFDADR	DMA next frame descriptor address register	111
0x064	MDIF_FLADR	DMA frame line address register	112
0x068	MDIF_FEPR	DMA frame element parameter register	113
0x06C	MDIF_FLPR	DMA frame line parameter register	115
0x070	MDIF_DWPIX	Direct pixel write register	116
0x074	MDIF_DWCMD	Direct command write register	116
0xFE0	MDIFPeriphID0	Peripheral identification register 0 (bits 7:0)	117
0xFE4	MDIFPeriphID1	Peripheral identification register 1 (bits 15:8)	117
0xFE8	MDIFPeriphID2	Peripheral identification register 2 (bits 23:16)	118
0xFEC	MDIFPeriphID3	Peripheral identification register 3 (bits 31:24)	118
0xFF0	MDIFPCellID0	PCell identification register 0 bits (bits 7:0)	118
0xFF4	MDIFPCellID1	PCell identification register 1 bits (bits 15:8)	119
0xFF8	MDIFPCellID2	PCell identification register 2 bits (bits 23:16)	119
0xFFC	MDIFPCellID3	PCell identification register 3 bits (bits 31:24)	119

9.3 MDIF register descriptions

MDIF_CR Control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAWL		DMASTREQ	RESPOL	RDPOL	WRPOL	CDPOL	CS2POL	CS1POL	RESEN	CS2EN	CS1EN	INBAND	SIZE	SYCEN	DIFEN
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MDIFBaseAddress + 0x000

Reset: 0x0000 9FC0

Description:

DMAWL DMA watermark level. Defines the DMA FIFO watermark level when a bus request is raised.
00: 16 or more empty locations 01: 8 or more empty locations
10: 4 or more empty locations 11: reserved

DMASTREQ DMA stop request. Requests that the DMA stops at the end of the current frame transfer. Returns 0.
0: no effect. 1: clears the next frame address value so that the DMA stops when the current frame transfer is finished. Use *MDIF_SR*.DMASTATE to monitor the DMA state.

RESPOL DIFRES signal polarity. Defines the default (inactive) polarity for DIFRES.
0: inactive level is low 1: inactive level is high (default after reset)

RDPOL DIFRD signal polarity. Defines the default (inactive) polarity for DIFRD.
0: inactive level is low 1: inactive level is high (default after reset)

WRPOL DIFWR signal polarity. Defines the default (inactive) polarity for DIFWR.
0: inactive level is low 1: inactive level is high (default after reset)

CDPOL DIFCD signal polarity. Defines DIFCD polarity.
0: DIFCD low for data, high for command. 1: high for data, low: command (default after reset).

CS2POL DIFCS2 signal polarity. Defines the default (inactive) polarity for DIFCS2.
0: inactive level is low 1: inactive level is high (default after reset)

CS1POL DIFCS1 signal polarity. Defines the default (inactive) polarity for DIFCS1.
0: inactive level is low 1: inactive level is high (default after reset)

RESEN Display module reset. DIFRES signal polarity is defined by RESPOL.
0: DIFRES is not active 1: DIFRES is active (default after reset)

CS2EN Display module 2 selection. Defines DIFCS2 status during read/write display module accesses (active and inactive levels of the DIFCS2 signal are defined by CS2POL).
0: inactive 1: active

CS1EN Display module 1 selection. Defines the DIFCS1 status during read/write display module accesses (active and inactive levels of the DIFCS1 signal are defined by CS1POL.
0: inactive 1: active

INBAND Inband mode selection. Selects between outband (command/data [CD] signal is defined by the extra 33rd bit of the DMA FIFO) and inband mode (CD signal is defined by data bit 32).
0: outband mode 1: inband mode

SIZE Interface size selection. Specifies the parallel interface data width.
0: 8-bit parallel bus 1: 16-bit parallel bus

SYCEN Sync capture enable. Enables LCD sync capture functionality.
0: disabled 1: enabled

DIFEN MDIF enable.
0: MDIF operation is disabled, the DMA is stopped before the end of frame, and the FIFO is cleared.
1: MDIF operation is enabled. The DMA can start frame transfers.
To avoid unexpected behavior when stopping the MDIF, this bit must be cleared only after clearing register *MDIF_NFDADR* (MDIF stopped after end of frame) and *MDIF_SR*.BCBUSY = 0.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MDIF_SR Status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	PBCL BUSY	MB ERR	BC BUSY
R	R	R	R	R	R	R	R	R	R	R	R	R	R	RHW	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DFF	DNE	DMASTATE							ECOUNT						
R	R	R							R						

Address: MDIFBaseAddress + 0x004

Reset: 0x0000 0000

Description: Indicates the FIFO fill status, the DMA activity status and element counter, the bus controller busy status and the AHB master bus error state.

PBCLBUSY Pixel bit conversion logic busy. A read returns the global pixel bit conversion logic status.
0: PBC is not running 1: PBC is elaborating pixel data or LCD command

MBERR AMBA AHB master bus error status (read/clear). Returns the AMBA AHB master bus error status. Writing a 0 has no effect. If set to 1, the DMA remains in this error state until the MBER bit is cleared by writing a 1 to it and the DMA restarts its operation by fetching the frame descriptor pointed to by register *MDIF_NFDADR*.
0: no error response 1: error response received

BCBUSY Bus controller busy. Returns the bus controller status. Writing has no effect.
0: not running 1: read/write occurring on display module interface.

DFF DMA FIFO full. Returns the DMA FIFO full status. Writing has no effect.
0: not full 1: full

DNE DMA FIFO not empty. Returns the DMA FIFO empty status. Writing has no effect.
0: empty 1: not empty

DMASTATE DMA state. Returns the DMA operation state. Writing has no effect.
00: idle 01: waiting for line start event
10: waiting for frame start event 11: running

ECOUNT Element counter. Returns the remaining number of elements to be transferred by the DMA for the current line. Writing has no effect.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MDIF_IMSC MDIF interrupt mask set/clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	VS IM	HS IM	DS IM	EOF IM	MBER IM
R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

Address: MDIFBaseAddress + 0x008

Reset: 0x0000 0000

Description: Returns the current value of the mask on the relevant interrupt. A write of 1 to the particular bit sets the mask, enabling the interrupt to be read. A write of 0 clears the corresponding mask.

VSIM Vertical sync interrupt mask.
0: masked 1: not masked

HSIM Horizontal sync interrupt mask.
0: masked 1: not masked

DSIM Delayed sync interrupt mask.
0: masked 1: not masked

EOFIM End of frame interrupt mask.
0: masked 1: not masked

MBERIM Master bus error interrupt mask.
0: masked 1: not masked

0 Reserved for future use. Reading returns 0. Must be written with 0.

MDIF_RIS MDIF raw interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	VS RIS	HS RIS	DS RIS	EOF RIS	MBER RIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: MDIFBaseAddress + 0x00C

Reset: 0x0000 0000

Description: Returns the raw status of corresponding interrupt prior to masking. A write has no effect.

VSRIS Vertical sync raw interrupt status prior to masking of the vertical sync interrupt.

HSRIS Horizontal sync raw interrupt status prior to masking of the horizontal sync interrupt.

DSRIS Delayed sync raw interrupt status prior to masking of the delayed sync interrupt.

EOFRIS End of frame raw interrupt status prior to masking of the DMA end of frame interrupt.

MBERRIS Master bus error raw interrupt status prior to masking of the master bus error interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MDIF_MIS MDIF masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	VS MIS	HS MIS	DS MIS	EOF MIS	MBER MIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: MDIFBaseAddress + 0x010

Reset: 0x0000 0000

Description: Returns the current masked status of the corresponding interrupt, that is, the value of the raw interrupt status ANDed by the corresponding mask bit of the *MDIF_IMSC* register. A write has no effect.

VSMIS Vertical sync masked interrupt status after masking of the vertical sync interrupt.

HSMIS Horizontal sync masked interrupt status after masking of the horizontal sync interrupt.

DSMIS Delayed sync masked interrupt status after masking of the delayed sync interrupt.

EOFMIS End of frame masked interrupt status after masking of the DMA end of frame interrupt.

MBERMIS Master bus error masked interrupt status after masking of the master bus error interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MDIF_ICR MDIF interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	VS IC	HS IC	DS IC	EOF IC	MBER IC
R	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W

Address: MDIFBaseAddress + 0x014

Reset: 0x0000 0000

Description: Reading returns 0.

VSIC Vertical sync clear interrupt.
0: no effect 1: clears the interrupt

HSIC Horizontal sync clear interrupt.
0: no effect 1: clears the interrupt

DSIC Delayed sync clear interrupt.
0: no effect 1: clears the interrupt

EOFIC End of frame clear interrupt.
0: no effect 1: clears the interrupt

MBERIC Master bus error clear interrupt.
0: no effect 1: clears the interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

MDIF_RDATA MDIF read data register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data read from Display module, bits [15:0]															
R															

Address: MDIFBaseAddress + 0x018

Reset: 0x0000 0000

Description: Reads the content of the display module.

For a correct read operation:

- the DMA must be idle (not running)
- the FIFO must be empty
- the bus controller interface must be idle

otherwise a read returns an AHB error response on the AHB slave bus.

On a correct read:

- a read transfer is triggered on the bus controller side, using the same timing parameters for accessing the bus as for a write transfer
- Incoming data is sampled when the DIFRDnot signal is deasserted, and is then presented on the AHB slave data bus
- The AHB ready signal is deasserted until the read value is available. This mechanism is not optimized (because of the low bandwidth)

MDIF_PBCCR MDIF pixel bit conversion configuration register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	PDM		0	0	BSDM			0	BSCM		
R	R	R	R	R	RW		R	R	RW			R	RW		

Address: MDIFBaseAddress + 0x01C

Reset: 0x0000 0000

Description:

PDM MDIF pixel duplexer mode. Number of byte transfers to be performed. In normal mode, the incoming word from the DMA FIFO is put through the pixel duplexer without any change. When the incoming word is a command word, the pixel duplexer is bypassed (in both inband and outband modes). If there are fewer incoming data words than required for the pixel duplexer to form correct output words followed by a command word, then the last data words are disregarded (meaning that they are not sent in a transaction).

00: normal mode
01: 16- to 32-bit packing mode
10: 24- to 32-bit packing mode, with right shift
11: 24- to 32-bit packing mode with left shift

BSDM MDIF data bit segmentation mode. Defines the bit segmentation mode when the input word is data, that is, the number of transactions to be done on the interface for each 32-bit word delivered by the pixel duplexer.

000: one 8-bit transfer (LSB of incoming word)
001: two 8-bit transfers
010: three 8-bit transfers
011: four 8-bit transfers
100: one 16-bit transfer
101: two 16-bit transfers
110: unused
111: unused

BSCM MDIF command bit segmentation mode. Defines the bit segmentation mode when the input word is a command, that is, the number of transactions to be done on the interface for each 32-bit word delivered by the pixel duplexer.

000: one 8-bit transfer (LSB of incoming word)
001: two 8-bit transfers
010: three 8-bit transfers
011: four 8-bit transfers
100: one 16-bit transfer
101: two 16-bit transfers
110: unused
111: unused

0 Reserved for future use. Reading returns 0. Must be written with 0.

MDIF_PBCBMRx PBC bit multiplexer register x (x = 0 to 4)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MUX 31,x	MUX 30,x	MUX 29,x	MUX 28,x	MUX 27,x	MUX 26,x	MUX 25,x	MUX 24,x	MUX 23,x	MUX 22,x	MUX 21,x	MUX 20,x	MUX 19,x	MUX 18,x	MUX 17,x	MUX 16,x
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MUX 15,x	MUX 14,x	MUX 13,x	MUX 12,x	MUX 11,x	MUX 10,x	MUX 9,x	MUX 8,x	MUX 7,x	MUX 6,x	MUX 5,x	MUX 4,x	MUX 3,x	MUX 2,x	MUX 1,x	MUX 0,x
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MDIFBaseAddress + 0x020 + 0x04 * x (where x = 0 to 4)

Reset: 0x0XXX XXXX

Description: This is a bit multiplexer configuration matrix MUXi,x. It selects each BMOUT_BIT(i) from IN_BIT(j) via a complete configuration matrix made of 32 32-bit input multiplexers.

- IN_BIT (j) is multiplexer input bit j
- POSITION_i is a 5-bit binary-encoded word = {MUXi,4 MUXi,3 MUXi,2 MUXi,1 MUXi,0}
- BMOUT_BIT (i) is multiplexer output bit i = IN_BIT (POSITION_i)

Example: MUX29,0 = 0; MUX29,1 = 0; MUX29,2 = 1; MUX29,3 = 0; MUX29,4 = 1
POSITION_29 = {1 0 1 0 0} = 20 (decimal)
BMOUT_BIT(29) = IN_BIT(20)
Reset: Reset bypasses the bit multiplexer (OUT_BIT (i) = IN_BIT (i)).

Table 22. Reset values

Register	IN_BIT							Value after reset
	31	30	29		2	1	0	
MDIF_PBCMR0	1	0	1		0	1	0	0xAAAA AAAA
MDIF_PBCMR1	1	1	0		1	0	0	0xCCCC CCCC
MDIF_PBCMR2	1	1	1	through to	0	0	0	0xF0F0 F0F0
MDIF_PBCMR3	1	1	1		0	0	0	0xFF00 FF00
MDIF_PBCMR4	1	1	1		0	0	0	0xFFFF 0000

MDIF_PBCBCRx PBC bit control register x (x = 0 to 1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTL 31,x	CTL 30,x	CTL 29,x	CTL 28,x	CTL 27,x	CTL 26,x	CTL 25,x	CTL 24,x	CTL 23,x	CTL 22,x	CTL 21,x	CTL 20,x	CTL 19,x	CTL 18,x	CTL 17,x	CTL 16,x
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTL15, x	CTL 14,x	CTL 13,x	CTL 12,x	CTL 11,x	CTL 10,x	CTL 9,x	CTL 8,x	CTL 7,x	CTL 6,x	CTL 5,x	CTL 4,x	CTL 3,x	CTL 2,x	CTL 1,x	CTL 0,x
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MDIFBaseAddress + 0x034 + 0x04*x

Reset: 0x0XXX XXXX

Description: CTLi,x is the MDIF bit configuration matrix. Selects each OUT_BIT(i) value.

CTL(i),1 CTL(i),0	OUT_BIT(i)
00	BMOUT_BIT(i)
01	CD
10	0
11	1

MDIF_BCNR Bus access cycle number register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	BCN							
R	R	R	R	R	R	R	R	RW							

Address: MDIFBaseAddress + 0x040

Reset: 0x0000 0001

Description:

BCN MDIF parallel access cycle number. Defines the number of DIFCLK clock (+ 1) cycles for a single parallel transfer on the MDIF interface signals.

0x00: reserved, do not use. 0xFF: reserved, do not use.

0x01: two DIFCLK clock cycles (default after reset) through to 0xFE: 255 DIFCLK clock cycles.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MDIF_CSCDTR Chip select and CD timing register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSCDDEACT											CSCDACT				
RW											RW				

Address: MDIFBaseAddress + 0x044

Reset: 0x0000 0000

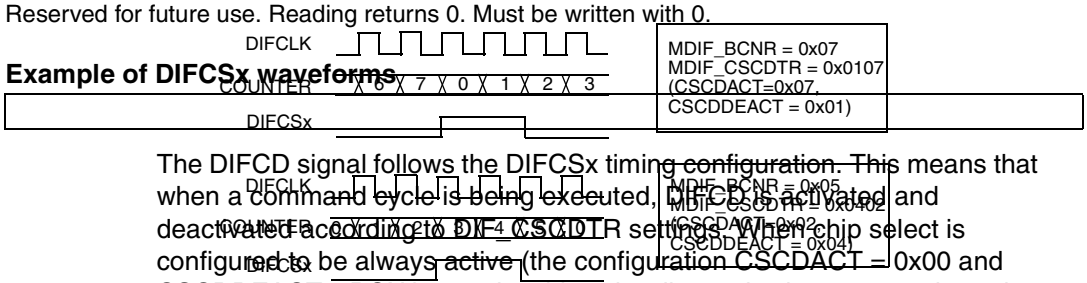
Description:

CSCDDEACT CS/CD signal deactivation. Defines the deactivation of chip select (CS1, CS2) and CD signals (last active cycle) via the parallel access counter. 0 <= CSCDACT < CSCDDEACT < 256.

CSCDACT CS/CD signal activation. Defines the activation of chip select (CS1, CS2) and CD signals (last inactive cycle) via the parallel access counter.

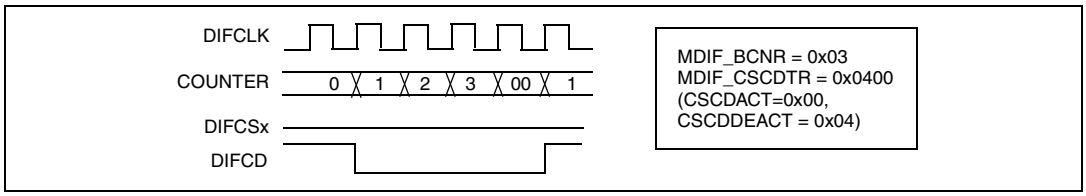
0 Reserved for future use. Reading returns 0. Must be written with 0.

Figure 5. Example of DIFCSx waveforms



The DIFCD signal follows the DIFCSx timing configuration. This means that when a command cycle is being executed, DIFCD is activated and deactivated according to DF_CSCDTR settings. When chip select is configured to be always active (the configuration CSCDACT = 0x00 and CSCDDEACT > BCN keeps the chip select line active between each read or write transfer), DIFCD is synchronized when the counter reaches the CSCDACT value. *Figure 6* gives an example.

Figure 6. DIFCD waveform when DIFCSx is always active



MDIF_RDWRTR Read/write timing register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MOT INT
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			RWDEACT								RWACT				
			RW								RW				

Address: MDIFBaseAddress + 0x048

Reset: 0x0000 0000

Description:

MOTINT Read/write Motorola/Intel mode selection.

0: Intel 8080-series-like protocol: pin DIFRDnot generates a strobe for read accesses, while pin DIFWRnot generates a strobe for a write access.

1: Motorola 68000-series-like protocol. The DIFWRnot pin becomes a read (high)/write (low) indication pin, while the DIFRDnot pin generates a strobe for both read and write access types.

RWDEACT Read/write signal deactivation. Defines the deactivation of read/write signals (last active cycle) via the parallel access counter. 0 <= RWACT < RWDEACTIVE <= BCN.

RWACT Read/write signal activation. Defines the activation of read/write signals (last inactive cycle) via the parallel access counter.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MDIF_DOTR Data output timing register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			DODEACT								DOACT				
			RW								RW				

Address: MDIFBaseAddress + 0x04C

Reset: 0x0000 0000

Description: The configuration DODEACT > DOACT keeps the data bus driven by the MDIF during the complete write frame

DODEACT DataOut signal deactivation. Defines the deactivation of the DataOut signal (last active cycle) via the parallel access counter. 0 <= DOACT < DODEACT < 256

DOACT DataOut signal activation. Defines the activation of the DataOut signal (last inactive cycle) via the parallel access counter

0 Reserved for future use. Reading returns 0. Must be written with 0.

The *Figure 7.* and *Figure 8.* are examples of MDIF bus output timing waveforms:

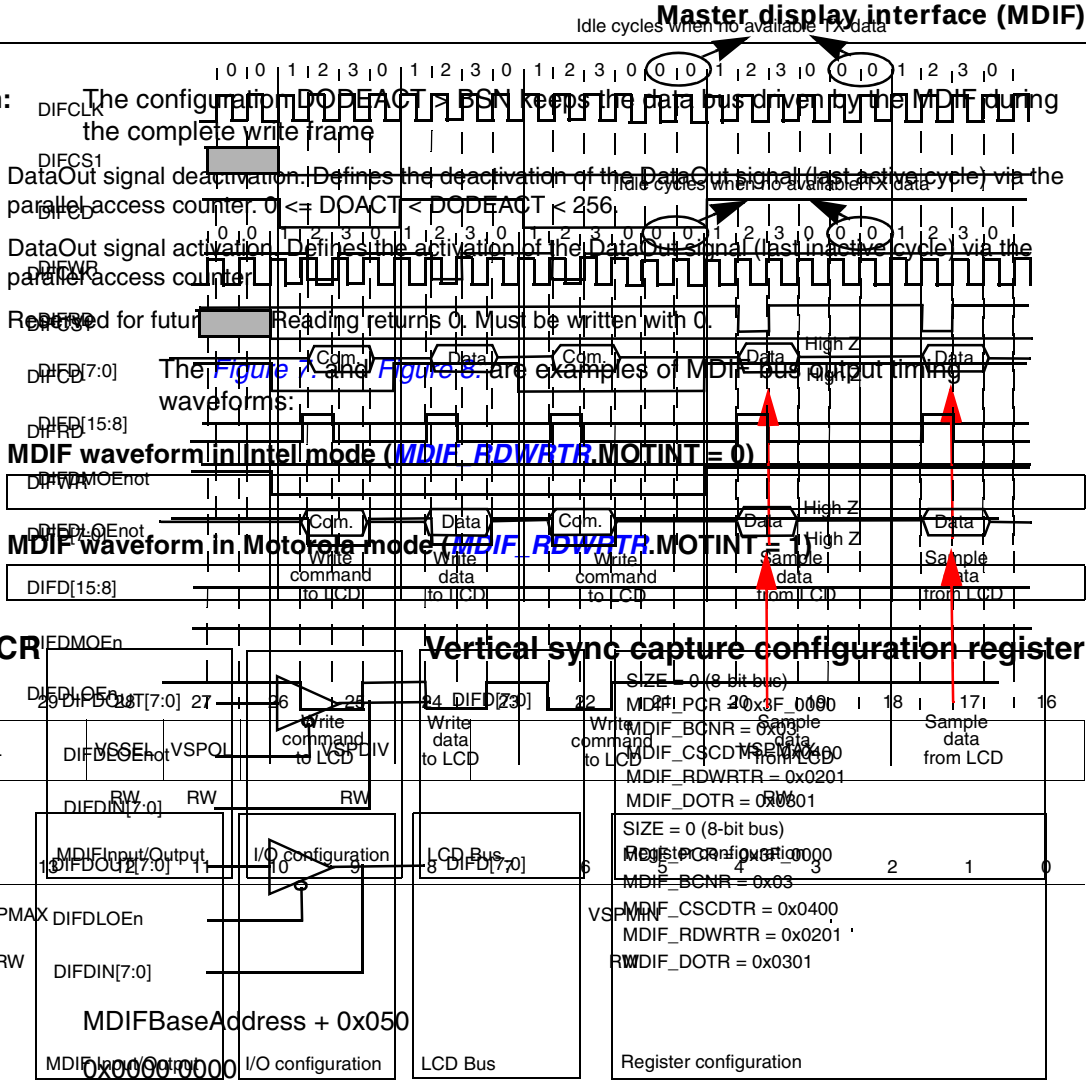


Figure 7. MDIF waveform in Intel mode (*MDIF_RDWRTR*.MOTINT = 0)

Figure 8. MDIF waveform in Motorola mode (*MDIF_RDWRTR*.MOTINT = 1)

MDIF_VSCR Vertical sync capture configuration register

Address: MDIFBaseAddress + 0x050

Reset: 0x0000 0000

Description:

VSDBL Vertical sync debounce length. Number of DIFCLK cycles during which DIFVSYNC must remain stable to be acknowledged and not disregarded as a glitch.

VSSEL Vertical sync input selection. Chooses the DIFVSYNC or DIFHSYNC input.

0: DIFVSYNC (default after reset) 1: DIFHSYNC

VSPOL Vertical sync polarity. Selects if DIFVSYNC input has active high or low sync pulses.

0: high (default after reset) 1: low

VSPDIV Vertical sync clock division factor. The MDIF clock (DIFCLK) is divided by the VSPDIV factor before clocking the pulse width comparator. The pulse width comparator clock is DIFCLK divided by:

000: 1 001: 2

010: 4 011: 8

100: 16 101: 32

110: 64 111: 128

VSPMAX Vertical sync maximum pulse duration. Maximum width of a vertical sync pulse, expressed in number of (DIFCLK / VSPDIV) clock periods.

VSPMIN Vertical sync minimum pulse duration. Minimum width of a vertical sync pulse, expressed in number of (DIFCLK / VSPDIV) clock periods.

MDIF_HSCR Horizontal sync capture configuration register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HSDBL			HSSEL	HSPOL	HSPDIV			HSPMAX							
RW			RW	RW	RW			RW							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSPMAX								HSPMIN							
RW								RW							

Address: MDIFBaseAddress + 0x054

Reset: 0x0000 0000

Description:

HSDBL Horizontal sync debounce length. Number of DIFCLK cycles during which DIFHSYNC must remain stable to be acknowledged and not disregarded as a glitch.

HSSEL Horizontal sync input selection.
0: DIFHSYNC is selected 1: DIFVSYNC is selected

HSPOL Horizontal sync polarity selection.
0: DIFHSYNC input has active high sync pulses 1: DIFHSYNC input has active low sync pulses

HSPDIV Horizontal sync clock division factor. The MDIF clock (DIFCLK) is divided by the HSPDIV factor before clocking the pulse width comparator. The pulse width comparator clock is DIFCLK divided by:
000: 1 001: 2
010: 4 011: 8
100: 16 101: 32
110: 64 111: 128

HSPMAX Horizontal sync maximum pulse duration. Maximum width of a horizontal sync pulse, expressed in number of (DIFCLK / HSPDIV) clock periods.

HSPMIN Horizontal sync minimum pulse duration. Minimum width of a horizontal sync pulse, expressed in number of (DIFCLK / HSPDIV) clock periods.

MDIF_SCTR Sync capture trigger event configuration register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	SYNCSEL							TRDELAY						
R	R	RW							RW						

Address: MDIFBaseAddress + 0x058

Reset: 0x0000 0000

Description:

SYNCSEL LCD sync output selection. Selects the event sent to the SVA to synchronize with the external display device (signal named DIFLCDSYNC).

00: DIFLCDSYNC is forced low
01: extracted horizontal sync event
10: extracted vertical sync event
11: trigger event

TRDELAY Delay after vertical sync to trigger event. Number of extracted horizontal sync pulses to wait after each vertical sync detected in order to generate the trigger event.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MDIF_SCSR Sync capture status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	VSTA	HSTA
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: MDIFBaseAddress + 0x05C

Reset: 0x0000 0000

Description:

VSTA Vertical sync status gives the debounced DIFVS input level (before logical inversion).

HSTA Horizontal sync status gives the debounced DIFHS input level (before logical inversion).

0 Reserved for future use. Reading returns 0. Must be written with 0.

MDIF_NFDADR MDIF DMA next frame descriptor address register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NFDAD[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NFDAD[15:4]												0	0	0	EOF ITEN
RW												R	R	R	RW

Address: MDIFBaseAddress + 0x060

Reset: 0x0000 0000

Description: Contains the memory address pointer of the next frame descriptor for the DMA channel and the end-of-frame interrupt enable bit. When the register is loaded with a non-zero value, and the MDIF is enabled (DIFEN = 1 in *MDIF_CR*), the DMA fetches the new 4-word descriptor starting at this address after finishing the current frame transfer, if any. The DMA writes this register with the word[0] of the new frame descriptor, then writes word[1:3] into the 3 registers described in the following sub-

sections, and then starts the frame element transfer. The address is always 16-byte aligned, so bits [0:3] are not implemented.

NFDAD Next frame descriptor address. Contains the source address of the MDIF DMA next frame descriptor. When written with a non-zero value, the DMA fetches the four words of the descriptor after finishing the current frame transfer. The address must be aligned to a 16-byte boundary (bits [3:0] are zeroed).

EOFITEN DMA end of frame interrupt enable. Enables generation of an end-of-frame interrupt (EOFINTR) when the current frame is completed (when the last word of the current frame has been fetched), before fetching the new frame descriptor at the address defined by the NFDAD bit field.

0: EOFINTR interrupt not set 1: EOFINTR interrupt set

0 Reserved for future use. Reading returns 0. Must be written with 0.

A frame descriptor consists of four words aligned on a 16-byte boundary in memory:
– word[0] contains the value loaded by the DMA into the *MDIF_NFDADR* register,
– word[1] contains the value loaded by the DMA into the *MDIF_FLADR* register,
– word[2] contains the value loaded by the DMA into the *MDIF_FEPR* register,
– word[3] contains the value loaded by the DMA into the *MDIF_FLPR* register.
Multiple frame descriptors can be chained together in a list, making it possible for the MDIF DMA to transfer data/commands to the display module from several discontinuous blocks of memory, so as to refresh part or all screen content of the display module.

MDIF_FLADR DMA frame line address register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLAD[31:16]															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLAD[15:0]															
R															

Address: MDIFBaseAddress + 0x064

Reset: 0x0000 0000

Description: Contains the memory address of the first element of the DMA channel frame line. It is loaded by the DMA when fetching the frame descriptor word[1]. When the DMA

fetches the last data of a frame line, the register is incremented with the LINC value before the DMA starts fetching the first element of the next line.

The fetch address delivered by the DMA on the AHB master port is equal to FLAD[31:0] with bit [1] or bit [0] masked when the element size *MDIF_FEPR*.ESIZE = 10 (word) or 01 (half-word) respectively. This guarantees the correct address alignment on the AHB bus.

FLAD Frame line address. Contains the frame line start address. It is incremented by LINC when the DMA fetches the last word of a frame line (FLAD += LINC). It is a read-only register, loaded by the DMA itself with word[1] of the frame descriptor.

MDIF_FEPR DMA frame element parameter register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EINC															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LWAITEV		ESIZE							ENB						
R		R							R						

Address: MDIFBaseAddress + 0x068

Reset: 0x0000 0000

Description: Defines the size of an element, the space between consecutive elements in the memory and the number of elements per frame line. It is loaded by the DMA when fetching the frame descriptor word[2].

When the DMA fetches a new frame line, the element counter of the DMA (MDIF_DMASR.ECOUNT) is loaded with the ENB value, and the internal element address register (MDIF_DMASR.EADR) is loaded with the *MDIF_FLADR* register. After each element fetch from the memory address pointed at by EADR, ECOUNT is decremented by 1, while EADR is incremented with the EINC value.

EINC Address increment between elements in a line. Defines the address increment between two elements in a frame line. The internal DMA element address register is incremented by EINC after the DMA has fetched an element in a line. EINC is a signed 16-bit value.

0x0000: the element address is not incremented. This could be used to send the same constant value to a screen area of the display module (default after reset).

0x0001: the element address is incremented by 1 (EADR += 1) each time a data element of the frame line is fetched by the DMA, and so on until

0x7FFF: the element address is incremented by 32767 (EADR += 32767) each time a data element of the frame line is fetched by the DMA.

0x8000: the element address is decremented by 32768 (EADR -= 32768) each time a data element of the frame line is fetched by the DMA, and so on until

0xFFFF: the element address is decremented by 1 (EADR -= 1) each time a data element of the frame line is fetched by the DMA.

LWAITEV DMA line start wait event. Defines if the start of the DMA line fetch must wait for one of the events computed by the sync capture block.

00: do not wait for any sync event before transferring data of the current line. DMA data transfers for the line start immediately after the frame descriptor has been fetched.

01: wait for horizontal event before starting the DMA data transfer for the current line.

10: wait for vertical event before starting the DMA data transfer for the current line.

11: wait for trigger event before starting the DMA data transfer for the current line.

ESIZE Element size. Defines the size of an element.

00: 1 byte 01: 2 bytes (half-word)

10: 4 bytes (word) (default) 11: reserved, do not use this value

ENB Number of elements per line. The DMA element counter, ECOUNT, is loaded with this value at the start of each frame line, and decremented after each element fetch. When ECOUNT= 0 and if the line counter (LCOUNT) is not 0, the DMA starts a new line, otherwise the frame is completed, and the DMA can restart fetching the new frame descriptor.

0x000: 1 (default value after reset), until 0xFFF: 4096 elements per line.

The total number of bytes transferred by the DMA for a line is (ENB+1) x 2^{ESIZE}.

Note: The HSIZE[1:0] signal directly reflects the ESIZE value on the AHB master port when the DMA is fetching data elements (HSIZE[1:0] is always 10 (words) when fetching a frame descriptor).

For optimum system bus utilization, the DMA automatically groups element fetches in AHB bursts, according to the following rules:

- When EINC = 0x001 and ESIZE = 00 (bytes) and ECOUNT >= 4 (that is, there are more than 4 remaining elements in the line) and the fetch address does not cross a 1kbyte boundary, then the DMA performs an incremental burst access of 4 bytes
(HBURST = INC4, HSIZE = byte),
- When EINC = 0x002 and ESIZE = 01 (half-words) and ECOUNT >= 4 (that is, there are more than 4 remaining elements in the line) and the fetch address does not cross a 1kbyte boundary, then the DMA performs an incremental burst access of 4 half-words.
(HBURST = INC4, HSIZE = half-word),
- When EINC = 0x004 and ESIZE = 10 (words) and ECOUNT >= 4 (that is, there are more than 4 remaining elements in the line) and the fetch address does not cross a 1kbyte boundary, then the DMA performs an incremental burst access of 4 words.
(HBURST = INC4, HSIZE = word).

Note: When the frame buffer is located in the external SDRAM memory, we recommend setting EINC = 0x004, and ESIZE = 10 (word) so as to minimize the bus bandwidth needed for refreshing the display module content; otherwise higher page switching on the SDRAM and increased AHB bus occupation may affect the overall performance of the system.

MDIF_FLPR DMA frame line parameter register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LINC															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FWAITEV		0	CDOUT	LCOUNT											
R		R	R	R											

Address: MDIFBaseAddress + 0x06C

Reset: 0x0000 0000

Description: Contains the number of lines per frame and address increments between two consecutive display lines. It is loaded by the DMA itself when fetching the frame descriptor word[3].

When the DMA has fetched the 4th word of a frame descriptor, it waits for the selected event before transferring the data frame. When the last word of a line has been fetched, the line counter (LCOUNT) value is decremented, the frame line address (*MDIF_FLADR*.FLAD) is incremented with the LINC value and a new line is fetched if non-zero, otherwise the DMA either loads a new frame descriptor or stops (if NFDAD=0).

LINC Address increment between first elements of two consecutive lines in a frame (a signed 16-bit value). The frame line address (MDIF FLADR.FLAD) is incremented with the LINC value after the last element of a line has been fetched (FLAD += LINC).

0x0000: FLAD is not incremented. This could be used to send the same line to a screen area of the display module (default value after reset).

0x0001: FLAD is incremented by 1 (FLAD += 1) after each frame line transfer, until

0x7FFF: FLAD is incremented by 32767 (FLAD += 32767) after each frame line transfer.

0x8000: FLAD is decremented by 32768 (FLAD -= 32768) each time a data element of the frame line is fetched by the DMA, until

0xFFFF: FLAD is decremented by 1 (FLAD -= 1) each time a data element of the frame line is fetched by the DMA.

FWAITEV DMA frame start wait event. Defines if the DMA frame fetch should wait for an event (computed by the sync capture block) before starting the DMA for the current data frame transfer.

00: do not wait for any sync event
01: wait for horizontal event
10: wait for vertical event
11: wait for trigger event

CDVAL Command/data value. In outband mode, this bit provides the CD value for the complete frame.

0: the 33rd bit of the DMA FIFO is written with 0 for each data frame pushed in the FIFO; thus, when outband operation is selected, the DIFCD signal is low when transferring corresponding data to the display module.

1: The 33rd bit of the DMA FIFO is written with 1 for each data frame pushed in the FIFO, thus, when outband operation is selected, the DIFCD signal is high when transferring corresponding data to the display module.

LCOUNT Lines per frame counter. Initially, when fetching a frame descriptor, this value is loaded with the total number of lines to be transferred per frame. When the DMA is running, this field returns the remaining number of lines still to be transferred by the DMA for the current frame.

0x000: 1 (default value after reset), until 0xFFF: 4096 lines per frame.

The total number of bytes transferred by the DMA for a frame is: (LCOUNT+1)x (ENB+1) x 2^{ESIZE}.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MDIF_DWPIX MDIF direct write pixel register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Direct Pixel value, bits [31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct Pixel value, bits [15:0]															
w															

Address: MDIFBaseAddress + 0x070

Reset: 0x0000 0000

Description: Allows the host CPU to send pixel data to the display module without programming the DMA (when the DMA is idle and MDIF enabled, *MDIF_CR*.DIFEN = 1). A write to this register puts the 32-bit value on top of the DMA FIFO with the extra bit cleared (FIFO bit 33 = 0). In outband mode, the 32-bit value is considered as pixel data. In inband mode, bit 32 of the written value determines if this is a pixel value (bit 32 = 0) or command (bit 32 = 1) for the display module. Reads from this register return zero.

*Note: There is no dynamic arbitration between the DMA and this register to access the FIFO. A write to the FIFO with this direct write register is **only** possible when the DMA is idle. A write to MDIF_DWPIX while the DMA is running or waiting a start event (when MDIF_SR.DMASTATE does not = 00) produces an AHB error response on the AHB slave interface.*
When the FIFO is full, the AHB write access is delayed by insertion of a wait state.
When the MDIF is disabled (MDIF_CR.DIFEN = 0), any write access to MDIF_DWCMD produces an error response on the AHB slave interface.

MDIF_DWCMD MDIF direct write command register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Direct Command value, bits [31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Direct Command value, bits [15:0]															
w															

Address: MDIFBaseAddress + 0x074

Reset: 0x0000 0000

Description: Allows the host CPU to send command data to the display module without programming the DMA (when the DMA is idle and MDIF enabled, *MDIF_CR*.DIFEN = 1). A write to this register puts the 32-bit value on top of the DMA FIFO with the extra bit set (FIFO bit 33 =1). In outband mode, the 32-bit value is considered as command data. In inband mode, bit 32 of the written value determines if this is a pixel value (bit

32 = 0) or command (bit 32 = 1) for the display module. Reads from this register return zero.

*Note: There is no dynamic arbitration between the DMA and this register to access the FIFO. A write to the FIFO with this direct write register is **only** possible when the DMA is idle. A write to MDIF_DWCMD while the DMA is running or waiting for a start event (MDIF_SR.DMASTATE does not = 00) produces an error response on the AHB slave interface.*
When the FIFO is full, the AHB write access is delayed by insertion of a wait state.
When the MDIF is disabled (MDIF_CR.DIFEN = 0), any write access to MDIF_DWCMD produces an error response on the AHB slave interface.

MDIFPeriphID0 MDIF peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: MDIFBaseAddress + 0xFE0

Reset: 0x0000 0001

Description: Returns 0x01.

MDIFPeriphID1 MDIF peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: MDIFBaseAddress + 0xFE4

Reset: 0x0000 0001

Description: Designer0 returns 0x00.
PartNumber1 returns 0x01.

MDIFPeriphID2 MDIF peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: MDIFBaseAddress + 0xFE8

Reset: 0x0000 0008

Description: Revision returns the peripheral revision.
Designer1 returns 0x08.

MDIFPeriphID3 MDIF peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: MDIFBaseAddress + 0xFEC

Reset: 0x0000 0010

Description: Returns 0x10 (16-bit MDIF interface).

MDIFPCellID0 MDIF PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MDIFPCellID0							
R	R	R	R	R	R	R	R	R							

Address: MDIFBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: Returns 0x0D.

MDIFPCellID1 MDIF PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MDIFPCellID1							
R	R	R	R	R	R	R	R	R							

Address: MDIFBaseAddress + 0xFF4

Reset: 0x00000F0

Description: Returns 0xF0.

MDIFPCellID2 MDIF PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MDIFPCellID2							
R	R	R	R	R	R	R	R	R							

Address: MDIFBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: Returns 0x05.

MDIFPCellID3 MDIF PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MDIFPCellID3							
R	R	R	R	R	R	R	R	R							

Address: MDIFBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: Returns 0xB1.

10 Multi-timer units (MTU0,1)

10.1 MTU register addressing

Register addresses are provided as the MTU base address, MTUnBaseAddress, plus the register offset.

The two MTU base addresses are different.

Table 23. MTU base addresses

MTU base address	MTU
0x101E 2000	MTU0
0x101E 3000	MTU1

10.2 MTU register summary

The device communicates to the system via 32-bit-wide control registers accessible via AMBA rev. 2.0 peripheral bus (APB).

Table 24. MTU register list

Offset	Register name	Description	Page
0x000	MTUn_IMSC	MTU interrupt mask set/clear register	121
0x004	MTUn_RIS	MTU raw interrupt status register	122
0x008	MTUn_MIS	MTU masked interrupt status register	122
0x00C	MTUn_ICR	MTU interrupt clear register	123
0x010	MTUn_T0LR	MTU timer 0 load value register	123
0x014	MTUn_T0VAL	MTU timer 0 value register	124
0x018	MTUn_T0CR	MTU timer 0 control register	124
0x01C	MTUn_T0BGLR	MTU timer 0 background load value register	125
0x020	MTUn_T1LR	MTU timer 1 load value register	123
0x024	MTUn_T1VAL	MTU timer 1 value register	124
0x028	MTUn_T1CR	MTU timer 1 control register	124
0x02C	MTUn_T1BGLR	MTU timer 1 background load value register	125
0x030	MTUn_T2LR	MTU timer 2 load value register	123
0x034	MTUn_T2VAL	MTU timer 2 value register	124
0x038	MTUn_T2CR	MTU timer 2 control register	124
0x03C	MTUn_T2BGLR	MTU timer 2 background load value register	125
0x040	MTUn_T3LR	MTU timer 3 load value register	123
0x044	MTUn_T3VAL	MTU timer 3 value register	124

Table 24. MTU register list (continued)

Offset	Register name	Description	Page
0x048	MTUn_T3CR	MTU timer 3 control register	124
0x04C	MTUn_T3BGLR	MTU timer 3 background load value register	125
0xFE0	MTUn_PeriphID0	MTU peripheral identification register 0 (bits 7:0)	125
0xFE4	MTUn_PeriphID1	MTU peripheral identification register 1 (bits 15:8)	125
0xFE8	MTUn_PeriphID2	MTU peripheral identification register 2 (bits 23:16)	125
0xFEC	MTUn_PeriphID3	MTU peripheral identification register 3 (bits 31:24)	125
0xFF0	MTUn_PCellID0	MTU Pcell identification register 0 (bits 7:0)	126
0xFF4	MTUn_PCellID1	MTU Pcell identification register 1 (bits 15:8)	126
0xFF8	MTUn_PCellID2	MTU Pcell identification register 2 (bits 23:16)	126
0xFFC	MTUn_PCellID3	MTU Pcell identification register 3 (bits 31:24)	126

10.3 MTU register descriptions

MTUn_IMSC MTU interrupt mask set/clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	T3IM	T2IM	T1IM	T0IM
R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Address: MTUnBaseAddress + 0x000

Reset: 0x0000 0000

Description: Returns the current value of the mask on the relevant timer interrupt. All bits are cleared to 0 upon reset.

T[0:3]IM Timer x interrupt mask. Reading gives the current mask for the timer interrupt.
0: clears the mask 1: sets the mask

0 Reserved for future use. Reading returns 0. Must be written with 0.

MTUn_RIS MTU raw interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	T3RIS	T2RIS	T1RIS	T0RIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: MTUnBaseAddress + 0x004

Reset: 0x0000 0000

Description: Returns the raw interrupt status.

T[0:3]RIS Timer x raw interrupt status. indicates the interrupt status from the counter.
0: timer interrupt is not set 1: timer interrupt is set

0 Reserved for future use. Reading returns 0. Must be written with 0.

MTUn_MIS MTU masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	T3MIS	T2MIS	T1MIS	T0MIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: MTUnBaseAddress + 0x008

Reset: 0x0000 0000

Description: Indicates the current masked status value of the corresponding timer interrupt. A write has no effect.

T[0:3]MIS Timer x masked interrupt status. Gives the masked value of the timer interrupt status.
0: timer line interrupt not active 1: timer line asserting interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

MTUn_ICR MTU interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	T3IC	T2IC	T1IC	T0IC
R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Address: MTUnBaseAddress + 0x00C

Reset: 0x0000 0000

Description: Clears the interrupts.

T[0:3]IC Timer x interrupt clear. Reading always returns 0.
0: no effect 1: clears corresponding timer interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

MTUn_TxLR MTU timer x load register (x = 0:3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TxLOAD[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxLOAD[15:0]															
RW															

Address: MTUnBaseAddress + 0x010 + x * 0x10 (x = 0 to 3)

Reset: 0x0000 0000

Description: This 32-bit register contains the value from which the timer x (x = 0 to 3) counter is to decrement. When this register is written to, the count is immediately restarted from the new value. The value in this register is overwritten automatically after the count reaches zero, if a write was done to the background load register since the previous time that zero was reached.

T[0:3]LOAD Timer x load value.

MTUn_TxVAL MTU timer x value register (x = 0:3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TxVAL[31:16]															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxVAL[15:0]															
R															

Address: MTUnBaseAddress + 0x014 + x * 0x10 (x = 0 to 3)

Reset: 0xFFFF FFFF

Description: Holds the current value of the decrementing counter x.

T[0:3]VAL Timer x value. Returns the current value of the decrementing counter. A write has no effect.

MTUn_TxCR MTU timer x control register (x = 0:3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	TxEN	TxMOD	0	0	TxPRE		TxSZ	TxOS
R	R	R	R	R	R	R	R	RW	RW	R	R	RW	RW	RW	RW

Address: MTUnBaseAddress + 0x018 + x * 0x10 (x = 0 to 3)

Reset: 0x0000 0000

Description: Configures timer x.

T[0:3]EN Timer x enable.
0: timer is disabled (default) 1: timer is enabled

T[0:3]MOD Timer x mode;
0: timer is in free-running mode (default) 1: timer is in periodic mode

T[0:3]PRE Timer x prescaler.
00: clock is divided by 1 (default) 01: clock is divided by 16
10: clock is divided by 256 11: undefined, do not use

T[0:3]SZ Timer x size.
0: timer is a 16-bit counter (default) 1: timer is a 32-bit counter

T[0:3]OS Timer x one-shot count. Selects one-shot or wrapping counter mode.
0: timer is in wrapping mode (default) 1: timer is in one-shot mode

0 Reserved for future use. Reading returns 0. Must be written with 0.

MTUn_TxBGLR MTU timer x background load register (x = 0:3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TxBGLOAD[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxBGLOAD[15:0]															
RW															

Address: MTUnBaseAddress + 0x01C + x * 0x10 (x = 0 to 3)

Reset: 0x0000 0000

Description: This 32-bit register contains the value from which the counter is to decrement, after completing the current countdown. When this register is written to, the current count is not interrupted but a flag is set. This flag indicates that when the count reaches zero, the background load value is to be copied to the load register, and the next countdown must continue from that value.

T[0:3]BGLOAD Timer x background load value.

MTUn_PeriphID0 MTU peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: MTUnBaseAddress + 0xFE0

Reset: 0x0000 0004

Description: PartNumber0 returns 0x04.

MTUn_PeriphID1 MTU peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: MTUnBaseAddress + 0xFE4

Reset: 0x0000 0008

Description: Designer0 returns 0x0. PartNumber1 returns 0x8.

MTUn_PeriphID2 MTU peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: MTUnBaseAddress + 0xFE8

Reset: 0x0000 0008

Description: Revision returns 0x0. Designer1 returns 0x8.

MTUn_PeriphID3 MTU peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: MTUnBaseAddress + 0xFEC

Reset: 0x0000 0004

Description: Configuration returns 0x04 (4 timers).

MTUn_PCellID0 MTU Pcell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MTUPCellID0							
R	R	R	R	R	R	R	R	R							

Address: MTUnBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: MTUPCellID0 returns 0x0D.

MTUn_PCellID1 MTU Pcell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MTUPCellID1							
R	R	R	R	R	R	R	R	R							

Address: MTUnBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: MTUPCellID1 returns 0xF0.

MTUn_PCellID2 MTU Pcell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MTUPCellID2							
R	R	R	R	R	R	R	R	R							

Address: MTUnBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: MTUPCellID2 returns 0x05.

MTUn_PCellID3 MTU Pcell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MTUPCellID3							
R	R	R	R	R	R	R	R	R							

Address: MTUnBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: MTUPCellID3 returns 0xB1.

11 Watchdog timer (WDT)

11.1 WDT register addressing

Register addresses are provided as the WDT base address, WDTBaseAddress, plus the register offset.

The WDTBaseAddress is 0x101E 1000.

11.2 WDT register summary

The device communicates to the system via 32-bit-wide control registers accessible via the AMBA rev. 2.0 peripheral bus (APB).

Table 25. WDT register list

Offset	Name	Description	Page
0x000	WDT_LR	Watchdog load value register	*129*
0x004	WDT_VAL	Watchdog current value (read-only) register	*129*
0x008	WDT_CR	Watchdog control register	*130*
0x00C	WDT_ICR	Watchdog interrupt clear register	*130*
0x010	WDT_RIS	Watchdog raw interrupt status register	*131*
0x014	WDT_MIS	Watchdog masked interrupt status register	*131*
0xC00	WDT_LOCK	Watchdog lock register	*132*
0xFE0	WDTPeriphID0	Watchdog peripheral identification register 0 (bits 7:0)	*132*
0xFE4	WDTPeriphID1	Watchdog peripheral identification register 1 (bits 15:8)	*133*
0xFE8	WDTPeriphID2	Watchdog peripheral identification register 2 (bits 23:16)	*133*
0xFEC	WDTPeriphID3	Watchdog peripheral identification register 3 (bits 31:24)	*133*
0xFF0	WDTPCellID0	Watchdog PCell identification register 0 (bits 7:0)	*134*
0xFF4	WDTPCellID1	Watchdog PCell identification register 1 (bits 15:8)	*134*
0xFF8	WDTPCellID2	Watchdog PCell identification register 2 (bits 23:16)	*134*
0xFFC	WDTPCellID3	Watchdog PCell identification register 3 (bits 31:24)	*135*

11.3 WDT register descriptions

WDT_LR Watchdog load register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDTLOAD[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTLOAD[15:0]															
RW															

Address: WDTBaseAddress + 0x000

Reset: 0xFFFF FFFF

Description: Contains the value from which the counter decrements. When written to, the count is immediately restarted from the new value. Minimum valid value is 0x1.

WDTLOAD Watchdog load value. Value from which the counter is to decrement.

WDT_VAL Watchdog value register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDTVAL[31:16]															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTVAL[15:0]															
R															

Address: WDTBaseAddress + 0x004

Reset: 0xFFFF FFFF

Description: Gives the current value of the decrementing watchdog counter.

WDTVAL Watchdog value. Returns the decrementing watchdog counter value. Write has no effect.

WDT_CR Watchdog control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	RESEN	INTEN
R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Address: WDTBaseAddress + 0x008

Reset: 0x0000 0000

Description: Configures the watchdog timer.

RESEN Watchdog reset enable. Enables watchdog reset output (WDOGRES). Acts as a mask for reset output.
0: disabled (default) 1: enabled

INTEN Watchdog interrupt enable. Enable the interrupt event (WDOGINT).
0: disabled (default) 1: enabled

0 Reserved for future use. Reading returns 0. Must be written with 0.

WDT_ICR Watchdog interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDTICLR															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTICLR															
RW															

Address: WDTBaseAddress + 0x00C

Reset: 0x0000 0000

Description: Any write to this register clears the interrupt output from the watchdog, and reloads the counter from the value in *WDT_LR* register.

WDTICLR Watchdog interrupt clear clears watchdog interrupt and reloads the counter. Reading returns zero.

WDT_RIS Watchdog raw interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WDTRIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: WDTBaseAddress + 0x010

Reset: 0x0000 0000

Description: This bit is ANDed with the interrupt enable bit from the control register to create the masked interrupt, which is passed to the interrupt output pin.

WDTRIS Watchdog raw interrupt status. Reflects the interrupt status from the watchdog.
0: watchdog interrupt is not set 1: watchdog interrupt is set

0 Reserved for future use. Reading returns 0. Must be written with 0.

WDT_MIS Watchdog masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WDTMIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: WDTBaseAddress + 0x014

Reset: 0x0000 0000

Description: This register value is the logical AND of the raw interrupt status with the timer interrupt enable bit from the control register. It is passed to the interrupt output pin. This register is read-only, and all bits are cleared by a reset.

WDTMIS Watchdog masked interrupt status. Masked value of watchdog interrupt status.
0: watchdog line interrupt not active 1: watchdog line asserting interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

WDT_LOCK Watchdog lock register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCKVAL															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCKVAL															
RW															

Address: WDTBaseAddress + 0x018

Reset: 0x0000 0000

Description: Allows write-access to all other registers to be disabled. This prevents rogue software from disabling the watchdog functionality.

LOCKVAL Watchdog lock value. Returns the lock status (the LSB). When locked, write access to all other watchdog registers is disabled.

0x0: not locked (default) 0x1: locked

When written, enables or disables write access to all other watchdog registers.

0x1ACC E551: enabled Any other value: disabled

WDTPeriphID0 Watchdog peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: WDTBaseAddress + 0xFE0

Reset: 0x0000 0005

Description: PartNumber0 reads back as 0x05.

WDTPeriphID1 Watchdog peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: WDTBaseAddress + 0xFE4

Reset: 0x0000 0018

Description: Designer0 reads back as 0x1. PartNumber1 reads back as 0x8.

WDTPeriphID2 Watchdog peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: WDTBaseAddress + 0xFE8

Reset: 0x0000 0004

Description: Revision reads back as 0x0. Designer1 reads back as 0x4.

WDTPeriphID3 Watchdog peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: WDTBaseAddress + 0xFEC

Reset: 0x0000 0000

Description: Configuration reads back as 0x00.

WDTPCellID0 Watchdog PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	WDTPCellID0							
R	R	R	R	R	R	R	R	R							

Address: WDTBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: WDTPCellID0 reads back as 0x0D.

WDTPCellID1 Watchdog PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	WDTPCellID1							
R	R	R	R	R	R	R	R	R							

Address: WDTBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: WDTPCellID1 reads back as 0xF0.

WDTPCellID2 Watchdog PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	WDTPCellID2							
R	R	R	R	R	R	R	R	R							

Address: WDTBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: WDTPCellID2 reads back as 0x05.

WDTPCellID3 Watchdog PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	WDTPCellID3							
R	R	R	R	R	R	R	R	R							

Address: WDTBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: WDTPCellID3 reads back as 0xB1.

12 Real-time clock (RTC)

12.1 L2CC register addressing

Register addresses are provided as the L2CC base address, L2CCBaseAddress, plus the register offset.

The L2CCBaseAddress is 0x101E 8000.

12.2 L2CC register summary

The device uses registers accessible via a 32-bit width AMBA rev. 2.0 peripheral bus (APB) to communicate with the system.

Table 26. RTC register list

Offset	Register	Description	Page
0x000	RTC_DR	RTC data register	*137*
0x004	RTC_MR	RTC match register	*137*
0x008	RTC_LR	RTC load register	*137*
0x00C	RTC_TCR	RTC trim and control register	*138*
0x010	RTC_IMSC	RTC and RTT interrupt mask register	*138*
0x014	RTC_RIS	RTC and RTT raw interrupt status register	*139*
0x018	RTC_MIS	RTC and RTT masked interrupt status register	*139*
0x01C	RTC_ICR	RTC and RTT interrupt clear register	*140*
0x100	PWL_CR	Pulse-width light modulator control (see the PWM register chapter)	*146*
0xFE0	RTC_PeriphID0	RTC peripheral identification register 0 (bits 7:0)	*140*
0xFE4	RTC_PeriphID1	RTC peripheral identification register 1 (bits 15:8)	*140*
0xFE8	RTC_PeriphID2	RTC peripheral identification register 2 (bits 23:16)	*141*
0xFEC	RTC_PeriphID3	RTC peripheral identification register 3 (bits 31:24)	*141*
0xFF0	RTC_PCellID0	RTC PCell identification register 0 (bits 7:0)	*141*
0xFF4	RTC_PCellID1	RTC PCell identification register 1 (bits 15:8)	*142*
0xFF8	RTC_PCellID2	RTC PCell identification register 2 (bits 23:16)	*142*
0xFFC	RTC_PCellID3	RTC PCell identification register 3 (bits 31:24)	*142*

12.3 L2CC register descriptions

RTC_DR — RTC data register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTC current value															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC current value															
R															

Address: RTCBaseAddress + 0x000

Reset: 0x0000 0000

Description: Returns the current RTC value. It is cleared on power-on reset (PORnot).

RTC_MR — RTC match register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTC match value															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC match value															
RW															

Address: RTCBaseAddress + 0x004

Reset: 0x0000 0000

Description: Writes to this register load the match register, reads return the last written value. It is cleared on a power-on reset (PORnot).

RTC_LR — RTC load register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTC load value															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC load value															
RW															

Address: RTCBaseAddress + 0x0008

Reset: 0x0000 0000

Description: Writes to this register load an update value into the RTC. It is cleared on any reset.

RTC_TCR RTC trim and control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	RTCEN	CKDEL									
R	R	R	R	R	RW	RW									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKDIV															
RW															

Address: RTCBaseAddress + 0x00C

Reset: 0x0000 7FFF

Description: The RTC control, trim divider factor and delete register is cleared on a power-on (PORnot) reset.

RTCEN Counter enable.

1: RTC is enabled, writes to this register have no effect until a power-on reset.

CKDEL Trim delete count. This value represents the number of CLK32K clock pulses to delete every 1023 CLK32K clock cycles to get a better reference 1 Hz clock for incrementing the RTC counter. Writes to this bit-field are disregarded if RTCEN = 1.

0x000: no CLK32K clock cycles are deleted (PORnot reset default).

0x001: 1 CLK32K clock cycle is deleted every 1023 CLK1HZ clock cycles through to

0x3FF: 1023 CLK32K clock cycles are deleted every 1023 CLK1HZ clock cycles.

CKDIV Clock divider factor. This value plus one represents the integer part of the CLK32K clock divider used to produce the reference 1 Hz clock. Writes to this bit-field are disregarded if RTCEN = 1.

0x0000: CLK1HZ clock is stopped.

0x0001: 2 CLK32K clock cycles per CLK1HZ clock cycle through to

0x7FFF: 32768 CLK32K clock cycles per CLK1HZ clock cycle (default PORnot reset value) through

0xFFFF: 65536 CLK32K clock cycles per CLK1HZ clock cycle.

0 Reserved for future use. Reading returns 0. Must be written with 0.

RTC_IMSC RTC interrupt mask register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	RTT IMSC	RTC IMSC
R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Address: RTCBaseAddress + 0x010

Reset: 0x0000 0000

Description: Controls the masking of the interrupts generated by the RTC and RTT. Reading this register returns the value of the mask on the RTC and RTT interrupts.
It is cleared upon a power-on reset only (PORnot).

RTTIMSC RTT interrupt enable.
0: clear interrupt mask (PORnot reset default) 1: set interrupt mask

RTCIMSC RTC interrupt enable.
0: clear interrupt mask (PORnot reset default) 1: set interrupt mask

0 Reserved for future use. Reading returns 0. Must be written with 0.

RTC_RIS RTC raw interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	RTTRIS	RTCRIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: RTCBaseAddress + 0x014

Reset: 0x0000 0000

Description: Gives the raw interrupt status.

RTTRIS RTT raw interrupt status gives the raw interrupt state (prior to masking) of the RTT interrupt.

RTCRIS RTC raw interrupt status gives the raw interrupt state (prior to masking) of the RTC interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

RTC_MIS RTC masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	RTTMIS	RTCMIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: RTCBaseAddress + 0x018

Reset: 0x0000 0000

Description: Indicates the masked interrupt status.

RTTMIS RTT masked interrupt status gives the masked interrupt status (after masking) of the RTT interrupt.

RTCMIS RTC masked interrupt status gives the masked interrupt status (after masking) of the RTC interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

RTC_ICR RTC interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	RTTIC	RTCIC
R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Address: RTCBaseAddress + 0x01C

Reset: 0x0000 0000

Description: Clears interrupts.

RTTIC RTT interrupt clear clears the RTT interrupt.
0: no effect 1: clears the interrupt

RTCIC RTC interrupt clear clears the RTC interrupt.
0: no effect 1: clears the interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

RTC_PeriphID0 RTC peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: RTCBaseAddress + 0xFE0

Reset: 0x0000 0031

Description: PartNumber0 reads back as 0x31.

RTC_PeriphID1 RTC peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: RTCBaseAddress + 0xFE4

Reset: 0x0000 0000

Description: Designer0 reads back as 0x00. PartNumber1 reads back as 0x00.

RTC_PeriphID2 RTC peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: RTCBaseAddress + 0xFE8

Reset: 0x0000 0018

Description: Revision reads back as 0x01. Designer1 reads back as 0x08.

RTC_PeriphID3 RTC peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: RTCBaseAddress + 0xFEC

Reset: 0x0000 0000

Description: Configuration reads back as 0x00.

RTC_PCellID0 RTC PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	RTCPCellID0							
R	R	R	R	R	R	R	R	R							

Address: RTCBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: RTCPCellID0 reads back as 0x0D.

RTC_PCellID1 RTC PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	RTCPCellID1							
R	R	R	R	R	R	R	R	R							

Address: RTCBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: RTCPCellID1 reads back as 0xF0.

RTC_PCellID2 RTC PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	RTCPCellID2							
R	R	R	R	R	R	R	R	R							

Address: RTCBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: RTCPCellID2 reads back as 0x05.

RTC_PCellID3 RTC PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	RTCPCellID3							
R	R	R	R	R	R	R	R	R							

Address: RTCBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: RTCPCellID3 reads back as 0xB1.

13 Real-time timer (RTT)

13.1 RTT register addressing

Register addresses are provided as the L2CC base address, L2CCBaseAddress, plus the register offset.

The L2CCBaseAddress is 0x101E 8000.

Note: The RTC module hosts the real time timer registers so the RTT base address is the same as the RTC base address. The interrupt registers are shared between the RTC and the RTT. The device communicates with the system via a control register accessible via a 32-bit width AMBA rev 2.0 peripheral bus (APB).

13.2 RTT register summary

Table 27. RTT register list

Offset	Register	Description	Page
0x010	RTC_IMSC	RTC and RTT shared interrupt mask set and clear register	*138*
0x014	RTC_RIS	RTC and RTT shared raw interrupt status register	*139*
0x018	RTC_MIS	RTC and RTT shared masked interrupt status register	*139*
0x01C	RTC_ICR	RTC and RTT shared interrupt clear register	*140*
0x020	RTT_DR	RTT data register	*143*
0x024	RTT_LR	RTT load register	*144*
0x028	RTT_CR	RTT control register	*144*

13.3 RTT register descriptions

RTT_DR RTT data register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTT current value															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTT current value															
R															

Address: RTC base address + 0x020

Reset: 0x0000 0000

Description: This 32-bit read-only data register returns the current value of the RTT. Reads can occur at any time. The 32-bit counter value is re-synchronized on PCLK clock domain

before being driven on the PRDATA bus. Writes have no effect. It is only cleared on power-on reset (PORnot).

RTT_LR RTT load register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RTT load value															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTT load value															
RW															

Address: RTC base address + 0x024

Reset: 0x0000 0000

Description: The RTT counter must be halted before writing to this register. Writing to this register when *RTT_CR*.RTTEN = 1 has no effect.
Writing to this register when RTT_CR.RTTEN = 0, sets this bit to 1 and restarts the counter from the new written value (in both periodic and one-shot modes). It is cleared on power-on reset only (PORnot).

RTT_CR RTT control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	RTT EN	RTT OS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	RWH	RW

Address: RTC base address + 0x028

Reset: 0x0000 7FFF

Description: When high, the counter enable signal is asserted to enable the counter. This register is cleared upon a power-on (PORnot) reset only.

RTTEN RTT enable. This bit is cleared by hardware when the counter reaches zero in one-shot mode and set by hardware when the *RTT_LR* register is written to while the counter is stopped. A read returns the value of the RTTEN bit. When written,

0: the counter is stopped on the next CLK32K cycle. When the RTT is stopped, the content of the counter is frozen.

1: RTT is enabled on the next CLK32K cycle.

RTTOS Timer x one shot count.

0: Timer x is in periodic mode (default). On reaching zero, the RTT raises its interrupt and is reloaded from the RTT_LR content.

1: Timer x is in one-shot mode. On reaching zero, the RTT raises its interrupt and stops.

0 Reserved for future use. Reading returns 0. Must be written with 0.

Note: Two consecutive writes to this register must be separated by at least 1 CLK32K period (31 µs). If this is not the case, the last value written is not guaranteed. The software can read back the RTT_CR register at least one CLK32K period after a write to see its new value.

14 Pulse width light modulator (PWL)

14.1 PWL register addressing

Register addresses are provided as the L2CC base address, L2CCBaseAddress, plus the register offset.

The L2CCBaseAddress is 0x101E 8000.

Note: The RTC module hosts the pseudo-noise pulse width light modulator logic and control register so the PWL base address is the same as the RTC base address.The device communicates with the system through a register accessible via a 32-bit width AMBA rev 2.0 peripheral bus (APB).

14.2 PWL register summary

Table 28. PWL register list

Offset	Register	Description	Page
0x100	PWL_CR	PWL control register	146

14.3 PWL register descriptions

PWL_CR PWL control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PWLEN				PWLVL				
R	R	R	R	R	R	R	RW				RW				

Address: RTCBaseAddress + 0x0100

Reset: 0x0000 0000

Description: Can be changed at any time without producing any glitches on the PWL output signal, as it is re-synchronized to the CLK32K clock domain. It is cleared on a power-on reset only (PORnot).

PWLEN PWLEN enables the PWL activity and output signal.
0: stop activity, output is always low 1: enable activity, toggle O/P according to PWLVL

PWLVL PWL level. Defines the mean value of the PWL output signal in terms of CLK32K cycles.
0x00: high for 1/256 cycles and so on until 0xFE: high for 255/256 cycles, 0xFF: always high

0 Reserved for future use. Reading returns 0. Must be written with 0.

Note: Two consecutive writes must be separated by at least 1 CLK32K period (31 µs). If not, the last value written is not guaranteed. The software must wait at least one CLK32K period after a write before reading its new value in the PWL_CR register.

15 Vectored interrupt controller (L2CC)

15.1 VIC register addressing

Register addresses are provided as the VIC base address, VICBaseAddress, plus the register offset.

The VICBaseAddress is 0x1014 0000.

15.2 VIC register summary

Table 29. List of VIC registers

Address	Register	Description	Page
0x000	VIC_IRQSR0	IRQ status register 0	*150*
0x004	VIC_FIQSR0	FIQ status register 0	*151*
0x008	VIC_RIS0	Raw interrupt status register 0	*152*
0x00C	VIC_ISEL0	Interrupt select register 0	*152*
0x010	VIC_IENS0	Interrupt enable set register 0	*153*
0x014	VIC_IENC0	Interrupt enable clear register 0	*154*
0x018	VIC_SWISR0	Software interrupt set register 0	*154*
0x01C	VIC_SWICR0	Software interrupt clear register 0	*155*
0x020	VIC_IRQSR1	IRQ status register 1	*150*
0x024	VIC_FIQSR1	FIQ status register 1	*151*
0x028	VIC_RIS1	Raw interrupt status register 1	*152*
0x02C	VIC_ISEL1	Interrupt select register 1	*152*
0x030	VIC_IENS1	Interrupt enable set register 1	*153*
0x034	VIC_IENC1	Interrupt enable clear register 1	*154*
0x038	VIC_SWISR1	Software interrupt set register 1	*154*
0x03C	VIC_SWICR1	Software interrupt clear register 1	*155*
0x040	VIC_PER	Protection enable register	*155*
0x050	VIC_VAR	Vector address register	*156*
0x054	VIC_DVAR	Default vector address register	*156*
0x100	VIC_VAR0	Vector address 0 register	*157*
0x104	VIC_VAR1	Vector address 1 register	*157*
0x108	VIC_VAR2	Vector address 2 register	*157*
0x10C	VIC_VAR3	Vector address 3 register	*157*
0x110	VIC_VAR4	Vector address 4 register	*157*
0x114	VIC_VAR5	Vector address 5 register	*157*

Table 29. List of VIC registers (continued)

Address	Register	Description	Page
0x118	VIC_VAR6	Vector address 6 register	157
0x11C	VIC_VAR7	Vector address 7 register	157
0x120	VIC_VAR8	Vector address 8 register	157
0x124	VIC_VAR9	Vector address 9 register	157
0x128	VIC_VAR10	Vector address 10 register	157
0x12C	VIC_VAR11	Vector address 11 register	157
0x130	VIC_VAR12	Vector address 12 register	157
0x134	VIC_VAR13	Vector address 13 register	157
0x138	VIC_VAR14	Vector address 14 register	157
0x13C	VIC_VAR15	Vector address 15 register	157
0x200	VIC_VCR0	Vector control 0 register	157
0x204	VIC_VCR1	Vector control 1 register	157
0x208	VIC_VCR2	Vector control 2 register	157
0x20C	VIC_VCR3	Vector control 3 register	157
0x210	VIC_VCR4	Vector control 4 register	157
0x214	VIC_VCR5	Vector control 5 register	157
0x218	VIC_VCR6	Vector control 6 register	157
0x21C	VIC_VCR7	Vector control 7 register	157
0x220	VIC_VCR8	Vector control 8 register	157
0x224	VIC_VCR9	Vector control 9 register	157
0x228	VIC_VCR10	Vector control 10 register	157
0x22C	VIC_VCR11	Vector control 11 register	157
0x230	VIC_VCR12	Vector control 12 register	157
0x234	VIC_VCR13	Vector control 13 register	157
0x238	VIC_VCR14	Vector control 14 register	157
0x23C	VIC_VCR15	Vector control 15 register	157
0xFE0	VICPeriphID0	Peripheral identification register 0 (bits 7:0)	157
0xFE4	VICPeriphID1	Peripheral identification register 1 (bits 15:8)	158
0xFE8	VICPeriphID2	Peripheral identification register 2 (bits 23:16)	158
0xFEC	VICPeriphID3	Peripheral identification register 3 (bits 31:24)	158
0xFF0	VICPCellID0	PCell identification register 0 (bits 7:0)	159
0xFF4	VICPCellID1	PCell identification register 1 (bits 15:8)	159
0xFF8	VICPCellID2	PCell identification register 2 (bits 23:16)	159
0xFFC	VICPCellID3	PCell identification register 3 (bits 31:24)	160

15.3 VIC register descriptions

VIC_IRQSR0/1 IRQ interrupt status registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQS31	IRQS30	IRQS29	IRQS28	IRQS27	IRQS26	IRQS25	IRQS24	IRQS23	IRQS22	IRQS21	IRQS20	IRQS19	IRQS18	IRQS17	IRQS16
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQS15	IRQS14	IRQS13	IRQS12	IRQS11	IRQS10	IRQS9	IRQS8	IRQS7	IRQS6	IRQS5	IRQS4	IRQS3	IRQS2	IRQS1	IRQS0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQS63	IRQS62	IRQS61	IRQS60	IRQS59	IRQS58	IRQS57	IRQS56	IRQS55	IRQS54	IRQS53	IRQS52	IRQS51	IRQS50	IRQS49	IRQS48
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IRQS47	IRQS46	IRQS45	IRQS44	IRQS43	IRQS42	IRQS41	IRQS40	IRQS39	IRQS38	IRQS37	IRQS36	IRQS35	IRQS34	IRQS33	IRQS32
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: VICBaseAddress + 0x000, 0x020

Reset: 0x0000 0000

Description: Provide the status of the interrupts 0 to 63 after IRQ masking (that is, after the operation of discrimination between the IRQ and the FIQ interrupt requests). The masking is operated by two register pairs:

- *VIC_IENS0* (for interrupts 0 to 31), *VIC_IENS1* (for interrupts 32 to 63),
- *VIC_ISEL0* (for interrupts 0 to 31), *VIC_ISEL1* (for interrupts 32 to 63).

IRQS[0:63] Interrupt request status. Indicates if interrupt line is active for IRQ generation.
0: inactive or masked for IRQ generation 1: active, and generates an IRQ to the processor

VIC_FIQSR0/1 FIQ interrupt status registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIQS31	FIQS30	FIQS29	FIQS28	FIQS27	FIQS26	FIQS25	FIQS24	FIQS23	FIQS22	FIQS21	FIQS20	FIQS19	FIQS18	FIQS17	FIQS16
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIQS15	FIQS14	FIQS13	FIQS12	FIQS11	FIQS10	FIQS9	FIQS8	FIQS7	FIQS6	FIQS5	FIQS4	FIQS3	FIQS2	FIQS1	FIQS0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FIQS63	FIQS62	FIQS61	FIQS60	FIQS59	FIQS58	FIQS57	FIQS56	FIQS55	FIQS54	FIQS53	FIQS52	FIQS51	FIQS50	FIQS49	FIQS48
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIQS47	FIQS46	FIQS45	FIQS44	FIQS43	FIQS42	FIQS41	FIQS40	FIQS39	FIQS38	FIQS37	FIQS36	FIQS35	FIQS34	FIQS33	FIQS32
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: VICBaseAddress + 0x004, 0x024

Reset: 0x0000 0000

Description: Provide the status of the interrupts 0 to 63, after FIQ masking (that is, after the operation of discrimination between the IRQ and the FIQ interrupt requests). The masking is operated by two register pairs:

- *VIC_IENS0* (for interrupts 0 to 31), *VIC_IENS1* (for interrupts 32 to 63),
- *VIC_ISEL0* (for interrupts 0 to 31), *VIC_ISEL1* (for interrupts 32 to 63).

FIQS[0:63] Fast interrupt request status. Indicates if interrupt line is active for FIQ generation.
0: inactive or masked for FIQ generation 1: active, and generates a FIQ to the processor

VIC_RIS0/1 Raw interrupt status registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RIS 31	RIS 30	RIS 29	RIS 28	RIS 27	RIS 26	RIS 25	RIS 24	RIS 23	RIS 22	RIS 21	RIS 20	RIS 19	RIS 18	RIS 17	RIS 16
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RIS 15	RIS 14	RIS 13	RIS 12	RIS 11	RIS 10	RIS 9	RIS 8	RIS 7	RIS 6	RIS 5	RIS 4	RIS 3	RIS 2	RIS 1	RIS 0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RIS 63	RIS 62	RIS 61	RIS 60	RIS 59	RIS 58	RIS 57	RIS 56	RIS 55	RIS 54	RIS 53	RIS 52	RIS 51	RIS 50	RIS 49	RIS 48
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RIS 47	RIS 46	RIS 45	RIS 44	RIS 43	RIS 42	RIS 41	RIS 40	RIS 39	RIS 38	RIS 37	RIS 36	RIS 35	RIS 34	RIS 33	RIS 32
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: VICBaseAddress + 0x008, 0x028

Reset: 0x0000 0000

Description: Provide the status of interrupts 0 to 63 before masking by the enable registers.

RIS[0:63] Raw interrupt. Indicates the interrupt request status before masking.
0: inactive 1: active

VIC_ISEL0/1 Interrupt selection registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISEL 31	ISEL 30	ISEL 29	ISEL 28	ISEL 27	ISEL 26	ISEL 25	ISEL 24	ISEL 23	ISEL 22	ISEL 21	ISEL 20	ISEL 19	ISEL 18	ISEL 17	ISEL 16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISEL 15	ISEL 14	ISEL 13	ISEL 12	ISEL 11	ISEL 10	ISEL 9	ISEL 8	ISEL 7	ISEL 6	ISEL 5	ISEL 4	ISEL 3	ISEL 2	ISEL 1	ISEL 0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISEL 63	ISEL 62	ISEL 61	ISEL 60	ISEL 59	ISEL 58	ISEL 57	ISEL 56	ISEL 55	ISEL 54	ISEL 53	ISEL 52	ISEL 51	ISEL 50	ISEL 49	ISEL 48
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISEL 47	ISEL 46	ISEL 45	ISEL 44	ISEL 43	ISEL 42	ISEL 41	ISEL 40	ISEL 39	ISEL 38	ISEL 37	ISEL 36	ISEL 35	ISEL 34	ISEL 33	ISEL 32
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: VICBaseAddress + 0x00C, 0x02C

Reset: 0x0000 0000

Description: Select the type of processor request (IRQ or FIQ) for interrupts 0 to 63.

ISEL[0:63] Interrupt select. Selects if interrupt line is used for IRQ or FIQ generation.
0: IRQ interrupt 1: FIQ interrupt

VIC_IENS0/1 Interrupt enable set registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IENS 31	IENS 30	IENS 29	IENS 28	IENS 27	IENS 26	IENS 25	IENS 24	IENS 23	IENS 22	IENS 21	IENS 20	IENS 19	IENS 18	IENS 17	IENS 16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IENS 15	IENS 14	IENS 13	IENS 12	IENS 11	IENS 10	IENS 9	IENS 8	IENS 7	IENS 6	IENS 5	IENS 4	IENS 3	IENS 2	IENS 1	IENS 0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IENS 63	IENS 62	IENS 61	IENS 60	IENS 59	IENS 58	IENS 57	IENS 56	IENS 55	IENS 54	IENS 53	IENS 52	IENS 51	IENS 50	IENS 49	IENS 48
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IENS 47	IENS 46	IENS 45	IENS 44	IENS 43	IENS 42	IENS 41	IENS 40	IENS 39	IENS 38	IENS 37	IENS 36	IENS 35	IENS 34	IENS 33	IENS 32
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: VICBaseAddress + 0x010, 0x030

Reset: 0x0000 0000

Description: Enable the interrupt request lines 0 to 63 for IRQ or FIQ generation.

IENS[0:63] Interrupt enable set. On reset all interrupts are disabled.
Write 0: no effect 1: sets corresponding *VIC_IENS* register bit
Read 0: interrupt disabled 1: interrupt enabled

VIC_IENC0/1 Interrupt enable clear registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IENC 31	IENC 30	IENC 29	IENC 28	IENC 27	IENC 26	IENC 25	IENC 24	IENC 23	IENC 22	IENC 21	IENC 20	IENC 19	IENC 18	IENC 17	IENC 16
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IENC 15	IENC 14	IENC 13	IENC 12	IENC 11	IENC 10	IENC 9	IENC 8	IENC 7	IENC 6	IENC 5	IENC 4	IENC 3	IENC 2	IENC 1	IENC 0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IENC 63	IENC 62	IENC 61	IENC 60	IENC 59	IENC 58	IENC 57	IENC 56	IENC 55	IENC 54	IENC 53	IENC 52	IENC 51	IENC 50	IENC 49	IENC 48
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IENC 47	IENC 46	IENC 45	IENC 44	IENC 43	IENC 42	IENC 41	IENC 40	IENC 39	IENC 38	IENC 37	IENC 36	IENC 35	IENC 34	IENC 33	IENC 32
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Address: VICBaseAddress + 0x014, 0x034

Reset: Undefined

Description: Disable the interrupt request lines 0 to 63.

IENC[0:63] Interrupt enable clear.

0: no effect 1: clears corresponding *VIC_IENS* register bit

VIC_SWISR0/1 Software interrupt set registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIS 31	SWIS 30	SWIS 29	SWIS 28	SWIS 27	SWIS 26	SWIS 25	SWIS 24	SWIS 23	SWIS 22	SWIS 21	SWIS 20	SWIS 19	SWIS 18	SWIS 17	SWIS 16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIS 15	SWIS 14	SWIS 13	SWIS 12	SWIS 11	SWIS 10	SWIS 9	SWIS 8	SWIS 7	SWIS 6	SWIS 5	SWIS 4	SWIS 3	SWIS 2	SWIS 1	SWIS 0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIS 63	SWIS 62	SWIS 61	SWIS 60	SWIS 59	SWIS 58	SWIS 57	SWIS 56	SWIS 55	SWIS 54	SWIS 53	SWIS 52	SWIS 51	SWIS 50	SWIS 49	SWIS 48
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIS 47	SWIS 46	SWIS 45	SWIS 44	SWIS 43	SWIS 42	SWIS 41	SWIS 40	SWIS 39	SWIS 38	SWIS 37	SWIS 36	SWIS 35	SWIS 34	SWIS 33	SWIS 32
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: VICBaseAddress + 0x018, 0x038

Reset: 0x0000 0000

Description: Generate software interrupts before interrupt masking.

SWIS[0:63] Software interrupt set. Generates a software interrupt for a source interrupt before interrupt masking.
0: no effect 1: sets the corresponding bit

VIC_SWICR0/1 Software interrupt clear registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIC 31	SWIC 30	SWIC 29	SWIC 28	SWIC 27	SWIC 26	SWIC 25	SWIC 24	SWIC 23	SWIC 22	SWIC 21	SWIC 20	SWIC 19	SWIC 18	SWIC 17	SWIC 16
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIC 15	SWIC 14	SWIC 13	SWIC 12	SWIC 11	SWIC 10	SWIC 9	SWIC 8	SWIC 7	SWIC 6	SWIC 5	SWIC 4	SWIC 3	SWIC 2	SWIC 1	SWIC 0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIC 63	SWIC 62	SWIC 61	SWIC 60	SWIC 59	SWIC 58	SWIC 57	SWIC 56	SWIC 55	SWIC 54	SWIC 53	SWIC 52	SWIC 51	SWIC 50	SWIC 49	SWIC 48
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIC 47	SWIC 46	SWIC 45	SWIC 44	SWIC 43	SWIC 42	SWIC 41	SWIC 40	SWIC 39	SWIC 38	SWIC 37	SWIC 36	SWIC 35	SWIC 34	SWIC 33	SWIC 32
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Address: VICBaseAddress + 0x01C, 0x03C

Reset: Undefined

Description: These write-only registers clear the software interrupts.

SWIC[0:63] Software interrupt clear. Clears bits in the *VIC_SWISR0/1* register.
0: no effect 1: clears the corresponding VIC_SWISR0/1 bit

VIC_PER VIC protection enable register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PROT
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Address: VICBaseAddress + 0x040

Reset: 0x0000 0000

Description: Moves between privilege mode and user mode within the VIC behavior.

PROT Protection. Enables or disables protected register access. When enabled, only privileged mode accesses (reads and writes) can access the interrupt controller registers. When disabled, both user mode and privileged mode can access the registers. This register is cleared on reset, and can only be accessed in privileged mode.

0 Reserved for future use. Reading returns 0. Must be written with 0.

VIC_VAR ISR vector address register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISRA[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISRA[15:0]															
RW															

Address: VICBaseAddress + 0x050

Reset: 0x0000 0000

Description: Contains the interrupt service routine (ISR) vector address of the currently active interrupt.

ISRA Interrupt service routine address. When read, returns the address of the currently active ISR. Any writes to this register clear the interrupt.

VIC_DVAR ISR default vector address register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRA[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRA[15:0]															
RW															

Address: VICBaseAddress + 0x054

Reset: 0x0000 0000

Description: Contains the default interrupt service routine (ISR) vector address.

DRA Default interrupt service routine address. Contains the address of the default ISR handler.

VIC_VAR[0:15] ISR vector address registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A[31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A[15:0]															
RW															

Address: VICBaseAddress + 0x0100 + 4*x (x = 0 to 15)

Reset: 0x0000 0000

Description: These sixteen registers contain the ISR vector addresses.

A[0:15] Contains the ISR vector address for vectored interrupt x.

VIC_VCR[0:15] Vectored interrupt control registers

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	E	IS					
R	R	R	R	R	R	R	R	R	RW	RW					

Address: VICBaseAddress + 0x0200 + 4 * x (x = 0 to 15)

Reset: 0x0000 0000

Description: These sixteen registers select the interrupt source for the vectored interrupt.

E Enable interrupt [0:15]. Enables vector interrupt. This bit is cleared on reset.

IS Interrupt [0:15] source. Selects any of the 64 interrupt sources for interrupt [0:15].

0 Reserved for future use. Reading returns 0. Must be written with 0.

VICPeriphID0 VIC peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: VICBaseAddress + 0xFE0

Reset: 0x0000 0090

Description: PartNumber0 reads back as 0x90.

VICPeriphID1 VIC peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: VICBaseAddress + 0xFE4

Reset: 0x0000 0001

Description: Designer0 reads back as 0x0. PartNumber1 reads back as 0x1.

VICPeriphID2 VIC peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: VICBaseAddress + 0xFE8

Reset: 0x0000 0008

Description: Revision reads back as 0x0. Designer1 reads back as 0x8.

VICPeriphID3 VIC peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: VICBaseAddress + 0xFEC

Reset: 0x0000 0001

Description: Configuration reads back as 0x01: 64 interrupt lines.

VICPCellID0 VIC PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	VICPCellID0							
R	R	R	R	R	R	R	R	R							

Address: VICBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: VICPCellID0 reads back as 0x0D.

VICPCellID1 VIC PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	VICPCellID1							
R	R	R	R	R	R	R	R	R							

Address: VICBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: VICPCellID1 reads back as 0xF0.

VICPCellID2 VIC PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	VICPCellID2							
R	R	R	R	R	R	R	R	R							

Address: VICBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: VICPCellID2 reads back as 0x05.

VICPCellID3 VIC PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	VICPCellID3							
R	R	R	R	R	R	R	R	R							

Address: VICBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: VICPCellID3 reads back as 0xB1.

16 SDRAM memory controller (SDMC)

16.1 SDMC register addressing

Register addresses are provided as the SDRAM memory controller base address, SDMCBaseAddress, plus the register offset.

The SDMCBaseAddress is 0x1011 0000.

16.2 SDMC register summary

The SDMC has to be configured according the external memory devices used, via registers.

Table 30. SDMC register list

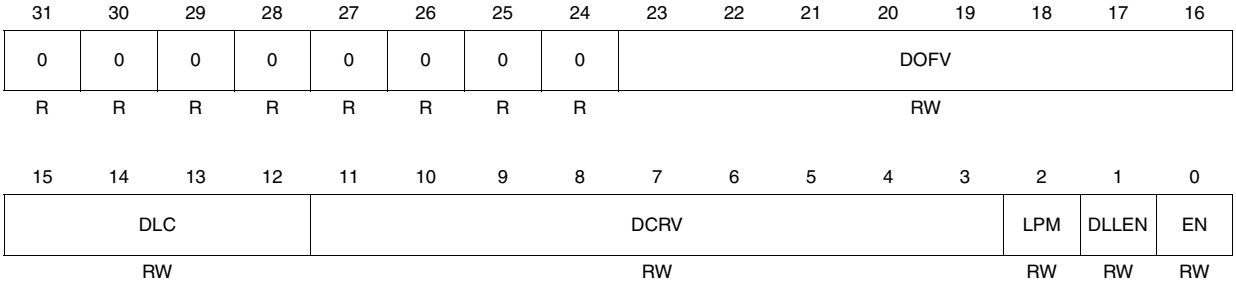
Offset	Register name	Description	Page
0x000	SDMC_CR	SDMC control register	*163*
0x004	SDMC_SR	SDMC status register	*164*
0x008	SDMC_GCFR	SDMC global configuration register	*164*
0x020	SDMC_DYCR	SDMC dynamic control register	*165*
0x024	SDMC_DYREF	SDMC dynamic refresh register	*166*
0x028	SDMC_DYRDCFR	SDMC dynamic read configuration register	*166*
0x030	SDMC_DYRP	SDMC dynamic t_{RP} register	*167*
0x034	SDMC_DYRAS	Dynamic memory precharge period (t_{RAS})	*168*
0x038	SDMC_DYSREX	Dynamic memory self refresh exit time (t_{SREX})	*168*
0x044	SDMC_DYWR	Dynamic memory write recovery time (t_{WR}, t_{DPL}, t_{RWL} or t_{RDL})	*169*
0x048	SDMC_DYRC	Dynamic memory active to active command period= (t_{RC})	*169*
0x04C	SDMC_DYRFC	Dynamic memory auto refresh period and auto refresh to active command period (t_{RFC})	*170*
0x050	SDMC_DYXSR	Dynamic memory exit self refresh to active command time (t_{XSR})	*171*
0x054	SDMC_DYRRD	Dynamic memory active bank A to active bank B time (t_{RRD})	*171*
0x058	SDMC_DYMRD	Dynamic memory load mode to active command time (t_{MRD})	*172*
0x05C	SDMC_DYCDLR	Dynamic memory last data-in to new read/write command time (t_{CDLR})	*172*
0x100	SDMC_DYCFG0	Dynamic memory configuration, chip select 0	*173*
0x104	SDMC_DYRASCAS0	Dynamic memory RAS and CAS delay, chip select 0	*174*
0x120	SDMC_DYCFG1	Dynamic memory configuration, chip select 1	*173*
0x124	SDMC_DYRASCAS1	Dynamic memory RAS and CAS delay, chip select 1	*174*
0x400	SDMC_IBCR0	Internal bus 0 control	*175*
0x404	SDMC_IBSR0	Internal bus 0 status	*176*
0x408	SDMC_IBTOR0	Internal bus 0 time out	*176*

Table 30. SDMC register list (continued)

Offset	Register name	Description	Page
0x420	SDMC_IBCR1	Internal bus 1 control	175
0x424	SDMC_IBSR1	Internal bus 1 status	176
0x428	SDMC_IBTOR1	Internal bus 1 time out	176
0x440	SDMC_IBCR2	Internal bus 2 control	175
0x444	SDMC_IBSR2	Internal bus 2 status	176
0x448	SDMC_IBTOR2	Internal bus 2 time out	176
0x460	SDMC_IBCR3	Internal bus 3 control	175
0x464	SDMC_IBSR3	Internal bus 3 status	176
0x468	SDMC_IBTOR3	Internal bus 3 time out	176
0x480	SDMC_IBCR4	Internal bus 4 control	175
0x484	SDMC_IBSR4	Internal bus 4 status	176
0x488	SDMC_IBTOR4	Internal bus 4 time out	176
0x4A0	SDMC_IBCR5	Internal bus 5 control	175
0x4A4	SDMC_IBSR5	Internal bus 5 status	176
0x4A8	SDMC_IBTOR5	Internal bus 5 time out	176
0xFD0	SDMCPeriphID4	Peripheral ID bits [39:32]	177
0xFD4	SDMCPeriphID5	Reserved (peripheral ID)	177
0xFD8	SDMCPeriphID6	Reserved (peripheral ID)	177
0xFDC	SDMCPeriphID7	Reserved (peripheral ID)	177
0xFE0	*SDMCPeriphID0*	Peripheral identification register (bits 7:0)	178
0xFE4	*SDMCPeriphID1*	Peripheral identification register (bits 15:8)	178
0xFE8	*SDMCPeriphID2*	Peripheral identification register (bits 23:16)	178
0xFEC	*SDMCPeriphID3*	Peripheral identification register (bits 31:24)	179
0xFF0	*SDMCPCellID0*	PCell identification register 0 (bits 7:0)	179
0xFF4	*SDMCPCellID1*	PCell identification register 1 (bits 15:8)	179
0xFF8	*SDMCPCellID2*	PCell identification register 2 (bits 23:16)	179
0xFFC	*SDMCPCellID3*	PCell identification register 3 (bits 31:24)	179

16.3 SDMC register descriptions

SDMC_CR SDMC control register



Address: SDMCBaseAddress + 0x000

Reset: 0x0000 0001

Description: Controls the memory controller operation. It can be altered during normal operation and can be accessed with zero wait states.

DOFV DLL offset value. Signed value that is added to the DLL delay line command.

DLC DLL lock control. Defines the number of consecutive up/down pulses that must be detected to trigger the lock status (lock goes high).

0000: 5 (default value) 0001: 2 (lock generation very sensitive to jitter)
0010: 3 0011: 4
0100: 5 0101: 6
0110: 7 0111: 8
1000: 9 1001: 10
1010: 11 Others: 5

DCRV DLL command value in slow mode (19.2 MHz input frequency). Delay value of the DLL counter when the DLL is in open loop i.e. in slow mode

0_0000_0000: minimum delay (250ps maximum in slow case)...

1_1111_1111: maximum delay (3.5ns minimum in fast case (recommended)).

LPM Low-power mode. Entering low-power mode reduces memory controller power consumption. Dynamic memory is refreshed as necessary. The memory controller returns to normal functional mode by clearing the low-power mode bit (or by system or power-on reset).

0: normal (PORnot reset value and system reset) 1: low power

DLLEN DLL enable. Each time the system frequency (AHB clock) is changed, the DLL must be stopped (cleared) and re-enabled when the system frequency has reached its target value. This allows DLL tracking logic to be cleared and to restart locking on new system frequency.

0: stopped (PORnot reset value and system reset) 1: enabled (running)

EN SDMC enable. When the memory controller is disabled the memory is not refreshed and the SDMC has reduced power consumption. The memory controller is enabled by setting the enable bit, or by system or power-on reset. The SDMC produces an error response (on HRESP) to memory accesses when the EN bit is low. SDMC registers can still be accessed.

0: disabled 1: enabled (PORnot reset value and system reset)

0 Reserved for future use. Reading returns 0. Must be written with 0.

SDMC_SR SDMC status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	DLK	SRA	0	BUSY
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: SDMCBaseAddress + 0x004

Reset: 0x0000 0005

Description: Provides SDMC status information. This register can be accessed with zero wait states.

SRA Self-refresh acknowledge. Indicates the operating mode of the SDMC.
0: normal mode 1: self-refresh acknowledge (PORnot reset value)

BUSY Busy bit. Determines if the SDMC is busy (performing memory transactions, commands or auto-refresh cycles) or not, to ensure that the memory controller enters low-power or disabled mode correctly.
0: idle (reset value on system reset) 1: busy or self-refresh mode (PORnot reset value)

DLK DLL lock. Indicates the lock condition of the DLL in the physical interface.
0: not locked (reset value on system reset) 1: locked

0 Reserved for future use. Reading returns 0. Must be written with 0.

SDMC_GCFR SDMC global configuration register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	END
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Address: SDMCBaseAddress + 0x008

Reset: 0x0000 0000

Address: Configures the memory controller operation. Should only be modified during system initialization. It is accessed with 1 wait state.

END Endianness. When switching between little-endian and big-endian mode all data in the SDMC should be flushed.
0: little-endian mode (PORnot reset value) 1: big-endian mode

0 Reserved for future use. Reading returns 0. Must be written with 0.

SDMC_DYCR SDMC dynamic control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	RP	DS	0	DLLE	DLLC	DLLS	INIT		0	IMMC	0	SR MMC	SR	CS	CE
R	RW	RW	R	RW	RW	R	RW		R	RW	RW	RW	RW	RW	RW

Address: SDMCBaseAddress + 0x020

Reset: 0x0000 0006

Description: Controls the dynamic memory operation. It can be altered during normal operation and can be accessed with zero wait states.

RP Set SyncFlash reset/power-down signal SDRPOUT.
0: low (PORnot reset value) 1: high

DS Low-power SDRAM deep-sleep mode.
0: normal operation (PORnot reset value) 1: enter deep power-down mode

DLLE DLL enable (not used). Enables DLL hand-shaking during auto-refresh cycles:
0: disabled (PORnot reset value) 1: enabled

DLLC DLL calibrate (not used). Enables software control of the DLL hand-shaking for initial DLL calibration.
0: disabled (PORnot reset value) 1: enabled

DLLS DLL status (not used). Indicates the DLL calibration acknowledge signal for software control of DLL hand-shaking.
0: DLL calibrating 1: DLL calibration terminated

INIT SDRAM initialization.
00: issue SDRAM normal operation (PORnot reset value) 01: issue SDRAM mode command
10: issue SDRAM PALL (precharge all) command 11: issue SDRAM NOP (no operation) command

IMMC Inverted memory clock control. Disabling SDRCKN can be performed if there are no DDR-SDRAM memory transactions requiring a differential clock. When enabled, this field can be used in conjunction with the dynamic memory clock control (CS) field.
0: SDRCKN enabled (PORnot reset value) 1: SDRCKN disabled

SRMMC Self-refresh memory clock control (SDRCKP and SDRCKN). Supported by SDR-SDRAM and DDR-SDRAM.
0: run continuously (PORnot reset value) 1: stop during self-refresh mode

SR Self-refresh request. *SDMC_SR*.SRA should be polled for the current operating mode of the SDMC.
0: normal mode (PORnot reset value) 1: enter self-refresh mode under software control

CS Dynamic memory clock control (SDRCKP and SDRCKN).
0: stop when SDRAMs are idle and during self-refresh mode.
1: run continuously (PORnot reset value).

CE Dynamic memory clock enable signal. SDRCKE control. Must be high during SDRAM initialization.
0: SDRCKE of idle device is de-asserted to save power (PORnot reset value).
1: SDRCKE is driven high continuously.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SDMC_DYREF SDMC dynamic refresh register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	REFRESH										
R	R	R	R	R	RW										

Address: SDMCBaseAddress + 0x024

Reset: 0x0000 0000

Description: Configures the dynamic memory operation. Should normally only be modified during system initialization, but these control bits can, if necessary, be altered during normal operation. It is accessed with 1 wait state.

REFRESH Refresh timer.

0x0: refresh disabled (PORnot reset value).

0x1: 1(x 16) = 16 HCLK ticks between SDRAM refresh cycles.

0x8: 8(x 16) = 128 HCLK ticks between SDRAM refresh cycles.

0x1 to 0x7FF: n(x 16) = 16n HCLK ticks between SDRAM refresh cycles.

0 Reserved for future use. Reading returns 0. Must be written with 0.

Example: for the refresh period of 16µs, and an HCLK frequency of 132 MHz, the following value should be programmed into this register: 16.10^{-6} x 132.10^{6} / 16 = 132 (or 0x84 hexadecimal).

Note: The refresh cycles are evenly distributed. However, there may be slight variations when the auto-refresh command is issued depending on the status of the memory controller.

SDMC_DYRDCFR SDMC dynamic read configuration register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	DRD POL	0	0	DRDSTRA		0	0	0	SRD POL	0	0	SRDSTRA	
R	R	R	RW	R	R	RW		R	R	R	RW	R	R	RW	

Address: SDMCBaseAddress + 0x028

Reset: 0x0000 1111

Description: Configures the dynamic memory delay between the command (address, control signals and write data) and the output clock (SDRCKP) strategy. It must only be modified during system initialization. It is accessed with 1 wait state.

DRDPOL DDR-SDRAM read data capture polarity from HCLK edge. Unaffected by system (AHB) reset.
0: negative edge 1: positive edge (PORnot reset value)

DRDSTRA DDR-SDRAM read data strategy. Unaffected by system (AHB) reset.
00: clock delay strategy. SDRCKP must be delayed externally to at least meet the SDRAM input setup time (t_{AS}).
01: command delayed strategy. Commands are delivered on pins on falling edge of SDRCKP clock (command delayed by 180° shifting). Adds 1 clock cycle latency, but gives more relaxed timing, compatible for SDRCKP freq > 100 MHz (reset value after PORnot reset).
10: command delayed plus one clock cycle strategy. Commands are delivered on pins on falling edge of SDRCKP clock (command delayed by 180° shifting). Adds 2 clock cycle latencies.
11: command delayed strategy plus two clock cycles. Commands are delivered on pins on falling edge of SDRCKP clock (command delayed by 180° shifting). Adds 3 clock cycle latencies.

SRDPOL SDR-SDRAM read data capture polarity from HCLK edge. Unaffected by system (AHB) reset.
0: negative edge 1: positive edge (PORnot reset value)

SRDSTRA SDR-SDRAM read data strategy. Unaffected by system (AHB) reset.
00: clock delay strategy: SDRCKP must be delayed externally to at least meet the SDRAM input setup time (t_{AS}).
01: command delayed strategy: commands are delivered on pins on falling edge of SDCKP clock (command delayed by 180° shifting). Adds 1 clock cycle latency, but gives more relaxed timing, compatible for SDRCKP frequency > 100 MHz (reset value after PORnot reset).
10: command delayed plus one clock cycle strategy: commands are delivered on pins on falling edge of SDRCKP clock (command delayed by 180° shifting). Adds 2 clock cycle latencies, but gives more relaxed timing, compatible for SDRCKP frequency > 133 MHz.
11: command delayed strategy plus two clock cycles: commands are delivered on pins on falling edge of SDRCKP clock (command delayed by 180° shifting). Adds 3 clock cycle latencies, but gives more relaxed timing, compatible for SDRCKP frequency > 150 MHz.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SDMC_DYRP SDMC dynamic t_{RP} register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	TRP			
R	R	R	R	R	R	R	R	R	R	R	R	RW			

Address: SDMCBaseAddress + 0x030

Reset: 0x0000 000F

Description: Programs the precharge command period (t_{RP}). Should only be modified during system initialization, or when there are no current or outstanding transactions. This can be ensured by waiting until the SDMC is idle and then entering low-power or

disabled mode. This value can normally be found in the SDRAM datasheet as t_{RP}. It can be accessed with 1 wait state.

TRP Precharge command period.
0x0 to 0xE: N+1 HCLK clock cycles 0xF: 16 HCLK clock cycles (PORnot reset value)

0 Reserved for future use. Reading returns 0. Must be written with 0.

Note: This register is used for all dynamic memory chip selects. Therefore, the worst case value of all the chip selects must be programmed

SDMC_DYRAS SDMC dynamic t_{RAS} register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	TRAS			
R	R	R	R	R	R	R	R	R	R	R	R	RW			

Address: SDMCBaseAddress + 0x034

Reset: 0x0000 000F

Description: Programs the active to precharge command period (t_{RAS}). Should only be modified during system initialization. This value can normally be found in the SDRAM datasheet as t_{RAS}. It can be accessed with 1 wait state.

TRAS Active to precharge command period.
0x0 to 0xE: N+1 HCLK clock cycles 0xF: 16 HCLK clock cycles (PORnot reset value)

0 Reserved for future use. Reading returns 0. Must be written with 0.

Note: This register is used for all dynamic memory chip selects. Therefore, the worst case value of all the chip selects must be programmed.

SDMC_DYSREX SDMC dynamic t_{SREX} register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	TSREX						
R	R	R	R	R	R	R	R	R	RW						

Address: SDMCBaseAddress + 0x038

Reset: 0x0000 007F

Description: Programs the self-refresh exit time (t_{SREX}). Should only be modified during system initialization, or when there are no current or outstanding transactions. This value can

normally be found in the SDRAM datasheet as t_{SREX}. For devices without this parameter use the t_{XSR} value. It can be accessed with 1 wait state.

TSREX Self-refresh exit time.
0x0 to 0xE: N+1 HCLK clock cycles 0x7F: 128 HCLK clock cycles (PORnot reset value)

0 Reserved for future use. Reading returns 0. Must be written with 0.

Note: This register is used for all dynamic memory chip selects. Therefore, the worst case value of all the chip selects must be programmed.

SDMC_DYWR SDMC dynamic t_{WR} register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	TWR			
R	R	R	R	R	R	R	R	R	R	R	R	RW			

Address: SDMCBaseAddress + 0x044

Reset: 0x0000 000F

Description: Programs the write recovery time (t_{WR}). Should only be modified during system initialization, or when there are no current or outstanding transactions. This value can normally be found in the SDRAM datasheet as t_{WR}, t_{DPL}, t_{RWL}, or t_{RDL}. It can be accessed with 1 wait state.

TWR Last-data-out to active command time.
0x0 to 0xE: N+1 HCLK clock cycles 0xF: 16 HCLK clock cycles (PORnot reset value)

0 Reserved for future use. Reading returns 0. Must be written with 0.

Note: This register is used for all dynamic memory chip selects. Therefore, the worst case value of all the chip selects must be programmed.

SDMC_DYRC SDMC dynamic t_{RC} register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	TRC				
R	R	R	R	R	R	R	R	R	R	R	RW				

Address: SDMCBaseAddress + 0x048

Reset: 0x0000 001F

Description: Programs the auto-refresh and active command period (t_{RC}). It should only be modified during system initialization, or when there are no current or outstanding transactions. This can be ensured by waiting until the SDMC is idle and then entering low-power or disabled mode. This value can normally be found in the SDRAM datasheet as t_{RC}. It can be accessed with 1 wait state.

TRC Auto-refresh and active command period.
0x0 to 0x1E: N+1 HCLK clock cycles 0x1F: 32 HCLK clock cycles (PORnot reset value)

0 Reserved for future use. Reading returns 0. Must be written with 0.

Note: This register is used for all dynamic memory chip selects. Therefore, the worst case value of all the chip selects must be programmed.

SDMC_DYRFC SDMC dynamic t_{RFC} register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	TRFC				
R	R	R	R	R	R	R	R	R	R	R	RW				

Address: SDMCBaseAddress + 0x04C

Reset: 0x0000 001F

Description: Programs the auto-refresh period and auto-refresh to active command period (t_{RFC}). It should only be modified during system initialization, or when there are no current or outstanding transactions. This can be ensured by waiting until the SDMC is idle and then entering low-power or disabled mode. This value can normally be found in the SDRAM datasheet as t_{RFC}, or sometimes t_{RC}. It can be accessed with 1 wait state.

TRFC Auto-refresh period and auto-refresh to active command period.
0x0 to 0x1E: N+1 HCLK clock cycles 0x1F: 32 HCLK clock cycles (PORnot reset value)

0 Reserved for future use. Reading returns 0. Must be written with 0.

Note: This register is used for all dynamic memory chip selects. Therefore, the worst case value of all the chip selects must be programmed.

SDMC_DYXSR — SDMC dynamic t_{XSR} register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	TXSR							
R	R	R	R	R	R	R	R	RW							

Address: SDMCBaseAddress + 0x050

Reset: 0x0000 001F

Description: Programs the exit self-refresh to active command time (t_{XSR}). It should only be modified during system initialization, or when there are no current or outstanding transactions. This can be ensured by waiting until the SDMC is idle and then entering low-power or disabled mode. This value can normally be found in the SDRAM datasheet as t_{XSR}. It can be accessed with 1 wait state.

TXSR Auto-refresh and active command period.
0x0 to 0x1E: N+1 HCLK clock cycles 0xFF:256 HCLK clock cycles (PORnot reset value)

0 Reserved for future use. Reading returns 0. Must be written with 0.

Note: This register is used for all dynamic memory chip selects. Therefore, the worst case value of all the chip selects must be programmed.

SDMC_DYRRD — SDMC dynamic t_{RRD} register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	TRRD			
R	R	R	R	R	R	R	R	R	R	R	R	RW			

Address: SDMCBaseAddress + 0x054

Reset: 0x0000 000F

Description: Programs the active bank A to active bank B latency (t_{RRD}). It should only be modified during system initialization, or when there are no current or outstanding transactions. This can be ensured by waiting until the SDMC is idle and then entering low-power or disabled mode. This value can normally be found in the SDRAM datasheet as t_{RRD}. It can be accessed with 1 wait state.

TRRD Active bank A to active bank B latency.
0x0 to 0xE: N+1 HCLK clock cycles 0xF: 16 HCLK clock cycles (PORnot reset value)

0 Reserved for future use. Reading returns 0. Must be written with 0.

Note: This register is used for all dynamic memory chip selects. Therefore, the worst case value of all the chip selects must be programmed.

SDMC_DYMRD — SDMC dynamic t_{MRD} register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	TMRD			
R	R	R	R	R	R	R	R	R	R	R	R	RW			

Address: SDMCBaseAddress + 0x058

Reset: 0x0000 000F

Description: Programs the load mode register to active command time (t_{MRD}). It should only be modified during system initialization, or when there are no current or outstanding transactions. This can be ensured by waiting until the SDMC is idle and then entering low-power or disabled mode.This value can normally be found in the SDRAM datasheet as t_{MRD}, or t_{RSA}. It can be accessed with 1 wait state.

TMRD Load mode register to active command time.
0x0 to 0xE: N+1 HCLK clock cycles 0xF: 16 HCLK clock cycles (PORnot reset value)

0 Reserved for future use. Reading returns 0. Must be written with 0.

Note: This register is used for all dynamic memory chip selects. Therefore, the worst case value of all the chip selects must be programmed.

SDMC_DYCDLR — SDMC dynamic t_{CDLR} register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	TCDLR			
R	R	R	R	R	R	R	R	R	R	R	R	RW			

Address: SDMCBaseAddress + 0x058

Reset: 0x0000 000F

Description: Programs the last data in to read command time (t_{CDLR}). It should only be modified at system initialization, or when there are no current or outstanding transactions. This can be ensured by waiting until the SDMC is idle and then entering low-power or

disabled mode. This value can normally be found in the SDRAM datasheet as t_{CDLR}. It can be accessed with 1 wait state.

TCDLR Last data in to read command time.
0x0 to 0xE: N+1 HCLK clock cycles 0xF: 16 HCLK clock cycles (PORnot reset value)

0 Reserved for future use. Reading returns 0. Must be written with 0.

Note: This register is used for all dynamic memory chip selects. Therefore, the chip selects worst case must be programmed.

SDMC_DYCFGx SDMC dynamic config register x (x = 0:1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	WPROT	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	RW	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	AM								0	0	0	0	MD		
R	RW								R	R	R	R	RW		

Address: SDMCBaseAddress + 0x100 + 0x20 * x (x = 0 to 1)

Reset: 0x0000 1486

Description: Configures the SDRAM chip select. It should only be modified at system initialization and can be accessed with 1 wait state.

WPROT Write protect.

0: not protected (PORnot reset value) 1: protected

AM Address mapping for chip select. Defines the SDRAM column and row width and number of banks.

16-bit external bus high-performance mapping (row-bank-column):

0000 0000: 16Mb (2Mx8), 2 banks, row length = 11, column length = 9 (PORnot reset value).
0000 0001: 16Mb (1Mx16), 2 banks, row length = 11, column length = 8.
0000 0100: 64Mb (8Mx8), 4 banks, row length = 12, column length = 9.
0000 0101: 64Mb (4Mx16), 4 banks, row length = 12, column length = 8.
0000 1000: 128Mb (16Mx8), 4 banks, row length = 12, column length = 10.
0000 1001: 128Mb (8Mx16), 4 banks, row length = 12, column length = 9.
0000 1100: 256Mb (32Mx8), 4 banks, row length = 13, column length = 10.
0000 1101: 256Mb (16Mx16), 4 banks, row length = 13, column length = 9.
0001 0000: 512Mb (64Mx8), 4 banks, row length = 13, column length = 11.
0001 0001: 512Mb (32Mx16), 4 banks, row length = 13, column length = 10.

16-bit external bus low-power SDRAM mapping (bank-row-column):

0010 0000: 16Mb (2Mx8), 2 banks, row length = 11, column length = 9.
0010 0001: 16Mb (1Mx16), 2 banks, row length = 11, column length = 8.
0010 0100: 64Mb (8Mx8), 4 banks, row length = 12, column length = 9.
0010 0101: 64Mb (4Mx16), 4 banks, row length = 12, column length = 8.
0010 1000: 128Mb (16Mx8), 4 banks, row length = 12, column length = 10.
0010 1001: 128Mb (8Mx16), 4 banks, row length = 12, column length = 9 (PORnot reset value).
0010 1100: 256Mb (32Mx8), 4 banks, row length = 13, column length = 10.
0010 1101: 256Mb (16Mx16), 4 banks, row length = 13, column length = 9.
0011 0000: 512Mb (64Mx8), 4 banks, row length = 13, column length = 11.
0011 0001: 512Mb (32Mx16), 4 banks, row length = 13, column length = 10.

Note: values not shown above are reserved.

MD Memory device.

000: SDR-SDRAM 001: SDR Micron SyncFlash
010: LowPower SDR-SDRAM 011: reserved
100: DDR-SDRAM 101: DDR Micron SyncFlash
110: LowPower DDR-SDRAM (PORnot reset value) 111: reserved

0 Reserved for future use. Reading returns 0. Must be written with 0.

SDMC_DYRASCASx SDMC dynamic RAS/CAS register x (x = 0:1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	CAS				0	0	0	RAS			
R	R	R	R	R	RW				R	R	R	RW			

Address: SDMCBaseAddress + 0x104 + 0x20 * x (x = 0 to 1)

Reset: 0x0000 0783

Description: Programs RAS and CAS latencies for the relevant dynamic memory. It should only be modified during system initialization. It can be accessed with 1 wait state.

CAS CAS latency.

0000: reserved	0001: reserved
0010: reserved	0011: reserved
0100: 1 SDRCKP clock cycles	0101: reserved
0110: 2 SDRCKP clock cycles	0111: reserved
1000: 3 SDRCKP clock cycles	1001: reserved
1010: 4 SDRCKP clock cycles	1011: reserved
1100: 5 SDRCKP clock cycles	1101: reserved
1110: 6 SDRCKP clock cycles	1111: reserved (PORnot reset value)

RAS RAS latency (active to read or write delay). Normally found in the SDRAM datasheet as t_{RCD}.

0000: reserved	0001 to 1110: n SDRCKP clock cycles
1111: 15 SDRCKP clock cycles	0011: 3 SDRCKP clock cycles (PORnot reset value)

0 Reserved for future use. Reading returns 0. Must be written with 0.

SDMC_IBCRx SDMC internal bus x control register (x = 0:5)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BUF EN
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Address: SDMCBaseAddress + 0x400 + x * 0x20 (x = 0 to 5)

Reset: 0x0000 0000

Description: Controls the internal AHB bus interfaces operation. These register can be re-programmed during normal operation.

BUFEN Internal AHB bus x buffer enable. The buffer is only used if it is enabled for the internal (AHB) bus, or if for writes the AHB HPROT[2] (bufferable) bit is high for the transfer, or if for reads the AHB HPROT[3] (cachable) bit is high for the transfer.

0: disable 1: enable

0 Reserved for future use. Reading returns 0. Must be written with 0.

SDMC_IBSRx SDMC internal bus x status register (x = 0:5)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BUF STA
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: SDMCBaseAddress + 0x404 + x * 0x20 (x = 0 to 5)

Reset: 0x0000 0000

Description: Provides internal AHB bus interfaces status information.

BUFSTA Internal AHB bus x buffer status.
0: empty 1: contains data

0 Reserved for future use. Reading returns 0. Must be written with 0.

SDMC_IBTORx SDMC internal bus x timeout register (x = 0:5)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	EVTIME OUT	IBTIMEOUT									
R	R	R	R	R	RW	RW									

Address: SDMCBaseAddress + 0x404 + x * 0x20 (x = 0 to 5)

Reset: 0x0000 0000

Description: Ensures that each AHB port is serviced within a programmed number of cycles. When an AHB request goes active on SDMC port x, the value in the SDMC_IBTORx register is loaded into a timeout down counter. If the transfer does not get processed

by the time the timeout counter has reached zero, the priority of the AHB port x is increased until the request is serviced.

These registers therefore enable transaction latency and, indirectly, the bandwidth, for a particular port to be defined. The value programmed into these registers depends on the latency required for the particular internal (AHB) port.

These registers can be reprogrammed during normal operation.

EVTIMEOUT AHB port x timeout event. Enables the external event that forces timeout. Only significant for AHB ports 4 and 5. External event means an interrupt (IRQ or FIQ). External event is grounded for other ports.

0: disabled (PORnot reset value) 1: enabled

IBTIMEOUT AHB port x timeout. Indicates the number of AHB clock cycles before timeout.

0: disabled (PORnot reset value) 1 to 1023: number of AHB clock cycles

0 Reserved for future use. Reading returns 0. Must be written with 0.

SDMCPeriphID4 SDMC peripheral identification register 4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	AHBPortNumber			
R	R	R	R	R	R	R	R	R	R	R	R	R			

Address: SDMCBaseAddress + 0xFD0

Reset: 0x0000 0005

Description: AHBPortNumber: 0101 = 6 AHB memory ports.

SDMCPeriphID[5:7] SDMC peripheral identification registers[5:7]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: SDMCBaseAddress + 0xFD4 (+ 0xFD8 and + 0xFDC)

Reset: 0x0000 0000

SDMCPeriphID0 SDMC peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: SDMCBaseAddress + 0xFE0

Reset: 0x0000 0075

Description: PartNumber0 returns 0x75.

SDMCPeriphID1 SDMC peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: SDMCBaseAddress + 0xFE4

Reset: 0x0000 0011

Description: Designer0 returns 0x1. PartNumber1 returns 0x1.

SDMCPeriphID2 SDMC peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: SDMCBaseAddress + 0xFE8

Reset: 0x0000 0014

Description: Revision returns the peripheral revision (0x1). Designer1 returns 0x4.

SDMCPeriphID3 SDMC peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: SDMCBaseAddress + 0xFEC

Reset: 0x0000 0007

Description: Configuration returns 0x07.

SDMCPCellID0 SDMC PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SDMCPCellID0							
R	R	R	R	R	R	R	R	R							

Address: SDMCBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: SDMCPCellID0 returns 0x0D.

SDMCPCellID1 SDMC PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SDMCPCellID1							
R	R	R	R	R	R	R	R	R							

Address: SDMCBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: SDMCPCellID1 returns 0xF0.

SDMCPCellID2 SDMC PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SDMCPCellID2							
R	R	R	R	R	R	R	R	R							

Address: SDMCBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: SDMCPCellID2 returns 0x05.

SDMCPCellID3 SDMC PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SDMCPCellID3							
R	R	R	R	R	R	R	R	R							

Address: SDMCBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: SDMCPCellID3 returns 0xB1.

17 Flexible static-memory controller (FSMC0,1,2,3)

17.1 FSMC register addressing

Register addresses are provided as the flexible static memory controller base address, FSMCnBaseAddress, plus the register offset.

There are 4 FSMC base addresses.

Table 31. FSMC base addresses

FSMC base address	NOR-Flash chip select
0x3C00 0000	3
0x3800 0000	2
0x3400 0000	1
0x3000 0000	0

FSMC3,2,1 and 0 are referred to as FSMCn throughout this document.

17.2 FSMC register summary

Table 32. FSMC register list

Offset	Register	Description	Page
0x000	FSMCn_BCR0	FSMC SRAM/NOR-Flash chip-select control register 0	*182*
0x004	FSMCn_BTR0	FSMC SRAM/NOR-Flash chip-select timing register 0	*183*
0x008	FSMCn_BCR1	FSMC SRAM/NOR-Flash chip-select control register 1	*182*
0x00C	FSMCn_BTR1	FSMC SRAM/NOR-Flash chip-select timing register 1	*183*
0x010	FSMCn_BCR2	FSMC SRAM/NOR-Flash chip-select control register 2	*182*
0x014	FSMCn_BTR2	FSMC SRAM/NOR-Flash chip-select timing register 2	*183*
0x018	FSMCn_BCR3	Reserved	
0x01C	FSMCn_BTR3	Reserved	
0x040	FSMCn_PCR0	FSMC PC-card/NAND-Flash control register 0	*184*
0x048	FSMCn_PMEM0	FSMC common memory space timing register 0	*185*
0x04C	FSMCn_PATT0	FSMC attribute memory space timing register 0	*186*
0x050	FSMCn_PIO0	FSMC I/O space timing register 0	*187*
0x054	FSMCn_ECCR0	FSMC ECC result register 0	*188*
0x060	FSMCn_PCR1	FSMC PC-card/NAND-Flash control register 1	*184*
0x068	FSMCn_PMEM1	FSMC common memory space timing register 1	*185*
0x06C	FSMCn_PATT1	FSMC attribute memory space timing register 1	*186*
0x070	FSMCn_PIO1	FSMC I/O space timing register 1	*187*
0x074	FSMCn_ECCR1	FSMC ECC result register 1	*188*

Table 32. FSMC register list (continued)

Offset	Register	Description	Page
0x0C4	FSMCn_ITIP0	FSMC boot strap vector logic register	189
0xFE0	FSMCn_PeriphID0	FSMC peripheral identification register 0 (bits 7:0)	189
0xFE4	FSMCn_PeriphID1	FSMC peripheral identification register 1 (bits 15:8)	189
0xFE8	FSMCn_PeriphID2	FSMC peripheral identification register 2 (bits 23:16)	190
0xFEC	FSMCn_PeriphID3	FSMC peripheral identification register 3 (bits 31:24)	190
0xFF0	FSMCn_PCellID0	FSMC PCell identification register 0 (bits 7:0)	190
0xFF4	FSMCn_PCellID1	FSMC PCell identification register 1 (bits 15:8)	191
0xFF8	FSMCn_PCellID2	FSMC PCell identification register 2 (bits 23:16)	191
0xFFC	FSMCn_PCellID3	FSMC PCell identification register 3 (bits 31:24)	191

17.3 FSMC register descriptions

FSMCn_BCRx — FSMC SRAM/NOR-Flash chip-select control register x (x = 0:3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	WAITEN	WREN	WAITCFG	WRAPMOD	WAITPOL	BURSTEN	FWPRLVL	FRSTLVL	MWID		MTYP		MUXEN	MBKEN
R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW		RW	RW

Address: FSMCnBaseAddress + 0x000 + 8 * x (x = 0 to 3)

Reset: 0x0000 30XX.

Description: Contains chip-select control information, used for SRAMs, ROMs and asynchronous or burst NOR-Flash memories. A NOR-Flash memory in reset-power-down (bit FRSTLVL = 1) is considered disabled, even if bit MBKEN = 1. A memory access in such conditions generates an AHB ERROR response, as if the chip-select were disabled. (Only 3 chip selects are available on the FSMC interface so FSMC_BCR3 is no longer used).

WAITEN Wait enable. Controls the number of wait states inserted (determined by the SMWAITnot signal value) after the programmed Flash latency period for Flash memory access in burst mode.
0: disabled (no wait-states inserted) 1: enabled (default after reset)

WREN Write enable. Controls write operations to be accepted in the bank by the FSMC. When enabled, a write may still be rejected by memory (see bit FWPRLVL).
0: disabled, an AHB error is reported 1: enabled (default after reset)

WAITCFG Wait timing configuration. Determines if the SMWAITnot signal is asserted by the Flash memory one clock cycle before the wait state, or during the wait state for Flash memory access in burst mode.
0: active 1 data cycle before (default after reset) 1: active during wait state

WRAPMOD Wrapped burst mode support. Defines whether or not the controller splits an AHB burst wrap access into two linear accesses. Valid only when accessing Flash memories in burst mode.
0: disabled (default after reset) 1: enabled

WAITPOL Wait signal polarity bit. Defines the polarity of the wait signal from Flash memory. Valid only when accessing Flash memory in burst mode.
0: SMWAITnot is active low (default after reset) 1: SMWAITnot is active high

BURSTEN Burst enable. Sets the Flash memory burst access mode for synchronous burst Flash memories.
0: disabled (default after reset) 1: enabled

FWPRLVL Flash write protection pin level. Defines the write protection signal level (SMFWPnot) for Flash memory connected on chip-selects 0 to 3. Only valid with Flash-type memory.
0: low (default after reset) 1: high

FRSTLVL Flash reset pin level. Defines the reset signal level (SMFRSTnot) for Flash memories connected on chip-selects 0 to 3. Only valid with Flash-type memories.
0: reset signal low 1: reset signal high (default after reset)

MWID Memory data bus width. Defines the external memory device width. Valid for all memory types. Value for chip-select 0 after reset depends on level of internal signals DefDevW[1:0] during reset (in STn8815, DefDevW[1] is grounded and DefDevW[0] is REMAP[1]).
00: 8 bits 01: 16 bits
10: reserved for 32 bits, do not use 11: reserved, do not use

MTYP Memory type. Defines the type of external memory attached to the corresponding chip-select.
00: SRAM, ROM (default after reset for chip-select 1 to 3) 01: reserved
10: NOR Flash (default after reset for chip-select 0) 11: reserved

MUXEN Address/data multiplexing enable bit. When set, the address and data values are multiplexed on the SMADQx signals. Valid only with NOR Flash memory.
0: non-multiplexed 1: multiplexed (default after reset)

MBKEN Memory chip-select enable. Enables the chip-select. After reset, chip-select 0 is enabled, and all others are disabled. When a disabled chip-select is accessed, an HRESP=ERROR is generated on the AHB.
0: disabled 1: enabled

0 Reserved for future use. Reading returns 0. Must be written with 0.

FSMCn_BTRx FSMC SRAM/NOR-Flash chip-select timing register x (x = 0:3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	DATLAT				CLKDIV				BUSTURN			
R	R	R	R	RW				RW				RW			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAST								ADDHLD				ADDSET			
RW								RW				RW			

Address: FSMCnBaseAddress + 0x004 + 8 * x (x = 0 to 3)

Reset: 0x0FFF FFFF

Description: Contains control information for each chip-select, used for SRAMs, ROMs and NOR-Flash memories. (Only 3 chip selects are available on the FSMC interface so FSMC_BTR3 is no longer used).

DATLAT Data latency (for synchronous burst NOR Flash) expressed in Flash clock (SMCKO) periods. Defines the number of memory clock cycles (+2) to issue to the memory before obtaining the first data. Not relevant for asynchronous NOR-Flash, SRAM or ROM accesses.
0x0: 2 SMCKO periods through to 0xF: 17 SMCKO periods (default value after reset).

CLKDIV Clock divide ratio (for SMCKO signal). Defines the SMCKO clock output signal period, expressed in number of HCLK cycles. Not relevant in asynchronous NOR-Flash, SRAM or ROM accesses.
0x0: HCLK period through to 0xF: 16 * HCLK period (default value after reset).

BUSTURN Bus turn-around phase duration used for all memory types when the current chip-select is no longer addressed on the next cycle.
0x0: 1 * HCLK cycle through to 0xF: 16 * HCLK cycles (default value after reset).

DATAST Data phase duration valid for all memory types.
0x0: 1 * HCLK cycle through to 0xF: 16 * HCLK cycles (default value after reset).

ADDHLD Address hold phase duration used in SRAMs, ROMs and asynchronous muxed NOR-Flash. In synchronous NOR-Flash accesses, this value is always 1 Flash clock period duration.
0x0: 1 * HCLK cycle through to 0xF: 16 * HCLK cycles (default value after reset).

ADDSET Address setup phase duration used in SRAMs, ROMs and asynchronous muxed NOR-Flash. In synchronous NOR-Flash accesses, this value is always 1 Flash clock period duration.
0x0: 1 * HCLK cycle through to 0xF: 16 * HCLK cycles (default value after reset).

0 Reserved for future use. Reading returns 0. Must be written with 0.

FSMCn_PCRx FSMC PC-card/NAND-Flash control register x (x = 0:1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	AD LOW	ECCPL EN	ECC EN	PWID		PTYP	PBK EN	PWAIT EN	PRST
R	R	R	R	R	R	R	RW	RW	RW	RW		RW	RW	RW	RW

Address: FSMCnBaseAddress + 0x040 + 0x20 * x (x = 0 to 1)

Reset: 0x0000 0008

Description:

ADLOW Address low bit delivery. Defines which PC-card/NAND-Flash controller address bits are delivered on the SMAD[24:16] signals: direct mapping or low address bit mapping.
0: AHB address lines [24:16] 1: AHB address lines [8:0]

ECCPLEN ECC page length. Defines the page length of the NAND-Flash memory device used by ECC computation logic.
0: 512 bytes (PWID = 0) (default after reset)
1: 256 bytes (PWID = 0) or 128 half-words (PWID = 1)

ECCEN ECC computation logic enable.
0: logic disabled and reset (default after reset) 1: logic enabled

PWID NAND-Flash data bus width. Defines the external NAND-Flash memory device width. Valid only if PTYP is NAND-Flash.
00: 8 bits (default after reset) 01: 16 bits
10: reserved for 32 bits (do not use) 11: reserved, do not use

PTYP Memory type. Defines the type of device attached to the corresponding chip-select.
0: PC-card, CompactFlash, CF+ or PCMCIA 1: NAND-Flash (default after reset)

PBKEN PC-card/NAND-Flash chip-select enable. Enables the corresponding chip-select.
If a disabled chip-select is accessed, an HRESP = ERROR is generated on the AHB bus.
0: disabled (default after reset) 1: enabled

PWAITEN Wait feature enable. Enables the wait feature for the PC-card/NAND-Flash chip-select.
0: disabled 1: enabled

PRST PC-card reset pin level. Defines the level of the PC-card reset signal (SMPRSTxn).
0: low 1: high (default after reset)

0 Reserved for future use. Reading returns 0. Must be written with 0.

FSMCn_PMEMx FSMC common memory space timing register x (x = 0:1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MEMxHIZ								MEMxHOLD							
RW								RW							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MEMxWAIT								MEMxSET							
RW								RW							

Address: FSMCnBaseAddress + 0x048 + 0x20* x (x = 0 to 1)

Reset: 0xFCFC FCFC

Description: Contains timing information for PC-card x or NAND-Flash chip-select x (where x = 0 or 1) used to access the common memory space of the 16-bit PC-card/compact-

Flash, or to access the NAND-Flash for commands, address writes and data read/write accesses.

MEM[0:1] HIZ — Common memory x data bus HiZ time. Defines the number of HCLK clock cycles during which the data bus is kept in HiZ after the start of a PC-card/NAND-Flash write access to common memory space on socket x. Only valid for write transactions.

0x00: 0 HCLK cycle through to 0xFF: 255 HCLK cycles (default value 0xFC after reset).

MEM[0:1] HOLD — Common memory x hold time. Defines the number of HCLK clock cycles to hold an address (and data for write access) after the command deassertion (SMWEnot, SMOEnot) for PC-card/NAND-Flash read/write access to common memory space on socket x.

0x00: reserved, do not use this value.

0x01: 1 HCLK cycle through to 0xFF: 255 HCLK cycles (default value 0xFC after reset).

MEM[0:1] WAIT — Common memory x wait time. Defines the minimum number of HCLK (+1) clock cycles to assert the command (SMWEnot, SMOEnot), for PC-card/NAND-Flash read or write access to common memory space on socket x. The duration of command assertion is extended if the wait signal (SMWAITnot) is active (low) at the end of the programmed HCLK value.

0x00: 1 HCLK cycle (+ wait cycle introduced by deassertion of SMWAITnot) through to

0xFF: 256 HCLK cycles (+ wait cycle introduced by card deassertion of SMWAITnot) (default value 0xFC after reset).

MEM[0:1] SET — Common memory x setup time. Defines the number of HCLK (+1) clock cycles to set-up the address before the command assertion (SMWEnot, SMOEnot), for PC-card/NAND-Flash read/write access to common memory space on socket x.

0x00: 1 HCLK cycle through to 0xFF: 256 HCLK cycles (default value 0xFC after reset).

0 — Reserved for future use. Reading returns 0. Must be written with 0.

FSMCn_PATTx — FSMC attribute memory space timing register x (x = 0:1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ATTxHIZ								ATTxHOLD							
RW								RW							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATTxWAIT								ATTxSET							
RW								RW							

Address: FSMCnBaseAddress + 0x04C + 0x20* x (x = 0 to 1)

Reset: 0xFCFC FCFC

Description: Contains timing information for the PC-card x or NAND-Flash chip-select x (where x = 0 or 1), used to access the attribute memory space of the 16-bit PC-card/compact-

Flash, or to access the NAND-Flash for last address writes if timing must differ from the other accesses.

ATT[0:1]HIZ Attribute memory x data bus HiZ time. Defines the number of HCLK clock cycles during which the data bus is kept in HiZ after the start of a PC-card/NAND-Flash write access to attribute memory space on socket x. Only valid for write transactions.

0x00: 0 HCLK cycle through to 0xFF: 255 HCLK cycles (default value 0xFC after reset).

ATT[0:1] HOLD Attribute memory x hold time. Defines the number of HCLK (+1) clock cycles to hold the address (and data for write access) after the command deassertion (SMWEnot, SMOEnot), for PC-card/NAND-Flash read or write access to attribute memory space on socket x.

0x00: 1 HCLK cycle through to 0xFF: 256 HCLK cycles (default value 0xFC after reset).

ATT[0:1] WAIT Attribute memory x wait time. Defines the minimum number of HCLK (+1) clock cycles to assert the command (SMWEnot, SMOEnot), for PC-card/NAND-Flash read or write access to attribute memory space on socket x. The duration of command assertion is extended if the wait signal (SMWAITnot) is active (low) at the end of the programmed HCLK value.

0x00: 1 HCLK cycle (+ wait cycle introduced by deassertion of SMWAITnot) through to

0xFF: 256 HCLK cycles (+ wait cycle introduced by the card deasserting SMWAITnot) (default value 0xFC after reset).

ATT[0:1] SET Attribute memory x setup time. Defines the number of HCLK (+1) clock cycles to set-up the address before the command assertion (SMWEnot, SMOEnot), for PC-card/NAND-Flash read or write access to attribute memory space on socket x.

0x00: 1 HCLK cycle through to 0xFF: 256 HCLK cycles (default value 0xFC after reset).

0 Reserved for future use. Reading returns 0. Must be written with 0.

FSMCn_PIOx FSMC I/O space timing register x (x = 0:1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IOxHIZ								IOxHOLD							
RW								RW							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOxWAIT								IOxSET							
RW								RW							

Address: FSMCnBaseAddress + 0x050 + 0x20* x (x = 0 to 1)

Reset: 0xFCFC FCFC

Description: Contains the timing information for PC-card x or NAND-Flash chip-select [0:1], used for access to the I/O space of the 16-bit PC-card/Compact-Flash.

IO[0:1]HIZ I/O x data bus HiZ time. Defines the number of HCLK clock cycles during which the data bus is kept in HiZ after the start of a PC-card/NAND-Flash write access to I/O space on socket x. Only valid for write transactions.

0x00: 0 HCLK cycle through to 0xFF: 255 HCLK cycles (default value 0xFC after reset).

IO[0:1]HOLD I/O x hold time. Defines the number of HCLK (+1) clock cycles to hold the address (and data for write access) after the command deassertion (SMWEnot, SMOEnot) for PC-card/NAND-Flash read or write access to I/O space on socket x.

0x00: 1 HCLK cycle through to 0xFF: 256 HCLK cycles (default value 0xFC after reset).

IO[0:1]WAIT I/O x wait time. Defines the minimum number of HCLK (+1) clock cycles to assert the command (SMWEnot, SMOEnot), for PC-card/NAND-Flash read or write access to I/O space on socket x. The command assertion duration is extended if the wait signal (SMWAITnot) is active (low) at the end of the programmed value of HCLK.

0x00: 1 HCLK cycle (+ wait cycle introduced by deassertion of SMWAITnot) through to

0xFF: 256 HCLK cycles (+ wait cycle introduced by the card deasserting SMWAITnot) (default value 0xFC after reset).

IO[0:1]SET I/O x setup time. Defines the number of HCLK (+1) clock cycles to set-up the address before the command assertion (SMWEnot, SMOEnot) for PC-card/NAND-Flash read or write access to I/O space on socket x.

0x00: 1 HCLK cycle through to 0xFF: 256 HCLK cycles (default value 0xFC after reset).

0 Reserved for future use. Reading returns 0. Must be written with 0.

FSMCn_ECCRx FSMC ECC result register x (x = 0:1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ECC3							
R	R	R	R	R	R	R	R	R							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC2								ECC1							
R								R							

Address: FSMCnBaseAddress + 0x054 + 0x20* x (x = 0 to 1)

Reset: 0x00FF FFFF

Description: Contains the error correction code computed by the ECC computation modules of the FSMC (one module per NAND-Flash chip-select). When the CPU reads a page from NAND-Flash or a SmartMedia component, the data (256 or 512 bytes, depending on *FSMCn_PCRx*.ECCPLEN) read from or written to the NAND-Flash is processed automatically by the ECC computation module. After reading, the CPU judges the validity of a page by comparing the computed parity data ECC value from register FSMCn_ECCRx with the parity value recorded in work space, and corrects it if necessary.

ECC1 to 3 These fields provide the value computed by the ECC computation logic for 256 x 8, 256 x 16 or 512 x 8 data read from a NAND-Flash page. This register is cleared automatically once read.

0 Reserved for future use. Reading returns 0. Must be written with 0.

Table 33. ECC parity codes for the various NAND-Flash organizations

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ECC parity code for 256 x 8 NAND-Flash								
ECC1	P64	P64’	P32	P32’	P16	P16’	P8	P8’
ECC2	P1024	P1024’	P512	P512’	P256	P256’	P128	P128’
ECC3	P4	P4’	P2	P2’	P1	P1’	0	0
ECC parity code for 256 x 16 NAND-Flash and 512 x 8 NAND-Flash								
ECC1	P64	P64’	P32	P32’	P16	P16’	P8	P8’
ECC2	P1024	P1024’	P512	P512’	P256	P256’	P128	P128’
ECC3	P4	P4’	P2	P2’	P1	P1’	P2048	P2048’

FSMCn_PeriphID0 FSMC peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: FSMCnBaseAddress + 0xFE0

Reset: 0x0000 0090

Description: PartNumber0 returns 0x90.

FSMCn_PeriphID1 FSMC peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: FSMCnBaseAddress + 0xFE4

Reset: 0x0000 0000

Description: Designer0 returns 0x0. PartNumber1 returns 0x0.

FSMCn_PeriphID2 FSMC peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: FSMCnBaseAddress + 0xFE8.

Reset: 0x0000 0008

Description: Revision returns the peripheral revision. Designer1 returns 0x8.

FSMCn_PeriphID3 FSMC peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: FSMCnBaseAddress + 0xFEC

Reset: 0x0000 0000

Description: Configuration returns 0x00

FSMCn_PCellID0 FSMC PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	FSMCPCellID0							
R	R	R	R	R	R	R	R	R							

Address: FSMCnBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: FSMCPCellID0 returns 0x0D.

FSMCn_PCellID1 FSMC PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	FSMCPCellID1							
R	R	R	R	R	R	R	R	R							

Address: FSMCnBaseAddress + 0xFF4.

Reset: 0x0000 00F0

Description: FSMCPCellID1 returns 0xF0.

FSMCn_PCellID2 FSMC PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	FSMCPCellID2							
R	R	R	R	R	R	R	R	R							

Address: FSMCnBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: FSMCPCellID2 returns 0x05.

FSMCn_PCellID3 FSMC PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	FSMCPCellID3							
R	R	R	R	R	R	R	R	R							

Address: FSMCnBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: FSMCPCellID3 returns 0xB1.

18 DMA controllers (DMAC0,1)

18.1 DMAC register addressing

Register addresses are provided as the DMAC base address, DMACnBaseAddress, and register offset.

There are 2 DMAC base addresses.

Table 34. DMAC base addresses

DMAC base address	DMAC
0x1013 0000	0
0x1015 0000	1

DMAC 0 and 1 are referred to as DMACn throughout this document.

18.2 DMAC register summary

Reserved or unused address locations must not be accessed as this can result in unpredictable behavior of the device.
Reserved or unused bits of registers must be written as zero, and ignored on read, unless otherwise stated in the relevant text.

All registers bits are reset to a logic 0 by the system or power on reset value, unless otherwise stated in the relevant text.
All registers support read and write accesses, unless otherwise stated in the relevant text. A write updates the contents of a register and a read returns the contents of the register.

All registers defined in this document can only be accessed using word reads and word writes, unless otherwise stated in the relevant text.

Table 35. DMAC register list

Offset	Register	Description	Page
0x000	DMACn_MIS	DMAC masked interrupt status register	*194*
0x004	DMACn_TCMIS	DMAC terminal count masked interrupt status register	*195*
0x008	DMACn_TCICR	DMAC terminal count interrupt clear register	*195*
0x00C	DMACn_EMIS	DMAC error masked interrupt status register	*196*
0x010	DMACn_EICR	DMAC error interrupt clear register	*196*
0x014	DMACn_TCRIS	DMAC terminal count raw interrupt register	*197*
0x018	DMACn_ERIS	DMAC error raw interrupt register	*197*
0x01C	DMACn_ECHSR	DMAC enabled channel status register	*198*
0x020	DMACn_SBREQ	DMAC status burst request register	*198*
0x024	DMACn_SSREQ	DMAC status single request register	*199*
0x028	DMACn_SLBREQ	DMAC status last burst request register	*199*
0x02C	DMACn_SLSREQ	DMAC status last single request register	*200*

Table 35. DMAC register list (continued)

Offset	Register	Description	Page
0x100	DMACn_C0SADR	DMAC channel 0 source address	200
0x104	DMACn_C0DADR	DMAC channel 0 destination address	201
0x108	DMACn_C0LLI	DMAC channel 0 linked list item	201
0x10C	DMACn_C0CR	DMAC channel 0 control	202
0x110	DMACn_C0CFG	DMAC channel 0 configuration	204
0x120	DMACn_C1SADR	DMAC channel 1 source address	200
0x124	DMACn_C1DADR	DMAC channel 1 destination address	201
0x128	DMACn_C1LLI	DMAC channel 1 linked list item	201
0x12C	DMACn_C1CR	DMAC channel 1 control	202
0x130	DMACn_C1CFG	DMAC channel 1 configuration	204
0x140	DMACn_C2SADR	DMAC channel 2 source address	200
0x144	DMACn_C2DADR	DMAC channel 2 destination address	201
0x148	DMACn_C2LLI	DMAC channel 2 linked list item	201
0x14C	DMACn_C2CR	DMAC channel 2 control	202
0x150	DMACn_C2CFG	DMAC channel 2 configuration	204
0x160	DMACn_C3SADR	DMAC channel 3 source address	200
0x164	DMACn_C3DADR	DMAC channel 3 destination address	201
0x168	DMACn_C3LLI	DMAC channel 3 linked list item	201
0x16C	DMACn_C3CR	DMAC channel 3 control	202
0x170	DMACn_C3CFG	DMAC channel 3 configuration	204
0x180	DMACn_C4SADR	DMAC channel 4 source address	200
0x184	DMACn_C4DADR	DMAC channel 4 destination address	201
0x188	DMACn_C4LLI	DMAC channel 4 linked list item	201
0x18C	DMACn_C4CR	DMAC channel 4 control	202
0x190	DMACn_C4CFG	DMAC channel 4 configuration	204
0x1A0	DMACn_C5SADR	DMAC channel 5 source address	200
0x1A4	DMACn_C5DADR	DMAC channel 5 destination address	201
0x1A8	DMACn_C5LLI	DMAC channel 5 linked list item	201
0x1AC	DMACn_C5CR	DMAC channel 5 control	202
0x1B0	DMACn_C5CFG	DMAC channel 5 configuration	204
0x1C0	DMACn_C6SADR	DMAC channel 6 source address	200
0x1C4	DMACn_C6DADR	DMAC channel 6 destination address	201
0x1C8	DMACn_C6LLI	DMAC channel 6 linked list item	201
0x1CC	DMACn_C6CR	DMAC channel 6 control	202
0x1D0	DMACn_C6CFG	DMAC channel 6 configuration	204

Table 35. DMAC register list (continued)

Offset	Register	Description	Page
0x1E0	DMACn_C7SADR	DMAC channel 7 source address	200
0x1E4	DMACn_C7DADR	DMAC channel 7 destination address	201
0x1E8	DMACn_C7LLI	DMAC channel 7 linked list item	201
0x1EC	DMACn_C7CR	DMAC channel 7 control	202
0x1F0	DMACn_C7CFG	DMAC channel 7 configuration	204
0xFE0	DMACnPeriphID0	DMAC peripheral identification register 0 (bits 7:0)	205
0xFE4	DMACnPeriphID1	DMAC peripheral identification register 1 (bits 15:8)	205
0xFE8	DMACnPeriphID2	DMAC peripheral identification register 2 (bits 23:16)	205
0xFEC	DMACnPeriphID3	DMAC peripheral identification register 3 (bits 31:24)	206
0xFF0	DMACnPCellID0	DMAC PCell identification register 0 (bits 7:0)	206
0xFF4	DMACnPCellID1	DMAC PCell identification register 1 (bits 15:8)	206
0xFF8	DMACnPCellID2	DMAC PCell identification register 2 (bits 23:16)	207
0xFFC	DMACnPCellID3	DMAC PCell identification register 3 (bits 31:24)	207

18.3 DMAC register descriptions

DMACn_MIS DMAC masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	IntStatus							
R	R	R	R	R	R	R	R	R							

Address: DMACnBaseAddress + 0x000

Reset: 0x0000 0000

Description: Provides the masked interrupt status of the DMAC.

IntStatus Interrupt status. Status of the DMA channel interrupt after masking. When set, indicates that a specific DMA channel interrupt request is active (the request can be generated by either error or terminal count interrupt requests).

0: not active 1: active

0 Reserved for future use. Reading returns 0. Must be written with 0.

DMACn_TCMIS DMAC terminal count masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	IntTCStatus							
R	R	R	R	R	R	R	R	R							

Address: DMACnBaseAddress + 0x004

Reset: 0x0000 0000

Description: Determines if an interrupt was generated due to transaction completing (terminal count).

If the DMACINTTC interrupt request is used, this register returns the source of the interrupt request.

If the combined interrupt request (DMACINTCOMBINE) is used, this register must be used in conjunction with register *DMACn_MIS*.

IntTCStatus Interrupt terminal count request status. Indicates the terminal count status after masking.

0: transaction not completed 1: transaction completed

0 Reserved for future use. Reading returns 0. Must be written with 0.

DMACn_TCICR DMAC terminal count interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	IntTCClear							
R	R	R	R	R	R	R	R	W							

Address: DMACnBaseAddress + 0x008

Reset: 0x0000 0000

Description: Clears a terminal count interrupt request. Each bit can be programmed to be protected.

IntTCClear Terminal count request clear. Clears corresponding bits in *DMACn_TCMIS* and *DMACn_TCRIS* registers.

0: no effect 1: clear

0 Reserved for future use. Reading returns 0. Must be written with 0.

DMACn_EMIS DMAC error masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	IntErrorStatus							
R	R	R	R	R	R	R	R	R							

Address: DMACnBaseAddress + 0x00C

Reset: 0x0000 0000

Description: Determines if an interrupt was generated due to an error being generated.
If the DMACINTERROR interrupt request is used, this register returns the source of the error request.
If the combined interrupt request (DMACINTCOMBINE) is used, this register must be used in conjunction with register *DMACn_MIS*.

IntErrorStatus Interrupt error status. Indicates the status of the error request after masking.

0 Reserved for future use. Reading returns 0. Must be written with 0.

DMACn_EICR DMAC error interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	IntErrClr							
R	R	R	R	R	R	R	R	W							

Address: DMACnBaseAddress + 0x010

Reset: 0x0000 0000

Description: Clears the error interrupt requests. Each bit can be programmed to be protected.

IntErrClr Interrupt error clear. Clears corresponding bits in *DMACn_EMIS* and *DMACn_ERIS* registers.
0: no effect 1: clear

0 Reserved for future use. Reading returns 0. Must be written with 0.

DMACn_TCRIS DMAC terminal count raw interrupt register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	RawIntTCStatus							
R	R	R	R	R	R	R	R	R							

Address: DMACnBaseAddress + 0x014

Reset: 0x0000 0000

Description: Provides the raw status of DMA terminal count interrupts prior to masking. Indicates which DMA channels are requesting a transfer complete (terminal count interrupt) prior to masking.

RawIntTCStatus Terminal count interrupt raw status prior to masking. 1: active

0 Reserved for future use. Reading returns 0. Must be written with 0.

DMACn_ERIS DMAC error raw interrupt register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	RawIntErrorStatus							
R	R	R	R	R	R	R	R	R							

Address: DMACnBaseAddress + 0x018

Reset: 0x0000 0000

Description: Provides the raw status of DMA error interrupts prior to masking. Indicates which DMA channels are requesting an error interrupt prior to masking.

RawIntErrorStatus Raw interrupt error status prior to masking. 1: active

0 Reserved for future use. Reading returns 0. Must be written with 0.

DMACn_ECHSR DMAC enabled channel status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	EnabledChannels							
R	R	R	R	R	R	R	R	RH							

Address: DMACnBaseAddress + 0x01C

Reset: 0x0000 0000

Description: Indicates which DMA channels are enabled, as indicated by the *DMACn_CxCFG*.E bit.
A bit is cleared on completion of the DMA transfer.

EnabledChannels Channel enable status. 1: enabled

0 Reserved for future use. Reading returns 0. Must be written with 0.

DMACn_SBREQ DMAC status burst request register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SoftBReq															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SoftBReq															
R															

Address: DMACnBaseAddress + 0x020

Reset: 0x0000 0000

Description: Bit-wise indication of which sources are requesting DMA burst transfers. It is recommended that software and hardware peripheral requests are not used at the same time.

SoftBReq Software burst request.
0: inactive 1: active

DMACn_SSREQ DMAC status single request register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SoftSReq															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SoftSReq															
R															

Address: DMACnBaseAddress + 0x024

Reset: 0x0000 0000

Description: Bit-wise indication of which sources are requesting DMA single transfers. It is recommended that software and hardware peripheral requests are not used at the same time.

SoftSReq Software single request.
0: inactive 1: active

DMACn_SLBREQ DMAC status last burst request register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SoftLBReq															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SoftLBReq															
R															

Address: DMACnBaseAddress + 0x028

Reset: 0x0000 0000

Description: Bit-wise indication of which sources are requesting DMA last burst transfers.

SoftLBReq Software last burst request.
0: inactive 1: active

DMACn_SLSREQ DMAC status last single request register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SoftLSReq															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SoftLSReq															
R															

Address: DMACnBaseAddress + 0x02C

Reset: 0x0000 0000

Description: Bit-wise indication of which sources are requesting DMA last single transfers.

SoftLSReq Software last single request.
0: inactive 1: active

DMACn_CxSADR DMAC channel x source address registers (x = 0:7)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SrcAddr															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SrcAddr															
RW															

Address: DMACnBaseAddress + 0x100 + 0x20*x, where x = 0 to 7

Reset: 0x0000 0000

Description: Contains the byte-aligned source address of the data to be transferred. Each register is programmed directly by software before the channel is enabled. When the DMA channel is enabled this register is updated when the source address is incremented, and it is updated by following the linked list when a complete packet of data has been transferred.
Reading the register when the channel is active does not provide useful information because by the time the software has processed the value read, the channel might have progressed. It is intended to be read only when the channel has stopped, in which case it shows the source address of the last item read.

SrcAddr DMA source address.

DMACn_CxDADR DMAC channel x destination address registers (x = 0:7)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DestAddr															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DestAddr															
RW															

Address: DMACnBaseAddress +0x104 + 0x20*x, where x = 0 to 7

Reset: 0x0000 0000

Description: Contains the byte-aligned destination address of the data to be transferred. Each register is programmed by software before the DMA channel is enabled. When the channel is enabled, this register is updated when the destination address is incremented, and it is updated by following the linked list when a complete packet of data has been transferred.
Reading the register when the channel is active does not provide useful information because by the time the software has processed the value read, the channel might have progressed. It is intended to be read only when the channel has stopped, in which case it shows the source address of the last item read. Linked-list usage is possible (write-protection is not effective when the register is reloaded by the LLI logic). Source and destination addresses must be aligned to source and destination widths.

DestAddr DMA destination address.

DMACn_CxLLI DMAC channel x linked list item registers (x = 0:7)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LLI															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LLI														0	LM
RW														R	RW

Address: DMACnBaseAddress +0x108 + 0x20*x, where x = 0 to 7.

Reset: 0x0000 0000

Description: Contains a word-aligned address of the next linked list item (LLI). If the LLI is 0, then the current LLI is the last in the chain, and the DMA channel is disabled once all DMA transfers associated with it are completed. Programming this register when the DMA

channel is enabled has unpredictable side effects. Linked-list usage is possible (write protection is not effective when the register is reloaded by the LLI logic).

LLI Linked list item. Contains the address for the next LLI. Address bits [1:0] are 0.

LM LLI AHB master selection. Selects AHB master for loading the next LLI.
0: AHB master 1 1: AHB master 2

0 Reserved for future use. Reading returns 0. Must be written with 0.

DMACn_CxCR DMAC channel x control registers (x = 0:7)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I	Prot			DI	SI	D	S	DWidth			SWidth			DBSize	
RW	RW			RW	RW	RW	RW	RW			RW			RW	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBSize	SBSize			TransferSize											
RW	RW			RW											

Address: DMACnBaseAddress +0x10C + 0x20*x, where x = 0 to 7

Reset: 0x0000 0000

Description: Contains DMA channel control information. Programmed by software before the DMA channel is enabled. When the channel is enabled, the register is updated when a complete data packet has been transferred by following the linked list.

Reading the register when the channel is active does not provide useful information as by the time the software has processed the value read, the channel might have progressed. It should be read only when the channel has stopped.

AHB access information is provided to the source and destination peripherals when a transfer occurs. The transfer information is provided by programming (with software) the DMA channel (*DMACn_CxCR*.Prot and *DMACn_CxCFG*.L).

I Terminal count interrupt enable. Controls if the current LLI triggers the terminal count interrupt.

Prot[0] Protection 0. Indicates the access mode.
0: user mode 1: privileged mode

Prot[1] Protection 1. Indicates if bufferable access is possible. It can, for example, indicate to an AMBA bridge that the read can complete in zero wait states on the source bus without waiting for it to arbitrate for the destination bus and for the slave to accept the data.
0: not bufferable 1: bufferable

Prot[2] Protection 2. Indicates if cacheable access is possible. This bit can, for example, indicate to an AMBA bridge that when the first read of a burst of 8 arrives, it can transfer the whole burst of 8 reads on the destination bus, rather than pass the transactions through one at a time.
0: not cacheable 1: cacheable

DI Destination increment.
0: no increment 1: increment destination address after each transfer

SI Source increment.
0: no increment 1: increment source address after each transfer

D Destination transfer. AHB master selected for the destination transfer.
0: AHB master 0 1: AHB master 1

S Source transfer AHB master selected for the source transfer.
0: AHB master 0 1: AHB master 1

DWidth Destination transfer width. Source and destination widths can differ (the hardware automatically packs and unpacks the data as required) but transfers wider than the AHB master bus width are not possible. See *Table 37* for width details.

SWidth Source transfer width. Source and destination widths can differ (the hardware automatically packs and unpacks the data as required) but transfers wider than the AHB master bus are not possible. See *Table 37* for width details.

DBSize Destination burst size. Indicates the number of transfers that make up a destination burst transfer request. This value must be set to the burst size of the destination peripheral or, if the destination is memory, to the memory boundary size. The burst size is the amount of data that is transferred when the DMABREQ signal goes active in the destination peripheral. This burst size is not related to the AHB HBURST signal. See *Table 36* for details.

SBSize Source burst size. Indicates the number of transfers which make up a source burst transfer request. This value must be set to the burst size of the source peripheral, or if the destination is memory, to the memory boundary size. The burst size is the amount of data that is transferred when the DMABREQ signal goes active in the source peripheral. This burst size is not related to the AHB HBURST signal. See *Table 36* for burst size details.

TransferSize Transfer size. If the DMAC is not the flow controller, the transfer size must be programmed to zero. For writes, indicates the number of source width transfers to perform when DMA is the flow controller. For reads, indicates the number of transfers completed on the destination bus, it should be used only when a channel is enabled and then disabled.

0 Reserved for future use. Reading returns 0. Must be written with 0.

Table 36. Source or destination burst size

Bit value of DBSize or SBSize	Source or destination burst transfer request size
000	1
001	4
010	8
011	16
100	32
101	64
110	128
111	256

Table 37. Source or destination transfer width

Bit value of DWidth or SWidth	Source or destination width
000	Byte (8 bits)
001	Halfword (16 bits)
010	Word (32 bits)
011-111	reserved

DMACn_CxCFG DMAC channel x configuration registers (x = 0:7)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	H	A	L
R	R	R	R	R	R	R	R	R	R	R	R	R	RW	R	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITC	IE	FlowCntrl			DestPeripheral					SrcPeripheral					E
RW	RW	RW			RW					RW					RW

Address: DMACnBaseAddress +0x110 + 0x20*x, where x = 0 to 7

Reset: 0x0000 0000

Description: Configures the DMA channel. Not updated when a new LLI is requested.

H Halt. Drains the contents of the channel FIFO. Used with bits A and E to disable DMA channel cleanly.
0: allow DMA requests 1: ignore further source DMA requests

A Active. Used with the H and E bits to cleanly disable a DMA channel.
0: there is no data in the channel's FIFO 1: the channel's FIFO has data

L Lock. 1: enable locked transfer

ITC Terminal count interrupt mask.
0: mask out the terminal count interrupt of the relevant channel

IE Interrupt error mask.
0: mask out the error interrupt of the relevant channel

FlowCntrl Flow controller and transfer type. Indicates the flow controller and transfer type.
000: DMA controller, memory-to-memory transfer type
001: DMA controller, memory-to-peripheral transfer type
010: DMA controller, peripheral-to-memory transfer type
011: DMA controller, source-peripheral-to-destination-peripheral transfer type
100: destination peripheral controller, source-peripheral-to-destination-peripheral transfer type
101: peripheral controller, memory-to-peripheral transfer type
110: peripheral controller, peripheral-to-memory transfer type
111: source peripheral controller, source-peripheral-to-destination-peripheral transfer type

DestPeripheral DMA destination request peripheral. Ignored if the transfer destination is memory.

SrcPeripheral DMA source request peripheral. Ignored if the transfer source is memory.

E Channel enable. Enables a channel. The channel enable bit status can also be found by reading the register *DMACn_ECHSR*.
When the channel is disabled (by clearing this bit), the current AHB transfer (if one is in progress) completes and then disables the channel. Any data in the channel's FIFO is lost. Restoring the channel by simply setting the channel enable bit has unpredictable effects and the channel must be fully re-initialized.
The channel is also disabled, and the channel enable bit cleared, when the last LLI is reached or if a channel error is encountered.
If a channel must be disabled without losing data in a channel's FIFO, bit H must be set so that further DMA requests are ignored. Bit A must then be polled until it reaches 0, indicating that there is no data left in the channel's FIFO. Bit E can then be cleared.
0: disable 1: enable

DMACnPeriphID0 DMAC peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: DMACnBaseAddress + 0xFE0

Reset: 0x0000 0080

Description: PartNumber0 reads back as 0x80.

DMACnPeriphID1 DMAC peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: DMACnBaseAddress + 0xFE4

Reset: 0x0000 0008

Description: Designer0 reads back as 0x0. PartNumber1 reads back as 0x8.

DMACnPeriphID2 DMAC peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: DMACnBaseAddress + 0xFE8

Reset: 0x0000 0028

Description: Revision reads back as 0x2. Designer1 reads back as 0x8.

DMACnPeriphID3 DMAC peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: DMACnBaseAddress + 0xFEC

Reset: 0x0000 008A

Description: Configuration reads back as 0x8A (32 DMA requests).

DMACnPCellID0 DMAC PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	DMACPCellID0							
R	R	R	R	R	R	R	R	R							

Address: DMACnBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: DMACPCellID0 reads back as 0x0D.

DMACnPCellID1 DMAC PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	DMACPCellID1							
R	R	R	R	R	R	R	R	R							

Address: DMACnBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: DMACPCellID1 reads back as 0xF0.

DMACnPCellID2 DMAC PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	DMACPCellID2							
R	R	R	R	R	R	R	R	R							

Address: DMACnBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: DMACPCellID2 reads back as 0x05.

DMACnPCellID3 DMAC PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	DMACPCellID3							
R	R	R	R	R	R	R	R	R							

Address: DMACnBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: DMACPCellID3 reads back as 0xB1.

18.4 Programming the DMAC

Enabling and disabling a DMA channel

To enable the DMA channel:

1. Fully initialize a channel before enabling it.
2. Set the channel enable bit in the relevant DMA channel configuration register.

A DMA channel can be disabled in three ways:

- By writing directly to the channel enable bit in the relevant DMA channel configuration register (outstanding data in the FIFOs is lost). The current AHB transfer (if one is in progress) completes and the channel is disabled.
- By using the active and halt bits in conjunction with the channel enable bit (data in the FIFO is not lost):
  - Set bit H in the relevant channel configuration register. Any further DMA requests are now ignored.
  - Poll bit A in the relevant channel configuration register until it reaches 0. This bit indicates if there is any remaining data in the channel that should be transferred.
  - Clear bit E in the relevant channel configuration register.
- Wait until the transfer completes. The channel is then automatically disabled.

Setting up a new DMA transfer

1. If the channel is not set aside for the DMA transaction:
   - Read the *DMACn_ECHSR* register and find out which channels are inactive.
   - Choose an inactive channel that has the required priority.
2. Program the DMAC.

Halting a DMA channel

Set bit H in the relevant DMA channel configuration register. The current source request is serviced. Any further source DMA requests are ignored until bit H is cleared.

Programming a DMA channel

1. Choose a free DMA channel with the required priority (DMA channel 0 has the highest priority and DMA channel 7 the lowest priority).
2. Clear pending interrupts on the channel to be used by writing to the *DMACn_TCICR* and *DMACn_EICR* registers. The previous channel operation may have left interrupts inactive.
3. Write the source address into the *DMACn_CxSADR* register.
4. Write the destination address into the *DMACn_CxDADR* register.
5. Write the address of the next LLI into the *DMACn_CxLLI* register. If the transfer comprises a single packet of data, 0 must be written into this register.
6. Write control information and channel configuration information into *DMACn_CxCFG*. If the enable bit is set, the DMA channel is automatically enabled.

Scatter/gather

Scatter/gather is supported using linked lists, which means the source and destination areas do not need to occupy contiguous areas in the memory. When scatter/gather is not required, the DMACn_CxLLI register must be set to 0.

Linked list items (LLI)

An LLI consists of four words that are organized in the following order:

1. DMACn_CxSADR
2. DMACn_CxDADR
3. DMACn_CxLLI
4. DMACn_CxCR

Note: The DMACn_CxCFG DMA channel configuration register is not part of the LLI.

Programming the DMAC for scatter/gather DMA

1. Write the LLIs for complete DMA transfer to memory. Each LLI contains four words: source address, destination address, pointer to next LLI and control word.
   The linked list word pointer of the last LLI is set to 0.
2. Choose a free DMA channel with the required priority (DMA channel 0 has the highest priority and DMA channel 7 the lowest priority).
3. Write the first LLI (previously written to memory) to the relevant channel in the DMAC.
4. Write the channel configuration information to the channel configuration register and set the channel enable bit. The DMAC then transfers the first and subsequent packets of data as each LLI is loaded.
5. An interrupt can be generated at the end of each LLI depending on the DMACn_CxCFG.ITC bit. If this bit is set, an interrupt is generated at the end of the relevant LLI. The interrupt request must then be serviced, and the relevant bit in the *DMACn_TCICR* register must be set to clear the interrupt.

Interrupt requests

Interrupt requests can be generated when an AHB error is encountered, or at the end of a transfer (terminal count) after the current LLI's data has been transferred to the destination. Interrupts are masked by programming relevant bits in the DMACn_CxCFG channel registers.
Interrupt status registers are provided which group the interrupt requests from all the DMA channels *prior to* interrupt masking (*DMACn_TCRIS*, *DMACn_ERIS*) and *after* interrupt masking (*DMACn_TCMIS*, *DMACn_EMIS*).
The *DMACn_MIS* register combines both the DMACn_TCMIS and DMACn_EMIS requests into a single register so that the source of an interrupt can be found quickly.
Interrupts are cleared selectively by setting a bit in the *DMACn_EICR* or DMACn_TCICR registers.

19 General-purpose input/output (GPIO 0,1,2,3)

19.1 GPIO register addressing

Register addresses are provided as the GPIO base address, GPIOnBaseAddress, plus the register offset.

There are 4 GPIO base addresses.

Table 38. GPIO base addresses

GPIO base address	GPIO
0x101E 4000	[0:31]
0x101E 5000	[32:63]
0x101E 6000	[64:95]
0x101E 7000	[96:123]

Note: The 4 upper GPIOs of block GPIO3 are not delivered on balls.

19.2 GPIO register summary

Table 39. GPIO register list

Offset	Register name	Description	Page
0x000	GPIOn_DAT	GPIO data register	*211*
0x004	GPIOn_DATS	GPIO data set register	*212*
0x008	GPIOn_DATC	GPIO data clear register	*212*
0x00C	GPIOn_PDIS	GPIO pull disable register	*213*
0x010	GPIOn_DIR	GPIO direction register	*213*
0x014	GPIOn_DIRS	GPIO direction set register	*214*
0x018	GPIOn_DIRC	GPIO direction clear register	*214*
0x01C	GPIOn_SLPM	GPIO sleep mode register	*215*
0x020	GPIOn_AFSLA	GPIO alternate function select register A	*215*
0x024	GPIOn_AFSLB	GPIO alternate function select register B	*216*
0x040	GPIOn_RIMSC	GPIO rising edge interrupt mask set clear register	*219*
0x044	GPIO_FIMSC	GPIO falling edge interrupt mask set clear register	*219*
0x048	GPIOn_IS	GPIO interrupt status register	*220*
0x04C	GPIOn_IC	GPIO interrupt clear register	*221*
0x050	GPIOn_RWMSC	GPIO rising edge wake-up mask set clear register	*221*
0x054	GPIO_FWMSCn	GPIO falling edge wake-up mask set clear register	*222*
0x058	GPIOn_WKS	GPIO wake-up status register	*222*
0xFE0	GPIOnPeriphID0	GPIO peripheral identification register 0 (bits 7:0)	*223*

Table 39. GPIO register list (continued)

Offset	Register name	Description	Page
0xFE4	GPIOnPeriphID1	GPIO peripheral identification register 1 (bits 15:8)	223
0xFE8	GPIOnPeriphID2	GPIO peripheral identification register 2 (bits 23:16)	224
0xFEC	GPIOnPeriphID3	GPIO peripheral identification register 3 (bits 31:24)	224
0xFF0	GPIOnPCellID0	GPIO PCell identification register 0 (bits 7:0)	224
0xFF4	GPIOnPCellID1	GPIO PCell identification register 1 (bits 15:8)	225
0xFF8	GPIOnPCellID2	GPIO PCell identification register 2 (bits 23:16)	225
0xFFC	GPIOnPCellID3	GPIO PCell identification register 3 (bits 31:24)	225

The GPIO block comprises 32 programmable input/output lines. When software control mode is enabled, a data register and a data direction register controls data on these lines. A read of the data register gives the GPIO pins status, whether they are configured as input or output. Writing to the data register only affects the pins that are configured as outputs.

19.3 GPIO register descriptions

GPIOn_DAT GPIO data register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT	GPDAT
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x000

Reset: 0xXXXX XXXX[a]

Description: In software control mode, values written to this register are transferred to the GPOUT pins if the respective pins have been configured as outputs in *GPIOn_DIR*. This only applies when the GPIO alternate function is not enabled and when

a. The reset value is GPIO dependent: for a line with pull-down enabled at power-up, the GPDAT initial value is 0; for a line with pulll-up enabled at power-up, the GPDAT initial value is 1. The GPDAT content is not changed by system (PRESETnot) reset.

PMU_CTRL.IOFORCE is not asserted if the corresponding bit in register *GPIOn_SLPM* is cleared.

So that independent software drivers can set their GPIO bits without affecting other pins, the data register has associated set (*GPIOn_DATS*) and clear (*GPIOn_DATC*) registers.

When the GPIO pin is an input (including in alternate function) and when pull-up/pull-down are enabled, (corresponding *GPIOn_PDIS*.PDIS x set to 0), the GPIO_DAT register defines the pull resistor polarity (0 = pull-down enabled, 1 = pull-up enabled).

This register is not changed by PRESETnot system reset. It is cleared by power-on reset only.

GPDAT[31:0] GPIOx data. In read operations, returns the GPIO input pin value (GPIN) when configured as input, or the last written value when configured as output.

In write operations, defines the GPIO line level (when configured as an output and the alternate function is disabled), or the pull resistor polarity (when configured as an input, regardless of the alternate function) if pull-up/-down is enabled.

GPIOn_DATS GPIO data set register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPDATS31	GPDATS30	GPDATS29	GPDATS28	GPDATS27	GPDATS26	GPDATS25	GPDATS24	GPDATS23	GPDATS22	GPDATS21	GPDATS20	GPDATS19	GPDATS18	GPDATS17	GPDATS16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPDATS15	GPDATS14	GPDATS13	GPDATS12	GPDATS11	GPDATS10	GPDATS9	GPDATS8	GPDATS7	GPDATS6	GPDATS5	GPDATS4	GPDATS3	GPDATS2	GPDATS1	GPDATS0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x004

Reset: 0x0000 0000

Description: Sets specific bits of the *GPIOn_DAT* register. Reading returns the GPIOn_DAT register content. All bits are cleared by PORnot reset only. Not changed by PRESETnot reset.

GPDATS[31:0] GPIOx data set.

0: no effect 1: sets corresponding bit in GPIOn_DAT

GPIOn_DATC GPIO data clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPDATC31	GPDATC30	GPDATC29	GPDATC28	GPDATC27	GPDATC26	GPDATC25	GPDATC24	GPDATC23	GPDATC22	GPDATC21	GPDATC20	GPDATC19	GPDATC18	GPDATC17	GPDATC16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPDATC15	GPDATC14	GPDATC13	GPDATC12	GPDATC11	GPDATC10	GPDATC9	GPDATC8	GPDATC7	GPDATC6	GPDATC5	GPDATC4	GPDATC3	GPDATC2	GPDATC1	GPDATC0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x008

Reset: 0x0000 0000

Description: Clears specific bits of the *GPIOn_DAT* register. Reading returns the GPIOn_DAT register content. All bits are cleared by PORnot reset only. Not changed by PRESETnot reset.

GPDATC[31:0] GPIOx data clear.

0: no effect 1: clears corresponding bit in GPIOn_DAT

GPIOn_PDIS GPIO pull disable register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PDIS 31	PDIS 30	PDIS 29	PDIS 28	PDIS 27	PDIS 26	PDIS 25	PDIS 24	PDIS 23	PDIS 22	PDIS 21	PDIS 20	PDIS 19	PDIS 18	PDIS 17	PDIS 16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDIS 15	PDIS 14	PDIS 13	PDIS 12	PDIS 11	PDIS 10	PDIS 9	PDIS 8	PDIS 7	PDIS 6	PDIS 5	PDIS 4	PDIS 3	PDIS 2	PDIS 1	PDIS 0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x00C

Reset: 0x0000 0000[b]

Description: Enables or disables the pull-up and pull-down associated with the GPIO pad. Regardless of the setting in this register, pull-up and pull-down are disabled when the pin is set in the output direction (by programming the *GPIOn_DIR* register when the GPIO line is not in alternate function mode, or through the alternate function peripheral). All bits are cleared by PORnot reset only. Not changed by PRESETnot system reset.

PDIS[31:0] GPIOx pull-up/-down disable. Returns the last value written in the register.

0: enabled if pin is an input (default after reset) 1: disabled

GPIOn_DIR GPIO direction register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPDIR 31	GPDIR 30	GPDIR 29	GPDIR 28	GPDIR 27	GPDIR 26	GPDIR 25	GPDIR 24	GPDIR 23	GPDIR 22	GPDIR 21	GPDIR 20	GPDIR 19	GPDIR 18	GPDIR 17	GPDIR 16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPDIR 15	GPDIR 14	GPDIR 13	GPDIR 12	GPDIR 11	GPDIR 10	GPDIR 9	GPDIR 8	GPDIR 7	GPDIR 6	GPDIR 5	GPDIR 4	GPDIR 3	GPDIR 2	GPDIR 1	GPDIR 0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x010

Reset: 0x0000 0000

Description: Configures the direction of the GPIO pin to be input or output. Only applies when the alternate function of the GPIO is not enabled (corresponding bit cleared in register

b. For GPIO3, the reset value is 0xFFFF FFFF.

GPIOn_AFSLA/B), and when PMU_CTRL.IOFORCE is not asserted. When the pin is programmed as an output, pull-up and pull-down are disabled, regardless of the setting in GPIOn_PDIS. All bits are cleared by PORnot reset only. Not changed by PRESETnot system reset.

GPDIR[31:0] GPIOx direction. GPIO pin direction when the corresponding GPIOn_AFSLA/B bit is cleared and when PMU_CTRL.IOFORCE is low.

0: corresponding output pin is input.

1: corresponding output pin is output, pull-up and pull-down are disabled.

GPIOn_DIRS GPIO direction set register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPDIRS31	GPDIRS30	GPDIRS29	GPDIRS28	GPDIRS27	GPDIRS26	GPDIRS25	GPDIRS24	GPDIRS23	GPDIRS22	GPDIRS21	GPDIRS20	GPDIRS19	GPDIRS18	GPDIRS17	GPDIRS16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPDIRS15	GPDIRS14	GPDIRS13	GPDIRS12	GPDIRS11	GPDIRS10	GPDIRS9	GPDIRS8	GPDIRS7	GPDIRS6	GPDIRS5	GPDIRS4	GPDIRS3	GPDIRS2	GPDIRS1	GPDIRS0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x014

Reset: 0x0000 0000

Description: Sets specific bits of the GPIOn_DIR register. Reading returns the GPIOn_DIR content. All bits are cleared by PORnot reset only. Not changed by PRESETnot system reset.

GPDIRS[31:0] GPIOx direction set.

0: no effect 1: sets corresponding bit in register GPIOn_DIR

GPIOn_DIRC GPIO direction clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPDIRC31	GPDIRC30	GPDIRC29	GPDIRC28	GPDIRC27	GPDIRC26	GPDIRC25	GPDIRC24	GPDIRC23	GPDIRC22	GPDIRC21	GPDIRC20	GPDIRC19	GPDIRC18	GPDIRC17	GPDIRC16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPDIRC15	GPDIRC14	GPDIRC13	GPDIRC12	GPDIRC11	GPDIRC10	GPDIRC9	GPDIRC8	GPDIRC7	GPDIRC6	GPDIRC5	GPDIRC4	GPDIRC3	GPDIRC2	GPDIRC1	GPDIRC0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x018

Reset: 0x0000 0000

Description: Clears specific bits of the GPIOn_DIR register. A read returns the GPIOn_DIR contents. All bits are cleared by PORnot reset only. Not changed by PRESETnot system reset.

GPDIRC[31:0] GPIOx direction clear.

0: no effect 1: clears corresponding bit in register GPIOn_DIR

GPIOn_SLPM GPIO sleep mode register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLPM 31	SLPM 30	SLPM 29	SLPM 28	SLPM 27	SLPM 26	SLPM 25	SLPM 24	SLPM 23	SLPM 22	SLPM 21	SLPM 20	SLPM 19	SLPM 18	SLPM 17	SLPM 16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLPM 15	SLPM 14	SLPM 13	SLPM 12	SLPM 11	SLPM 10	SLPM 9	SLPM 8	SLPM 7	SLPM 6	SLPM 5	SLPM 4	SLPM 3	SLPM 2	SLPM 1	SLPM 0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x01C

Reset: 0x0000 0000

Description: Defines the GPIO mode when sleep or deep-sleep mode is entered, that is when *PMU_CTRL*.IOFORCE is high. Each GPIO can be configured to be forced in the input direction with pull-up or pull-down enabled when IOFORCE is high, overriding the normal settings defined by *GPIOn_DIR*/*GPIOn_DAT*/*GPIOn_PDIS* and *GPIOn_AFSLA*/*B*. When PMU_CTRL.IOFORCE returns low (after sleep or deep-sleep exit) the GPIOs return to the normal settings defined by GPIOn_DIR/GPIOn_DAT/GPIOn_PDIS and GPIOn_AFSLA/B.
A read from this register returns the last value written into it. All bits are cleared by PORnot reset only. Not changed by PRESETnot system reset.

SLPM[31:0] GPIOx sleep mode. Defines the GPIO mode when sleep or deep-sleep mode is entered.

0: switched to input with pull-up/-down enabled when PMU_CTRL.IOFORCE is high.

1: remains controlled by registers GPIOn_DAT/GPIOn_DIR/GPIOn_PDIS (when GPIOn_AFSLA/B = 00) or on-chip peripherals (when GPIOn_AFSLA/B not = 00) when IOFORCE is high.

GPIOn_AFSLA GPIO alternate function select A register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPAFS A31	GPAFS A30	GPAFS A29	GPAFS A28	GPAFS A27	GPAFS A26	GPAFS A25	GPAFS A24	GPAFS A23	GPAFS A22	GPAFS A21	GPAFS A20	GPAFS A19	GPAFS A18	GPAFS A17	GPAFS A16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPAFS A15	GPAF SA14	GPAFS A13	GPAFS A12	GPAFS A11	GPAFS A10	GPAFS A9	GPAFS A8	GPAFS A7	GPAFS A6	GPAFS A5	GPAFS A4	GPAFS A3	GPAFS A2	GPAFS A1	GPAFS A0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x020

Reset: 0x0000 0000. The reset value for GPIO3 is 0xFFF FFFF.

Description: These two registers select together the alternate functions for the GPIO lines (a line may have up to three alternate functions). All bits are cleared by PORnot reset only, therefore all GPIO direction/levels are controlled by registers *GPIOn_DIR*/*GPIOn_DAT*/*GPIOn_PDIS* by default after power-on reset. The registers are not changed by PRESETnot system reset.

GPIOn_AFSLB GPIO alternate function select B register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPAFBA31	GPAFBA30	GPAFBA29	GPAFBA28	GPAFBA27	GPAFBA26	GPAFBA25	GPAFBA24	GPAFBA23	GPAFBA22	GPAFBA21	GPAFBA20	GPAFBA19	GPAFBA18	GPAFBA17	GPAFBA16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPAFSB15	GPAFSB14	GPAFSB13	GPAFSB12	GPAFSB11	GPAFSB10	GPAFSB9	GPAFSB8	GPAFSB7	GPAFSB6	GPAFSB5	GPAFSB4	GPAFSB3	GPAFSB2	GPAFSB1	GPAFSB0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x024

Reset: 0x0000 0000

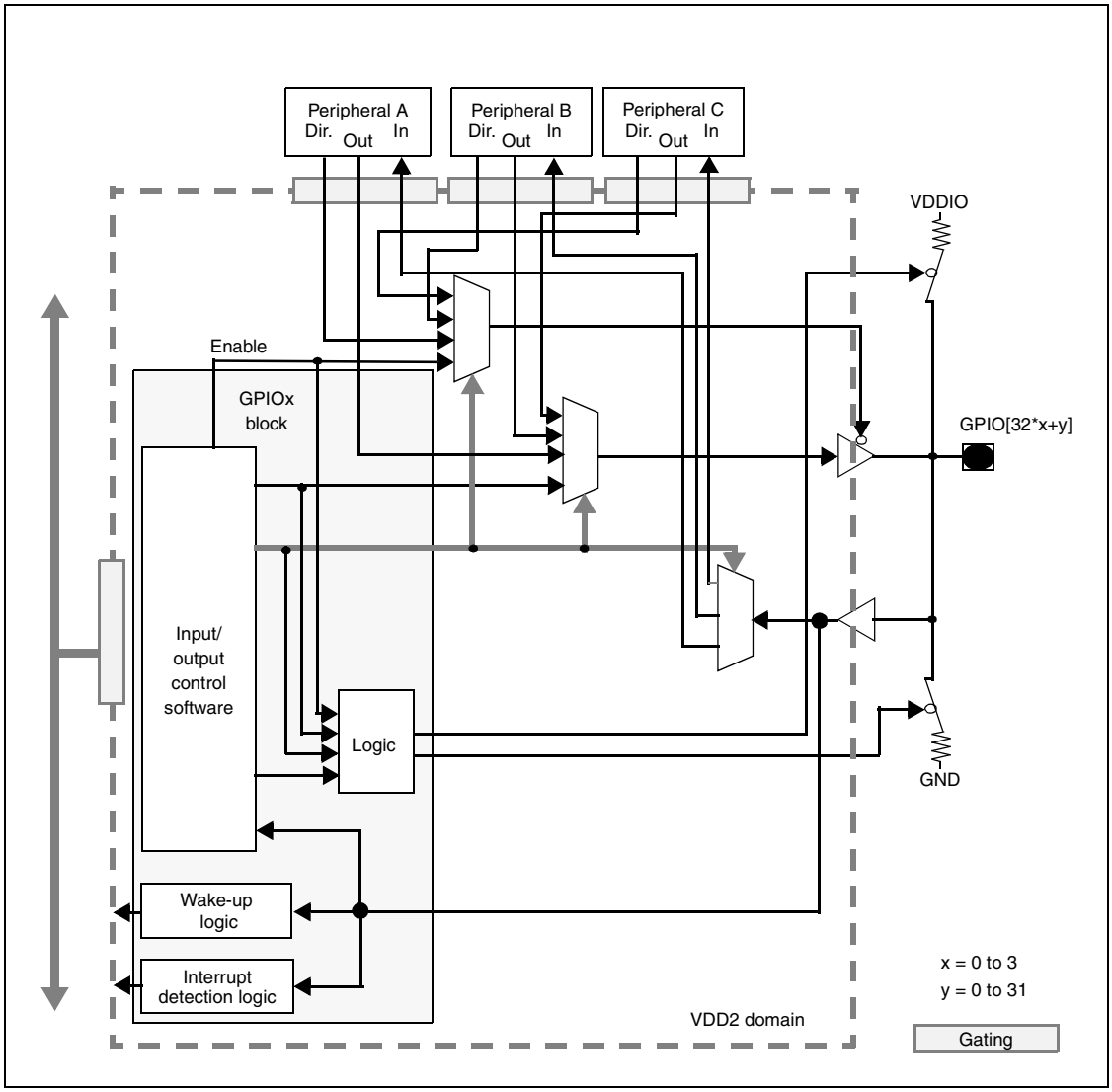
Description:

Table 40. Alternate function selection

GPIOn_AFSLA (bit y)	GPIOn_AFSLB (bit y)	GPIO [y] line mode
0	0	General purpose IO line: the direction of the line is defined by bit y of *GPIOn_DIR* register, the level is defined by bit y of *GPIOn_DAT* register when configured as output, or can be read from bit y of GPIOn_DAT register when configured as an input.
1	0	Alternate function A: the GPIO line y is under control of an on-chip peripheral.(1)
0	1	Alternate function B: the GPIO line y is under control of an on-chip peripheral.(1)
1	1	Alternate function C: the GPIO line y is under control of an on-chip peripheral.(1)

1. See *Figure 9 on page 218*.

Figure 9. GPIO mode control multiplexors



In alternate function mode:

- the direction and level of the GPIO pin is controlled by the associated peripheral,
- *GPIOn_DIR* and *GPIOn_DAT* register values are ignored,
- data and control transfers are controlled by the associated peripheral or internal logic.
- the pull-up/down resistor is not automatically switched off if the alternate function is using the GPIO as an output pin, it can only be disabled by software setting *GPIOn_PDIS*.PDIS.

The multiplexors for the alternate functions are not part of the GPIO block but, for implementation reasons, are instantiated near the I/O pads.

GPIOn_RIMSC Rising edge interrupt mask set clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPRIM 31	GPRIM 30	GPRIM 29	GPRIM 28	GPRIM 27	GPRIM 26	GPRIM 25	GPRIM 24	GPRIM 23	GPRIM 22	GPRIM 21	GPRIM 20	GPRIM 19	GPRIM 18	GPRIM 17	GPRIM 16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPRIM 15	GPRIM 14	GPRIM 13	GPRIM 12	GPRIM 11	GPRIM 10	GPRIM 9	GPRIM 8	GPRIM 7	GPRIM 6	GPRIM 5	GPRIM 4	GPRIM 3	GPRIM 2	GPRIM 1	GPRIM 0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x040

Reset: 0x0000 0000

Description: When set, the corresponding pins trigger their individual interrupts and the combined GPIOINTR line when a rising edge transition is detected on the corresponding GPIO line. Clearing a bit disables interrupt triggering on the rising edge for that pin. All bits are cleared by PORnot and system reset. See *GPIOn_IS* register description for more details.

GPRIM[31:0] GPIO x rising-edge detection interrupt mask. Shows corresponding pin rising-edge detection interrupt status.

0: pin rising-edge detection interrupt masked 1: pin rising-edge detection interrupt enabled

GPIOn_FIMSC GPIO falling edge interrupt mask set clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPFIM 31	GPFIM 30	GPFIM 29	GPFIM 28	GPFIM 27	GPFIM 26	GPFIM 25	GPFIM 24	GPFIM 23	GPFIM 22	GPFIM 21	GPFIM 20	GPFIM 19	GPFIM 18	GPFIM 17	GPFIM 16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPFIM 15	GPFIM 14	GPFIM 13	GPFIM 12	GPFIM 11	GPFIM 10	GPFIM 9	GPFIM 8	GPFIM 7	GPFIM 6	GPFIM 5	GPFIM 4	GPFIM 3	GPFIM 2	GPFIM 1	GPFIM 0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x044

Reset: 0x0000 0000

Description: When set, the corresponding pins trigger their individual interrupts and the combined GPIOINTR line when a falling edge transition is detected on the corresponding GPIO line. Clearing a bit disables interrupt triggering on the falling edge for that pin. All bits are cleared by PORnot and system reset.See *GPIOn_IS* register description for more details.

GPFIM[31:0] GPIOx falling-edge detection interrupt mask. Shows corresponding pin falling-edge detection interrupt status.

0: pin falling-edge detection interrupt masked 1: pin falling-edge detection interrupt enabled

GPIOn_IS GPIO interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIS31	GPIS30	GPIS29	GPIS28	GPIS27	GPIS26	GPIS25	GPIS24	GPIS23	GPIS22	GPIS21	GPIS20	GPIS19	GPIS18	GPIS17	GPIS16
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIS15	GPIS14	GPIS13	GPIS12	GPIS11	GPIS10	GPIS9	GPIS8	GPIS7	GPIS6	GPIS5	GPIS4	GPIS3	GPIS2	GPIS1	GPIS0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: GPIOnBaseAddress + 0x048

Reset: 0x0000 0000

Description: Bits read as 1 in this register mean a rising edge transition (if corresponding *GPIOn_RIMSC* bit is set) or a falling edge transition (if corresponding *GPIOn_FIMSC* bit is set) occurs. Bits read as 0 indicate that either no interrupt has been generated, or the interrupt is disabled (masked). Bits are cleared by PORnot and system reset.

GPIO interrupts are controlled by four registers. Any GPIO can be independently enabled or disabled (masked) for interrupt generation. For each GPIO, one can select which edge (rising, falling or both) will trigger an interrupt. All individual GPIO interrupt signals for the 32 GPIOs of a block are ORed together to produce a single interrupt output, GPIOINTR, sent to the interrupt controller. The software must clear the interrupt to enable any further interrupts.

Setting *GPIOn_RIMSC* causes the corresponding GPIO to send an interrupt when a rising edge is detected on the GPIO input.

Setting *GPIOn_FIMSC* causes the corresponding GPIO to send an interrupt when a falling edge is detected on the GPIO input.

Therefore, when the same bits are set in GPIOn_RISMC and GPIOn_FIMSC, both edges (rising and falling) will send an interrupt, and when the same bits are cleared in GPIOn_RISMC and GPIOn_FIMSC, the interrupt detection for that GPIO is disabled (masked). Detected interrupts are recorded in the GPIOn_IS register and are cleared by writing 1 in the corresponding bit of the *GPIOn_IC* register.

Interrupt detection logic works with PCLK. The GPIO signals are resynchronized to PCLK, except when the GPIO is an output (*GPIOn_DIRx* = 1, with *GPIOn_AFSLAx*/*GPIOn_AFSLBx* = 0); since the data is already synchronous to PCLK, the synchronization stage is bypassed.

GPIS[31:0] GPIO x interrupt status. Indicates status of interrupt on corresponding pin. A write has no effect.
0: GPIO line interrupt not active 1: GPIO line asserting interrupt

GPIOn_IC GPIO interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPIC 31	GPIC 30	GPIC 29	GPIC 28	GPIC 27	GPIC 26	GPIC 25	GPIC 24	GPIC 23	GPIC 22	GPIC 21	GPIC 20	GPIC 19	GPIC 18	GPIC 17	GPIC 16
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIC 15	GPIC 14	GPIC 13	GPIC 12	GPIC 11	GPIC 10	GPIC 9	GPIC 8	GPIC 7	GPIC 6	GPIC 5	GPIC 4	GPIC 3	GPIC 2	GPIC 1	GPIC 0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Address: GPIOnBaseAddress + 0x04C

Reset: 0x0000 0000

Description: Write-only register. All bits are cleared by PORnot and system reset.

GPIC[31:0] GPIO x interrupt clear. Clears the corresponding interrupt edge detection logic register (*GPIOn_RIMSC* or *GPIOn_FIMSC*).

0: no effect 1: clears interrupt edge detection logic

GPIOn_RWMSC GPIO rising edge wake-up mask set clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR WM31	GPR WM30	GPR WM29	GPR WM28	GPR WM27	GPR WM26	GPR WM25	GPR WM24	GPR WM23	GPR WM22	GPR WM21	GPR WM20	GPR WM19	GPR WM18	GPR WM17	GPR WM16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR WM15	GPR WM14	GPR WM13	GPR WM11	GPR WM11	GPR WM10	GPR WM9	GPR WM8	GPR WM7	GPR WM6	GPR WM5	GPR WM4	GPR WM3	GPR WM2	GPR WM1	GPR WM0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x050

Reset: 0x0000 0000

Description: When bits are set to 1 in this register, the corresponding pins trigger a system wake-up from sleep/deep-sleep (asserting GPIOWKUP line) when a rising-edge transition is detected on the corresponding GPIO line. Clearing a bit disables wake-up triggering on the rising edge for that pin. All bits are cleared by PORnot reset only. This register is not changed by system reset. See *GPIOn_WKS* register description for more details.

GPRWM[31:0] GPIO x rising-edge detection wakeup mask. Indicates corresponding pin rising-edge detection wakeup status.

0: corresponding pin rising-edge detection for wakeup is masked.

1: corresponding pin rising-edge detection for wakeup is enabled.

GPIOn_FWMSC GPIO falling edge wake-up mask set clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPFWM31	GPFWM30	GPFWM29	GPFWM28	GPFWM27	GPFWM26	GPFWM25	GPFWM24	GPFWM23	GPFWM22	GPFWM21	GPFWM20	GPFWM19	GPFWM18	GPFWM17	GPFWM16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPFWM15	GPFWM14	GPFWM13	GPFWM11	GPFWM11	GPFWM10	GPFWM9	GPFWM8	GPFWM7	GPFWM6	GPFWM5	GPFWM4	GPFWM3	GPFWM2	GPFWM1	GPFWM0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: GPIOnBaseAddress + 0x054

Reset: 0x0000 0000

Description: When bits are set to 1 in this register, the corresponding pins trigger a system wake-up from sleep/deep-sleep (asserting GPIOWKUP line) when a falling-edge transition is detected on the corresponding GPIO line. Clearing a bit disables wake-up triggering on the falling edge for that pin. All bits are cleared by PORnot reset only. The register is not changed by system reset. See *GPIOn_WKS* register description for more details.

GPFWM[31:0] GPIO x falling-edge detection wakeup mask. Indicates corresponding pin falling-edge detection wakeup status.

0: corresponding pin falling-edge detection for wakeup is masked.

1: corresponding pin falling-edge detection for wakeup is enabled.

GPIOn_WKS GPIO wakeup status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPWKS31	GPWKS30	GPWKS29	GPWKS28	GPWKS27	GPWKS26	GPWKS25	GPWKS24	GPWKS23	GPWKS22	GPWKS21	GPWKS20	GPWKS19	GPWKS18	GPWKS17	GPWKS16
RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPWKS15	GPWKS14	GPWKS13	GPWKS11	GPWKS11	GPWKS10	GPWKS9	GPWKS8	GPWKS7	GPWKS6	GPWKS5	GPWKS4	GPWKS3	GPWKS2	GPWKS1	GPWKS0
RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH

Address: GPIOnBaseAddress + 0x058

Reset: 0x0000 0000

Description: Indicates which GPIO triggered the wake-up event. Several bits may be read as 1 since memorization is performed from the entry to sleep/deep-sleep mode until an actual exit from this mode (several 32 kHz clock cycles). All bits are cleared by a PORnot reset. The register is not changed by PRESETnot system reset.

GPIO wake-ups are controlled by two registers. Any GPIO can be independently enabled or disabled (masked) for wakeup event generation. For each GPIO, one can select which edge (rising, falling or both) will trigger a wakeup from sleep/deep-sleep mode. All individual GPIO wakeup signals for the 32 GPIOs of a block are ORed together to produce a single wakeup output, GPIOWKUP, sent to the PMU. After wakeup, the software determines which GPIO triggered the wakeup event.

Setting *GPIOn_RWMSC* causes the corresponding GPIO to send a wakeup event when a rising edge is detected on the GPIO input (asynchronous detection).
Setting *GPIOn_FWMSC* causes the corresponding GPIO to send a wakeup event when a falling edge is detected on the GPIO input (asynchronous detection).
Therefore, when the same bits are set in *GPIOn_RWMSC* and *GPIOn_FWMSC*, both edges (rising and falling) will send a wakeup event. When the same bits are cleared in these registers, wakeup generation for that GPIO is disabled (masked). Detected wakeups are recorded in GPIOn_WKS. When *PMU_CTRL*.IOFORCE is low, this register is cleared, a read returns zero and writes are ignored. Therefore, the software must read this register upon wakeup to determine which GPIO woke up the device from sleep/deep-sleep before restoring normal GPIO usage by writing 1 to PMU_CTRL.IOFORCE.

GPWKS[31:0] GPIO x wake-up status. Indicates which GPIO triggered the last wakeup.
0: corresponding GPIO did not trigger wake-up 1: corresponding GPIO triggered wake-up

GPIOnPeriphID0 GPIO peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: GPIOnBaseAddress + 0xFE0

Reset: 0x0000 0060

Description: PartNumber0 returns 0x60.

GPIOnPeriphID1 GPIO peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: GPIOnBaseAddress + 0xFE4

Reset: 0x0000 0000

Description: Designer0 returns 0x0, PartNumber1 returns 0x0.

GPIOnPeriphID2 GPIO peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: GPIOnBaseAddress + 0xFE8

Reset: 0x0000 0018

Description: Revision returns 0x1, Designer1 returns 0x8.

GPIOnPeriphID3 GPIO peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: GPIOnBaseAddress + 0xFEC

Reset: 0x0000 001F

Description: Configuration returns 0x1F (32 lines).

GPIOnPCellID0 GPIO PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	GPIOPCellID0							
R	R	R	R	R	R	R	R	R							

Address: GPIOnBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: GPIOPCellID0 returns 0x0D.

GPIOnPCellID1 GPIO PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	GPIOPCellID1							
R	R	R	R	R	R	R	R	R							

Address: GPIOnBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: GPIOPCellID1 returns 0xF0.

GPIOnPCellID2 GPIO PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	GPIOPCellID2							
R	R	R	R	R	R	R	R	R							

Address: GPIOnBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: GPIOPCellID2 returns 0x05.

GPIOnPCellID3 GPIO PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	GPIOPCellID3							
R	R	R	R	R	R	R	R	R							

Address: GPIOnBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: GPIOPCellID3 returns 0xB1.

20 USB On-The-Go interface (USBOTG) registers

20.1 L2CC register addressing

Register addresses are provided as the L2CC base address, L2CCBaseAddress, plus the register offset.

The L2CCBaseAddress is 0x1017 0000.

20.2 L2CC register summary

Table 41. USBOTG register and memory map

Offset	Register	Description
0x00 - 0x0F	Common USB	Core status information and core control.
0x10 - 0x1F	Indexed endpoint control and status	Status and control information for the currently selected (by index register) endpoint. The registers mapped into this section depend on whether the core is in peripheral or host mode and on the value of the index register (*OTG_INDX*).
0x20 - 0x3F	FIFOs	Access to the endpoint FIFOs.
0x60 - 0x6F	Additional control	Additional device status and control information.
0x70 - 0x77	ULPI control	Additional status, control and data information.
0x78 - 0x7F	Additional configuration	Additional configuration registers.
0x80 - 0xFF	Target endpoint control	Target function and hub address details for each endpoint. Accessible in host mode only.
0x100 - 0x1FF	Non-indexed endpoint control and status	Direct access to endpoint control and status registers. The registers available at range (0x10 - 0x1F) accessible independently of the index register setting 0x100 to 0x10F are EP0 registers, 0x110 to 0x11F are EP1 registers, 0x120 to 0x12F are EP2 registers and so on.
0x200	Top-level control and status	Clock and reset generators, DMA selector lines and mode selection.

Configuration of the OTG core register and memory map depends upon the host or device mode setting and the setting of the index register.

Table 42. USBOTG register list

Offset	Register name	Description	Host(1)	Page
Common registers				
0x00	OTG_FADDR	Function address register (device mode only)		*232*
0x01	OTG_PWR	Power register		*232*
0x02	OTG_INTTX	Tx interrupt register		*233*
0x04	OTG_INTRX	Rx interrupt register		*233*

Table 42. USBOTG register list (continued)

Offset	Register name	Description	Host(1)	Page
0x06	OTG_INTTXEN	Interrupt enable register for *OTG_INTTX*		*234*
0x08	OTG_INTRXEN	Interrupt enable register for *OTG_INTRX*		*234*
0x0A	OTG_INTUSB	Interrupt register for common USB interrupts		*234*
0x0B	OTG_INTUSBEN	Interrupt enable register for *OTG_INTUSB*		*235*
0x0C	OTG_FMNO	Frame number register		*235*
0x0E	OTG_INDX	Index register		*236*
Indexed registers				
0x10	OTG_TXMAXP	Max packet size register (Tx EPs)		*236*
0x12	OTG_TXCSR	Control/status register (Tx EPs)		*238*
0x14	OTG_RXMAXP	Max packet size register (Rx EPs)		*239*
0x16	OTG_RXCSR	Control/status register (Rx EPs)		*241*
0x18	OTG_RXCNT	Bytes received counter (Rx EPs)		*243*
0x1A	OTG_TXTYPE	Transaction control register (Tx EPs)	Y	*243*
0x1B	OTG_TXINTV	Polling interval register (Tx EPs)	Y	*244*
0x1C	OTG_RXTYPE	Transaction control register (Rx EPs)	Y	*244*
0x1D	OTG_RXINTV	Polling interval register (Rx EPs)	Y	*245*
0x1F	OTG_CFD	Core configuration (for EP0 only) register		*245*
FIFOs				
0x20	OTG_FIFO_0	FIFO for EP0		-
0x24	OTG_FIFO_1	FIFO for EP1		-
0x28	OTG_FIFO_2	FIFO for EP2		-
0x2C	OTG_FIFO_3	FIFO for EP3		-
0x30	OTG_FIFO_4	FIFO for EP4		-
0x34	OTG_FIFO_5	FIFO for EP5		-
0x38	OTG_FIFO_6	FIFO for EP6		-
0x3C	OTG_FIFO_7	FIFO for EP7		-
Dynamic FIFO control registers				
0x60	OTG_DEVCTL	Device control register		*247*
0x62	OTG_TXFSZ	Tx FIFO size register		*248*
0x63	OTG_RXFSZ	Rx FIFO size register		*248*
0x64	OTG_TXFA	Tx FIFO address register		*249*
0x66	OTG_RXFA	Rx FIFO address register		*249*
0x6C	OTG_HWVER	Hardware version number register		*371*
ULPI registers				
0x70	OTG_UVBCTRL	ULPI VBUS control register		*249*

Table 42. USBOTG register list (continued)

Offset	Register name	Description	Host(1)	Page
0x71	OTG_UCKIT	ULPI VBUS CarKit register		250
0x72	OTG_UINTMASK	ULPI INT mask register		250
0x73	OTG_UINTSRC	ULPI INT source register		251
0x74	OTG_UREGDATA	ULPI Reg data register		251
0x75	OTG_UREGADDR	ULPI Reg address register		252
0x76	OTG_UREGCTRL	ULPI Reg control register		252
0x77	OTG_URAWDATA	ULPI raw data register (async)		252
0x77	OTG_URAWDATA	ULPI raw data register (sync)		252
Additional control registers				
0x78	OTG_EPINFO	EP information register		253
0x79	OTG_RAMINFO	RAM information register		254
0x7A	OTG_LINKINFO	Link information register		254
0x7B	OTG_VPLEN	VPLEN register		254
0x7C	OTG_HSEOF1	HS time buffer register		255
0x7D	OTG_FSEOF1	FS time buffer register		255
0x7E	OTG_LSEOF1	LS time buffer register		255
Target address registers				
0x80	OTG_TXFAD0	Tx function address register 0	Y	256
0x82	OTG_TXHAD0	Tx hub address register 0	Y	256
0x83	OTG_TXHP0	Tx hub port register 0	Y	256
0x84	OTG_RXFAD0	Rx function address register 0	Y	256
0x86	OTG_RXHAD0	Rx hub address register 0	Y	256
0x87	OTG_RXHP0	Rx hub port register 0	Y	256
0x88	OTG_TXFAD1	Tx function address register 1	Y	256
0x8A	OTG_TXHAD1	Tx hub address register 1	Y	256
0x8B	OTG_TXHP1	Tx hub port register 1	Y	256
0x8C	OTG_RXFAD1	Rx function address register 1	Y	256
0x8E	OTG_RXHAD1	Rx hub address register 1	Y	256
0x8F	OTG_RXHP1	Rx hub port register 1	Y	256
0x90	OTG_TXFAD2	Tx function address register 2	Y	256
0x92	OTG_TXHAD2	Tx hub address register 2	Y	256
0x93	OTG_TXHP2	Tx hub port register 2	Y	256
0x94	OTG_RXFAD2	Rx function address register 2	Y	256
0x96	OTG_RXHAD2	Rx hub address register 2	Y	256
0x97	OTG_RXHP2	Rx hub port register 2	Y	256

Table 42. USBOTG register list (continued)

Offset	Register name	Description	Host(1)	Page
0x98	OTG_TXFAD3	Tx function address register 3	Y	256
0x9A	OTG_TXHAD3	Tx hub address register 3	Y	256
0x9B	OTG_TXHP3	Tx hub port register 3	Y	256
0x9C	OTG_RXFAD3	Rx function address register 3	Y	256
0x9E	OTG_RXHAD3	Rx hub address register 3	Y	256
0x9F	OTG_RXHP3	Rx hub port register 3	Y	256
0xA0	OTG_TXFAD4	Tx function address register 4	Y	256
0xA2	OTG_TXHAD4	Tx hub address register 4	Y	256
0xA3	OTG_TXHP4	Tx hub port register 4	Y	256
0xA4	OTG_RXFAD4	Rx function address register 4	Y	256
0xA6	OTG_RXHAD4	Rx hub address register 4	Y	256
0xA7	OTG_RXHP4	Rx hub port register 4	Y	256
0xA8	OTG_TXFAD5	Tx function address register 5	Y	256
0xAA	OTG_TXHAD5	Tx hub address register 5	Y	256
0xAB	OTG_TXHP5	Tx hub port register 5	Y	256
0xAC	OTG_RXFAD5	Rx function address register 5	Y	256
0xAE	OTG_RXHAD5	Rx hub address register 5	Y	256
0xAF	OTG_RXHP5	Rx hub port register 5	Y	256
0xB0	OTG_TXFAD6	Tx function address register 6	Y	256
0xB2	OTG_TXHAD6	Tx hub address register 6	Y	256
0xB3	OTG_TXHP6	Tx hub port register 6	Y	256
0xB4	OTG_RXFAD6	Rx function address register 6	Y	256
0xB6	OTG_RXHAD6	Rx hub address register 6	Y	256
0xB7	OTG_RXHP6	Rx hub port register 6	Y	256
0xB8	OTG_TXFAD7	Tx function address register 7	Y	256
0xBA	OTG_TXHAD7	Tx hub address register 7	Y	256
0xBB	OTG_TXHP7	Tx hub port register 7	Y	256
0xBC	OTG_RXFAD7	Rx function address register 7	Y	256
0xBE	OTG_RXHAD7	Rx hub address register 7	Y	256
0xBF	OTG_RXHP7	Rx hub port register 7	Y	256
0xC0	OTG_TXFAD8	Tx function address register 8	Y	256
0xC2	OTG_TXHAD8	Tx hub address register 8	Y	256
0xC3	OTG_TXHP8	Tx hub port register 8	Y	256
0xC4	OTG_RXFAD8	Rx function address register 8	Y	256
0xC6	OTG_RXHAD8	Rx hub address register 8	Y	256

Table 42. USBOTG register list (continued)

Offset	Register name	Description	Host(1)	Page
0xC7	OTG_RXHP8	Rx hub port register 8	Y	256
0xC8	OTG_TXFAD9	Tx function address register 9	Y	256
0xCA	OTG_TXHAD9	Tx hub address register 9	Y	256
0xCB	OTG_TXHP9	Tx hub port register 9	Y	256
0xCC	OTG_RXFAD9	Rx function address register 9	Y	256
0xCE	OTG_RXHAD9	Rx hub address register 9	Y	256
0xCF	OTG_RXHP9	Rx hub port register 9	Y	256
0xD0	OTG_TXFAD10	Tx function address register 10	Y	256
0xD2	OTG_TXHAD10	Tx hub address register 10	Y	256
0xD3	OTG_TXHP10	Tx hub port register 10	Y	256
0xD4	OTG_RXFAD10	Rx function address register 10	Y	256
0xD6	OTG_RXHAD10	Rx hub address register 10	Y	256
0xD7	OTG_RXHP10	Rx hub port register 10	Y	256
0xD8	OTG_TXFAD11	Tx function address register 11	Y	256
0xDA	OTG_TXHAD11	Tx hub address register 11	Y	256
0xDB	OTG_TXHP11	Tx hub port register 11	Y	256
0xDC	OTG_RXFAD11	Rx function address register 11	Y	256
0xDE	OTG_RXHAD11	Rx hub address register 11	Y	256
0xDF	OTG_RXHP11	Rx hub port register 11	Y	256
0xE0	OTG_TXFAD12	Tx function address register 12	Y	256
0xE2	OTG_TXHAD12	Tx hub address register 12	Y	256
0xE3	OTG_TXHP12	Tx hub port register 12	Y	256
0xE4	OTG_RXFAD12	Rx function address register 12	Y	256
0xE6	OTG_RXHAD12	Rx hub address register 12	Y	256
0xE7	OTG_RXHP12	Rx hub port register 12	Y	256
0xE8	OTG_TXFAD13	Tx function address register 13	Y	256
0xEA	OTG_TXHAD13	Tx hub address register 13	Y	256

Table 42. USBOTG register list (continued)

Offset	Register name	Description	Host(1)	Page
0xEB	OTG_TXHP13	Tx hub port register 13	Y	256
0xEC	OTG_RXFAD13	Rx function address register 13	Y	256
0xEE	OTG_RXHAD13	Rx hub address register 13	Y	256
0xEF	OTG_RXHP13	Rx hub port register 13	Y	256
0xF0	OTG_TXFAD14	Tx function address register 14	Y	256
0xF2	OTG_TXHAD14	Tx hub address register 14	Y	256
0xF3	OTG_TXHP14	Tx hub port register 14	Y	256
0xF4	OTG_RXFAD14	Rx function address register 14	Y	256
0xF6	OTG_RXHAD14	Rx hub address register 14	Y	256
0xF7	OTG_RXHP14	Rx hub port register 14	Y	256
0xF8	OTG_TXFAD15	Tx function address register 15	Y	256
0xFA	OTG_TXHAD15	Tx hub address register 15	Y	256
0xFB	OTG_TXHP15	Tx hub port register 15	Y	256
0xFC	OTG_RXFAD15	Rx function address register 15	Y	256
0xFE	OTG_RXHAD15	Rx hub address register 15	Y	256
0xFF	OTG_RXHP15	Rx hub port register 15	Y	256
EP0 registers				
0x102	OTG_CSR0	Control/status register for EP0		360
0x108	OTG_COUNT0	Bytes received counter for EP0		368
0x10A	OTG_TYPE0	Speed control register for EP0	Y	368
0x10B	OTG_NAKLIMIT0	NAK response timeout register	Y	368
0x10F	OTG_CFD	Core configuration register		367
EPy (y = 1 to 15) registers				
0x1y0	OTG_TXMAXPy	Max packet size register (Tx EPs)		236
0x1y2	OTG_TXCSRy	Control/status register (Tx EPs)		238
0x1y4	OTG_RXMAXPy	Max packet size register (Rx EPs)		239
0x1y6	OTG_RXCSRy	Control/status register (Rx EPs)		241
0x1y8	OTG_RXCNTy	Bytes received counter (Rx EPs)		243
0x1yA	OTG_TXTYPEy	Transaction control register (Tx EPs)	Y	243
0x1yB	OTG_TXINTVy	Polling interval register (Tx EPs)	Y	244
0x1yC	OTG_RXTYPEy	Transaction control register (Rx EPs)	Y	244
0x1yD	OTG_RXINTVy	Polling interval register (Rx EPs)	Y	245
Top-level control and status registers				
0x200	OTG_DMASEL	DMA selector register		257
0x204	OTG_TOPCTRL	Top control register		257

1. These registers are **only** available in host mode.

20.3 L2CC register descriptions

OTG_FADDR OTG function address register

7	6	5	4	3	2	1	0
0	FUNC_ADDR						
R	RW						

Address: L2CCBaseAddress + 0x00

Reset: 0x00

Description: This register is written with the 7-bit address of the peripheral part of the transaction. When the core is being used in device mode (*OTG_DEVCTL*.D2 = 0), it is written with the address received through a SET_ADDRESS command, which is then used for decoding the function address in subsequent token packets.

This register only applies when the core is in device mode. In host mode it has no affect.

FUNC_ADDR The function address.

OTG_PWR OTG power register

7	6	5	4	3	2	1	0
ISO_UP	SFCN	HSEN	HSMD	RST	RSM	SUSP	SUSPM
RW	RW	RW	RW	RW	RW	RW	RW

Address: L2CCBaseAddress + 0x01

Reset: 0x20

Description: Controls suspend and resume signaling and some basic operational aspects of the OTG controller.

ISO_UP ISO update. When set by the CPU, the core waits for an SOF token from the time Tx-packet-ready is set before sending the packet. If an IN token is received before an SOF token, then a zero length data packet is sent. Only valid in peripheral mode. This bit only affects endpoints performing isochronous transfers.

SFCN Soft connect. If the soft connect/disconnect feature is enabled, then the USB D+ and D- lines are enabled when this bit is set by the CPU and are tri-stated when this bit is cleared by the CPU. Only valid in peripheral mode.

HSEN High-speed enable. When set by the CPU, the core negotiates for high-speed mode when the device is reset by the hub. If not set, the device only operates in full-speed mode.

HSMD High-speed mode. When set, this read-only bit indicates that high-speed mode is successfully negotiated during USB reset. In peripheral mode, this bit becomes valid when USB reset completes (as indicated by the USB reset interrupt). In host mode, this bit becomes valid when bit RST is cleared. The value remains valid for the duration of the session.

RST Reset is set when reset signaling is present on the bus. This bit is read/write from the CPU in host mode but read-only in peripheral mode.

RSM Resume is set by the CPU to generate resume signaling when the function is in suspend mode. The CPU should clear this bit after 10 ms (a maximum of 15 ms) to end resume signaling. In host mode, this bit is automatically set when resume signaling from the target is detected while the core is suspended.

SUSP Suspend mode. In host mode, this bit is set by the CPU to enter suspend mode. In peripheral mode, this bit is set on entry into suspend mode. It is cleared when the CPU reads the interrupt register, or sets bit RSM.

SUSPM Enable suspend mode is set by the CPU to enable the SUSPENDM output.

OTG_INTTX OTG Tx interrupt register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP15	EP14	EP13	EP12	EP11	EP10	EP9	EP8	EP7	EP6	EP5	EP5	EP3	EP2	EP1	EP0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: USBOTGBaseAddress + 0x02

Reset: 0x0000

Description: Indicates which endpoints generated the Tx interrupt. Bits relating to endpoints that have not been configured always return 0. All active endpoint interrupts are cleared when this register is read.

EP[0:15] EPx Tx interrupt. Indicates that the transmit interrupt was received from this endpoint.

OTG_INTRX OTG Rx interrupt register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP15	EP14	EP13	EP12	EP11	EP10	EP9	EP8	EP7	EP6	EP5	EP4	EP3	EP2	EP1	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: USBOTGBaseAddress + 0x04

Reset: 0x0000

Description: Indicates which interrupts are currently active for Rx endpoints. Bits relating to endpoints that have not been configured always return 0. All active interrupts are cleared when this register is read.

EP[0:15] EPx Rx interrupt signals the receive interrupt received from this endpoint.

OTG_INTTXEN OTG Tx interrupt enable register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP15	EP14	EP13	EP12	EP11	EP10	EP9	EP8	EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: USBOTGBaseAddress + 0x06

Reset: 0xFFFF

Description: Provides enable bits for the interrupts in *OTG_INTTX*. When a bit is set to 1, the CPU interrupt on the Transmit event from the corresponding endpoint is asserted. If a bit is set to 0, the interrupt bit in *OTG_INTTX* is still set but the CPU interrupt is not asserted. All interrupts are allowed after reset.

EP[0:15] EP x Tx interrupt mask.
0: masks the transmit interrupt from endpoint x 1: allows the interrupt

OTG_INTRXEN OTG Rx interrupt enable register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EP15	EP14	EP13	EP12	EP11	EP10	EP9	EP8	EP7	EP6	EP5	EP4	EP3	EP2	EP1	0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: USBOTGBaseAddress + 0x08

Reset: 0xFFFE

Description: Provides enable bits for the interrupts in *OTG_INTRX*. When a bit is set to 1, the CPU interrupt on the receive event from the corresponding endpoint is asserted. If a bit is set to 0, the interrupt bit in OTG_INTRX is still set but the CPU interrupt is not asserted. All interrupts are allowed after reset.

EP[1:15] EP x Rx interrupt mask.
0: masks the transmit interrupt from endpoint x 1: allows the interrupt

0 Reserved. Read returns 0.

OTG_INTUSB Common USB interrupt register

7	6	5	4	3	2	1	0
VBE	SREQ	DISCON	CONN	SOF	RST	RES	SUSP
R	R	R	R	R	R	R	R

Address: USBOTGBaseAddress + 0x0A

Reset: 0x00

Description: Indicates active USB interrupts. All active interrupts are cleared when this register is read.

VBE VBUS error is set when VBus drops below the VBus valid threshold during a session.
Only valid in peripheral mode.

SREQ Session request is set when session request signal has been detected.
Only valid when the core is an A-device.

DISCON Disconnect. Peripheral: Set when a session ends.
Host: Set when a device disconnect is detected (HOSTDISCON going high).

CONN Connect is set when a device connection is detected (HOSTDISCON signal going low).
Only valid in host mode.

SOF Start of frame is set when a new frame starts.

RST Reset/Babble.
Peripheral: Set when reset signaling is detected on the USB.
Host: Set when babble condition is detected.

RES Resume is set when resume signaling is detected on the bus while the core is in suspend mode.

SUSP Suspend. Set when suspend signaling is detected on the bus. Only valid in peripheral mode.

OTG_INTUSBEN Common USB interrupt enable register

7	6	5	4	3	2	1	0
VBEN	SREQN	DSCEN	CONEN	SOFEN	RSTEN	RESEN	SSPEN
R	R	R	R	R	R	R	R

Address: USBOTGBaseAddress + 0x0B

Reset: 0x06

Description: Provides interrupt enable bits for each of the interrupts in *OTG_INTUSB*. Zero disables a corresponding interrupt and a one enables it.

VBEN VBUS error enable. Enables the VBUS interrupt bit in OTG_INTUSB.

SREQN Session request enable. Enables the SREQ interrupt bit in OTG_INTUSB.

DSCEN Disconnect enable. Enables the DISCON interrupt bit in OTG_INTUSB.

CONEN Connect enable. Enables the CONN interrupt bit in OTG_INTUSB.

SOFEN Start of frame enable. Enables the SOF interrupt bit in OTG_INTUSB.

RSTEN Reset/babble enable. Enables the RST interrupt bit in OTG_INTUSB.

RESEN Resume enable. Enables the RES interrupt bit in OTG_INTUSB.

SSPEN Suspend enable. Enables the SUSP interrupt bit in OTG_INTUSB.

OTG_FMNO OTG frame number register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	FMNO										
R	R	R	R	R	R										

Address: USBOTGBaseAddress + 0x0C

Reset: 0x0000

Description: Holds the last received frame number.

FMNO Frame number. Indicates the current frame number.

0 Reserved. Read returns 0.

OTG_INDX OTG index register

7	6	5	4	3	2	1	0
0	0	0	0	EPNO			
R	R	R	R	RW			

Address: USBOTGBaseAddress + 0x0E

Reset: 0x00

Description: Determines which endpoint control/status registers are accessed. Each Tx endpoint and each Rx endpoint has its own set of control/status registers located between address offsets 0x100 and 0x1FF. In addition one set of Tx control/status and one set of Rx control/status registers appear at 0x10 and 0x19.

Before accessing control/status registers at endpoint offset 0x10 to 0x19, the endpoint number must be written to this register to ensure that the correct registers appear in the memory map.

EPNO Endpoint number. Programs the current active endpoint.

0 Reserved. Read returns 0.

OTG_TXMAXP OTG Tx max packet size register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		MULT								MAXP					
		RW								RW					

Address: USBOTGBaseAddress + 0x10

Reset: 0x0000

Description: Provides control and status bits for transfers through the currently-selected Tx endpoint. There is one *OTG_TXMAXP* register for each configured Tx endpoint (not including Endpoint 0). The functionality of this register depends on whether the core is acting as a peripheral or as a host.

The register includes either 2 or 5 more bits that define a multiplier MULT which is equal to one more than the value recorded.

In the case of bulk endpoints with the packet splitting option enabled, MULT can be up to 32 and defines the maximum number of USB packets of the specified payload into which a single data packet placed in the FIFO should be split, prior to transfer. (If the packet splitting option is not enabled, D15 D13 is not implemented and D12 D11(if included) is ignored.) The data

packet must be an exact multiple of the payload specified by bits 10:0, which itself must be either 8, 16, 32, 64 or (in the case of high speed transfers) 512 bytes.

For isochronous endpoints operating in high-speed mode, MULT may only be 2 or 3 (corresponding to bit 11 set or bit 12 set, respectively) and it specifies the maximum number of transactions that can take place in a single microframe. If either bit 11 or bit 12 is non-zero, the core automatically splits any data packet written to the FIFO into two or three USB packets, each containing the specified payload (or less). The maximum payload for each transaction is 1024 bytes, so this allows up to 3072 bytes to be transmitted in each microframe. (For isochronous transfers in full-speed mode, or if high-bandwidth is not enabled, bits 11 and 12 are ignored.) The value written to MAXP (multiplied by MULT in the case of high-bandwidth isochronous transfers) must match the value given in the wMaxPacketSize field of the Standard Endpoint Descriptor for the associated endpoint (see USB Specification Revision 2.0, Chapter 9). A mismatch could cause unexpected results. The total amount of data represented by the value written to this register (specified payload x MULT) must not exceed the FIFO size for the Tx endpoint, and should not exceed half the FIFO size if double-buffering is required. If this register is changed after packets have been sent from the endpoint, the Tx endpoint FIFO should be completely flushed (*OTG_TXCSR*.FF = 1) after writing the new value to this register.

MAXP Maximum payload transmitted. Defines (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for bulk, interrupt and isochronous transfers in full-speed and high-speed operations.

MULT Multiplier. See above.

OTG_CSR0 OTG EP0 control and status register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FF	SSE	SRDY	SND	STE	DE	SNT	TRDY	RRDY
R	R	R	R	R	R	R	RW	RW	RW	RW	R	RW	RW	RW	R

Address: USBOTGBaseAddress + 0x12

Reset: 0x0000

Description: Provides control and status information for Endpoint 0.

FF Flush FIFO is set to flush the next packet to be transmitted from the endpoint Tx FIFO. The FIFO pointer is reset and the TRDY bit (below) is cleared. Flush FIFO has no effect unless Tx packet ready or Rx packet ready is set.

SSE Serviced setup end is set to clear the STE bit. Writing zero has no effect.

SRDY Serviced Rx packet ready is set to clear the RRDY bit. Writing zero has no effect.

SND Send stall is set to terminate the current transaction. The STALL handshake is transmitted and after that this bit is cleared automatically.

STE Setup end is set when a control transaction ends before the data end (DE) bit has been set. An interrupt is generated and the FIFO flushed at this time. The bit is cleared by the software setting the serviced setup end (SSE) bit.

DE Data end. This bit is cleared automatically. Writing zero has no effect. It is set when:
- Setting TRDY for the last data packet.
- Clearing RRDY after unloading the last data packet.
- Setting TRDY for a zero length data packet.

SNT Sent stall is set when a STALL handshake is transmitted. The software should clear this bit.

TRDY Tx packet ready is set after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared.

RRDY Rx packet ready is set when the data packet is received. An interrupt is generated when RRDY is set (unless disabled). This bit can be cleared by software by setting SRDY bit.

OTG_TXCSR OTG Tx control and status register (host)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASET	0	MD	DMR	FDT	DRM	DWE	DT	NAK	CLR	RXS	STP	FF	ERR	FNE	TRDY
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: USBOTGBaseAddress + 0x12.

Reset: 0x0000

Description: When the core is in host mode, this register provides control and status bits for transfers through the currently-selected Tx endpoint. There is one register for each configured Tx endpoint (not including Endpoint 0 that has a different register *OTG_CSR0*).

ASET Auto set. If the CPU sets this bit, TRDY is automatically set when data of the maximum packet size (value in *OTG_TXMAXP*) is loaded into the Tx FIFO. If a packet of less than the maximum packet size is loaded, then TRDY has to be set manually. This bit should not be set for high-bandwidth Isochronous endpoints.

MD Mode is set to enable the endpoint direction as Tx, and clears the bit to enable it as Rx. This bit has an effect only where the same endpoint FIFO is used for both Tx and Rx transactions.

DMR DMA request enable is set to enable the DMA request for the Tx endpoint.

FDT Force data toggle is set to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received. This can be used by interrupt Tx endpoints that are used to communicate rate feedback for isochronous endpoints.

DRM DMA request mode is set to select DMA request mode 1 and clears it to select DMA request mode 0.

DWE Data toggle write enable is set to enable the current state of the Tx Endpoint data toggle to be written (see DT bit, below). This bit is automatically cleared once the new value is written.

DT Data toggle indicates the current state of the Tx Endpoint data toggle. If DRM is high, this bit may be written with the required setting of the data toggle. If DRM is low, any value written to DT is ignored.

NAK NAK timeout is set when the Tx endpoint is halted following the receipt of NAK responses for longer than the time set as the NAK limit by the *OTG_TXINTV* register. The CPU should clear this bit to allow the endpoint to continue. Valid only for bulk endpoints.

CLR Clear data toggle is set to reset the endpoint data toggle to 0.

RXS Rx stall is set when a STALL handshake is received. The FIFO is flushed and the TRDY bit is cleared (see below). The CPU should clear this bit.

STP Setup packet is set at the same time as the TRDY bit is set, to send a SETUP token instead of an OUT token for the transaction. Setting this bit also clears the data toggle.

FF Flush FIFO is set to flush the next packet to be transmitted from the endpoint Tx FIFO. The FIFO pointer is reset and the TRDY bit (below) is cleared. Flush FIFO has no effect unless Tx-packet-ready is set. If the FIFO is double-buffered, FF may need to be set twice to completely clear the FIFO.

ERR Error is set when three attempts have been made to send a packet and no handshake packet has been received. The CPU should clear this bit. An interrupt is generated when the bit is set. Valid only when the endpoint is operating in bulk or interrupt mode.

FNE FIFO not empty is set when there is at least 1 packet in the Tx FIFO.

TRDY Tx packet ready is set after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared.

OTG_TXCSR OTG Tx control and status register (device)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASET	ISO	MD	DMR	FDT	DRM	0	0	ITX	CLR	STS	SDS	FF	UND	FNE	TRDY
RW	RW	RW	RW	RW	RW	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: USBOTGBaseAddress + 0x12.

Reset: 0x0000

Description: When the core is in device mode, this register provides control and status bits for transfers through the currently-selected Tx endpoint. There is one register for each configured Tx endpoint (not including Endpoint 0 that has a different register, *OTG_CSR0*).

ASET Auto set. If the CPU sets this bit, TRDY is automatically set when data of the maximum packet size (value in *OTG_TXMAXP*) is loaded into the Tx FIFO. If a packet of less than the maximum packet size is loaded, then TRDY has to be set manually. This bit should not be set for high-bandwidth Isochronous endpoints.

ISO Isochronous transfers. The CPU sets this bit to enable the Tx endpoint for isochronous transfers, and clears it to enable the Tx endpoint for bulk or interrupt transfers.

MD Mode. Has an effect only where the same endpoint FIFO is used for both Tx and Rx transactions.
0: enable endpoint direction as Rx 1: enable endpoint direction as Tx

DMR DMA request enable is set to enable the DMA request for the Tx endpoint.

FDT Force data toggle is set to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received. This can be used by interrupt Tx endpoints that are used to communicate rate feedback for isochronous endpoints.

DRM DMA request mode is set to select DMA request mode 1 and clears it to select DMA request mode 0.

ITX Incomplete. When the endpoint is being used for high-bandwidth isochronous/interrupt transfers, this bit is set to indicate where a large packet has been split into 2 or 3 packets for transmission but insufficient IN tokens have been received to send all the parts. In anything other than a high-bandwidth transfer, this bit always returns zero.

CLR Clear data toggle is set to reset the endpoint data toggle to 0.

STS Sent stall is set when a stall handshake is transmitted. The FIFO is flushed and the Tx-packer-ready bit is cleared (see below). The CPU should clear this bit.

SDS Send stall is set by the CPU to issue a stall handshake to an IN token. The CPU clears this bit to terminate the stall condition. This bit has no effect where the endpoint is being used for isochronous transfers.

FF Flush FIFO is set to flush the next packet to be transmitted from the endpoint Tx FIFO. The FIFO pointer is reset and the TRDY bit (below) is cleared. Flush FIFO has no effect unless Tx-packet-ready is set. If the FIFO is double-buffered, FF may need to be set twice to completely clear the FIFO.

UND Underrun is set if an IN token is received when Tx-packet-ready is not set. The CPU should clear this bit.

FNE FIFO not empty is set when there is at least 1 packet in the Tx FIFO.

TRDY Tx packet ready is set after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared.

OTG_RXMAXP OTG Rx max packet size register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		MULT								MAXP					
		RW								RW					

Address: USBOTGBaseAddress + 0x14.

Reset: 0x0000

Description: Defines the maximum amount of data that can be transferred through the selected Rx endpoint in a single operation. There is one *OTG_RXMAXP* register for each Rx endpoint (except Endpoint 0).

The register includes either 2 or 5 further bits that define a multiplier MULT which is equal to one more than the value recorded. For bulk endpoints with the packet splitting option enabled, the MULT can be up to 32 and defines the number of USB packets of the specified payload which are to be amalgamated into a single data packet within the FIFO. (If the packet splitting option is not enabled, D15 D13 is not implemented and D12 D11 (if included) is ignored.)

For isochronous endpoints operating in high-speed mode and with the high-bandwidth option enabled, MULT may only be either 2 or 3 (corresponding to bit 11 set or bit 12 set, respectively) and it specifies the maximum number of such transactions that can take place in a single microframe. If either bit 11 or bit 12 is non-zero, the core automatically combines the separate USB packets received in any microframe into a single packet within the Rx FIFO. The maximum payload for each transaction is 1024 bytes, so this allows up to 3072 bytes to be received in each microframe. (For isochronous transfers in full-speed mode or if high-bandwidth is not enabled, bits 11 and 12 are ignored.) The value written to MAXP field (multiplied by MULT in the case of high-bandwidth isochronous transfers) must match the value given in the wMaxPacketSize field of the Standard Endpoint Descriptor for the associated endpoint (see USB Specification Revision 2.0, Chapter 9). A mismatch could cause unexpected results. The total amount of data

represented by the value written to this register (specified payload x MULT) must not exceed the FIFO size for the Rx endpoint, and should not exceed half the FIFO size if double-buffering is required.

MAXP Maximum payload transmitted defines (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for bulk, interrupt and isochronous transfers in full-speed and high-speed operations.

MULT Multiplier. See above.

OTG_RXCSR OTG Rx control/status register (device)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASET	ISO	DMR	DNY	DMD	0	0	IRX	CLR	SNT	SND	FF	ERR	OR	FFUL	RRDY
RW	RW	RW	RW	RW	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: USBOTGBaseAddress + 0x16.

Reset: 0x0000

Description: In peripheral mode, this register provides control and status bits for transfers through the currently-selected Rx endpoint. There is one OTG_RXCSR register for each configured Rx endpoint (not including Endpoint 0). The register has different functionality depending on whether the OTG controller is acting as a peripheral or as a host.

ASET Auto Set. If the CPU sets this bit then the RRDY bit is automatically cleared when a packet of (*OTG_RXMAXP*) bytes has been unloaded from the Rx FIFO. When packets of less than the maximum packet size are unloaded, RRDY has to be cleared manually. This bit should not be set for high-bandwidth Isochronous endpoints.

ISO ISO is set to enable the Rx endpoint for Isochronous transfers, and clears it to enable the Rx endpoint for bulk/interrupt transfers.

DMR DMA request enable is set to enable the DMA request for the Rx endpoint.

DNY Disable NYET is set to disable the sending of NYET handshakes. When set, all successfully received Rx packets are ACK'd including at the point at which the FIFO becomes full. This bit only has an effect in high-speed mode, when it should be set for all interrupt endpoints.

DMD DMA request mode is set to select DMA request mode 1 and clears it to select DMA request mode 0.

IRX Incomplete Rx is set in a high-bandwidth isochronous transfer if the packet in the Rx FIFO is incomplete because parts of the data were not received. It is cleared when RRDY is cleared. In anything other than a high-bandwidth isochronous transfer, this bit always returns 0.

CLR Clear data toggle is set to reset the endpoint data toggle to 0.

SNT Sent stall is set when a STALL handshake is transmitted. The CPU should clear this bit.

SND Send stall is set to issue a STALL handshake. The CPU clears this bit to terminate the stall condition. This bit has no effect where the endpoint is being used for isochronous transfers.

FF Flush FIFO is set to flush the next packet to be transmitted from the endpoint Tx FIFO. The FIFO pointer is reset and the RRDY bit is cleared. FF bit has no effect unless RRDY is set. If the FIFO is double-buffered, FF may need to be set twice to completely clear the FIFO.

FFUL FIFO full is set when no more packets can be loaded into the Rx FIFO.

ERR Data error is set when RRDY is set if the data packet has a CRC or bit-stuff error. It is cleared when RRDY is cleared. It is only valid when the endpoint is operating in ISO mode. In bulk mode it always returns zero.

OR Overrun is set if an OUT packet cannot be loaded into the Rx FIFO. The CPU should clear this bit. This bit is only valid when the endpoint is operating in ISO mode. In bulk mode it always returns zero.

RRDY Rx packet ready is set when a data packet has been received. The CPU should clear this bit when the packet has been unloaded from the Rx FIFO. An interrupt is generated when the bit is set.

OTG_RXCSR OTG Rx control/status register (host)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASET	ARQ	DMR	DNY	DMD	DWE	DT	IRX	CLR	RS	RPK	FF	DER	ER	FFUL	RRDY
RW	RW	RW	RW	RW	R	R	R	RW	RW	RW	RW	RW	RW	R	RW

Address: USBOTGBaseAddress + 0x16.

Reset: 0x0000

Description: This section describes the functionality of the OTG_RXCSR in host mode.

ASET Auto set. If the CPU sets this bit then the RRDY bit is automatically cleared when a packet of (*OTG_RXMAXP*) bytes has been unloaded from the Rx FIFO. When packets of less than the maximum packet size are unloaded, RRDY has to be cleared manually. This bit should not be set for high-bandwidth Isochronous endpoints.

ARQ Auto request. If the CPU sets this bit, the RPK bit is automatically set when the RRDY bit is cleared.

DMR DMA request enable is set to enable the DMA request for the Rx endpoint.

DNY Disable NYET is set to disable the sending of NYET handshakes thus all successfully received Rx packets are ACK'd including at the point at which the FIFO becomes full. This bit only has any effect in high-speed mode, in which mode it should be set for all Interrupt endpoints.

DMD DMA request mode is set to select DMA request mode 1 and clears it to select DMA request mode 0.

DWE Data toggle write enable is set to enable the current state of the Endpoint 0 data toggle to be written (see bit DT). This bit is automatically cleared once the new value is written.

DT Data toggle indicates the current state of the Endpoint 0 data toggle. If DWE is high, this bit may be written with the required setting of the data toggle. If DWE is low, any value written to DT is ignored.

IRX Incomplete Rx is set in a high-bandwidth isochronous transfer if the packet received is incomplete. It is cleared when RRDY is cleared. In anything other than a high-bandwidth isochronous transfer, this bit always returns 0. If USB protocols are followed correctly, this bit should never be set. The bit becoming set indicates a failure of the associated peripheral device to behave correctly.

CLR Clear data toggle. When a STALL handshake is received, this bit is set and an interrupt is generated. The CPU should clear this bit.

RS Rx stall. When a STALL handshake is received, this bit is set and an interrupt is generated. The CPU clears this bit.

RPK Request packet is set to request an IN transaction. It is cleared when RRDY is set.

FF Flush FIFO is set to flush the next packet to be transmitted from the endpoint Rx FIFO. The FIFO pointer is reset and the RRDY bit is cleared. FF bit has no effect unless RRDY is set. If the FIFO is double-buffered, FF may need to be set twice to completely clear the FIFO.

DER Data error/NAK time-out. When operating in ISO mode, this bit is set when RRDY is set if the data packet has a CRC or bit-stuff error and cleared when RRDY is cleared. In bulk mode, this bit is set when the Rx endpoint is halted following the receipt of NAK responses for longer than the time set as the NAK limit by the Rx Interval register. The CPU clears this bit to allow the endpoint to continue.

ER Error. The USB sets this bit when three attempts have been made to receive a packet and no data packet has been received. The CPU clears this bit. An interrupt is generated when the bit is set. It is only valid when the Tx endpoint is operating in bulk or interrupt mode. In ISO mode it always returns zero.

FFUL FIFO full is set when no more packets can be loaded into the Rx FIFO.

RRDY Rx packet ready is set when a data packet has been received. The CPU clears this bit when the packet has been unloaded from the Rx FIFO. An interrupt is generated when the bit is set.

OTG_RXCNT OTG Rx bytes received counter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	RXCNT												
R	R	R	R												

Address: USBOTGBaseAddress + 0x18.

Reset: 0x0000

Description: This 13-bit register holds the number of received data bytes in the packet in the Rx FIFO. The value returned changes as the FIFO is unloaded and is only valid while *OTG_RXCSR*.RRDY is set.

RXCNT Endpoint Rx count is the number of bytes received in Rx FIFO. It is a 7-bit field for Endpoint 0.

0 Reserved. Read returns 0

OTG_TXTYPE OTG Tx transaction control register (host only)

7	6	5	4	3	2	1	0
SPEED		PROT			EP		
RW		RW			RW		

Address: USBOTGBaseAddress + 0x1A.

Reset: 0x00

Description: Contains the endpoint number to be targeted by the endpoint, the transaction protocol to use for the currently-selected Tx endpoint and its operating speed. It is only active

in host mode. There is one OTG_TXTYPE register for each configured Tx endpoint (except Endpoint 0, which has its own *OTG_TYPE0* register).

SPEED Operating speed of the target device:
00: unused 01: high
10: full 11: low

PROT Protocol selects the required protocol for the Tx endpoint:
00: control 01: isochronous
10: bulk 11: interrupt

EP Target endpoint number. The CPU should set this value to the endpoint number contained in the Tx endpoint descriptor returned to the OTG Controller during device enumeration.

OTG_TXINTV OTG Tx polling interval register (host only)

7	6	5	4	3	2	1	0
TXPI							
RW							

Address: USBOTGBaseAddress + 0x1B.

Reset: 0x00

Description: Defines the polling interval for the currently-selected Tx endpoint for Interrupt and Isochronous transfers. There is one OTG_TXINTV register for each configured Tx endpoint (except Endpoint 0).

TXPI Tx polling interval/NAK limit defines the polling interval for the currently-selected Tx endpoint for interrupt and isochronous transfers. For bulk mode, this field sets the number of frames/ microframes after which the endpoint should time-out on receiving a stream of NAK responses.

The following table defines the valid TXPI values for different modes:

Table 43 •Valid TXPI values for different modes

Type	Speed	Valid values	Interpretation
Interrupt	Low or full	1-255	Polling interval is TXPI frames.
	High	1-16	Polling interval is 2^{TXPI-1} microframes.
ISO	Full or high	1-16	Polling interval is 2^{TXPI-1} frames/microframes.
Bulk	Full or high	2-16	NAK Limit is 2^{TXPI-1} frames/microframes. A value of 0 or 1 disables the NAK time-out function.

OTG_RXTYPE OTG Rx transaction control register (host only)

7	6	5	4	3	2	1	0
SPEED		PROT			EP		
RW		RW			RW		

Address: USBOTGBaseAddress + 0x1C.

Reset: 0x00

Description: Contains the endpoint number to be targeted by the endpoint, the transaction protocol to use for the currently-selected Rx endpoint and its operating speed. There is one OTG_RXTYPE register for each configured Rx endpoint (except Endpoint 0).

SPEED Operating of the target device:
00: unused 01: high
10: full 11: low

PROT Protocol selects the required protocol for the Tx endpoint:
00: control 01: isochronous
10: bulk 11: interrupt

EP Target endpoint number. The CPU sets this value to the endpoint number contained in the Tx endpoint descriptor returned to the OTG controller during device enumeration.

OTG_RXINTV OTG Rx polling interval register (host only)

7	6	5	4	3	2	1	0
RXPI							
RW							

Address: USBOTGBaseAddress + 0x1D.

Reset: 0x00

Description: Defines (for interrupt and isochronous transfers) the polling interval for the currently selected Rx endpoint. For bulk endpoints, this register sets the number of frames/microframes after which the endpoint should time-out on receiving a stream of NAK responses. There is a OTG_RXINTV register for each configured Rx endpoint (except Endpoint 0).

RXPI Rx polling interval/NAK limit defines the polling interval for the currently-selected Rx endpoint for interrupt and isochronous transfers. In bulk mode, this field sets the number of frames/microframes after which the endpoint should time-out on receiving a stream of NAK responses.

The following table defines the valid TXPI values for different modes:

Type	Speed	Valid values	Interpretation
Interrupt	Low or full	1-255	Polling interval is RXPI frames.
	High	1-16	Polling interval is 2^{RXPI-1} microframes.
ISO	Full or high	1-16	Polling interval is 2^{RXPI-1} frames/microframes.
Bulk	Full or high	2-16	NAK Limit is 2^{RXPI-1} frames/microframes. A value of 0 or 1 disables the NAK timeout function.

OTG_CFD OTG core configuration register

7	6	5	4	3	2	1	0
MPRXE	MPTXE	BE	HBRXE	HBTXE	DYNF	SC	UI
R	R	R	R	R	R	R	R

Address: USBOTGBaseAddress + 0x1F

Reset: 0xDE

Description: Returns selected core configuration information. This register is selected for access when *OTG_INDX* points to Endpoint 0.

MPRXE When set, automatic amalgamation of bulk packets is selected.

MPTXE When set, automatic splitting of bulk packets is selected.

BE Big endian. When set, indicates big endian ordering.

HBRXE High-bandwidth ISO support. When set, indicates high-bandwidth Rx ISO endpoint support selected.

HBTXE High-bandwidth ISO support. When set, indicates high-bandwidth Tx ISO endpoint support selected.

DYNF Dynamic FIFO sizing. When set, indicates dynamic FIFO sizing option selected.

SC Soft connect. When set, indicates soft connect/disconnect option selected.

UDI UTMI data width indicates selected UTMI+ data width:
0: 8 bits
1: 16 bits

OTG_COUNT0 OTG EP0 bytes received counter

7	6	5	4	3	2	1	0
0				RXCNT0			
R				R			

Address: USBOTGBaseAddress + 0x18

Reset: 0x00

Description: Indicates the number of received data bytes in EP0 Rx FIFO. This register is selected for access when *OTG_INDX* points to Endpoint 0.

RXCNT0 EP0 bytes received.

OTG_TYPE0 OTG EP0 speed control register (host only)

7	6	5	4	3	2	1	0
SPEED		0	0	0	0	0	0
RW		R	R	R	R	R	R

Address: USBOTGBaseAddress + 0x1A

Reset: 0x00

Description: Controls the core operating speed. This register is selected for access when *OTG_INDX* points to Endpoint 0.

SPEED Operating speed.
00: unused
01: high-speed
10: full-speed
11: low-speed

OTG_NAKLIMIT0 OTG EP0 NAK response register (host only)

7	6	5	4	3	2	1	0
0	0	0			NAKLIMIT		
R	R	R			RW		

Address: USBOTGBaseAddress + 0x1B

Reset: 0x00

Description: Sets the number of frames (microframes for HS) after which the EP0 should time-out. This register is selected for access when *OTG_INDX* points to Endpoint 0.

NAKLIMIT Frames to NAK.

OTG_DEVCTL OTG device control register

7	6	5	4	3	2	1	0
BDEV	FSDEV	LSDEV	VBUS		HOST	HREQ	SESS
R	R	R	R		R	RW	RW

Address: USBOTGBaseAddress + 0x60

Reset: 0x80

Description: Selects the operating mode of the core (peripheral/host) and controls and monitors the USB VBus line.

BDEV B-Device indicates whether the core is operating as an A-Device or a B-Device. Only valid while a session is in progress. If the core is in force host mode (session started with OTG_TM.TFRH = 1) this bit indicates the state of the HOSTDISCON input signal from the transceiver.
0: A-Device 1: B-Device

FSDEV Full speed device is set when a full- or high-speed device has been connected to the port. High-speed devices are distinguished from full-speed by checking for high-speed chirps when the device is reset. Only valid in host mode.

LSDEV Low speed is set when a low-speed device has been connected to the port. Only valid in host mode.

VBUS VBUS encodes the current VBus level as follows:
00: below SessionEnd 01: above SessionEnd, below AValid
10: above AValid, below VBusValid 11: above VBusValid

HOST Host mode is set when the core is acting as a host.

HREQ Host request. When set, the core initiates the host negotiation when suspend mode is entered. It is cleared when host negotiation is completed. (B-Device only)

SESS Session. When operating as an A-Device, this bit is set or cleared by the software to start or end a session. When operating as a B-Device, this bit is set/cleared by the core when a session starts/ends. It may also be set by the software to initiate the SRP. When the core is in suspend mode, the bit may be cleared by the software to perform a software disconnect.

OTG_TXFSZ OTG Tx FIFO size register

7	6	5	4	3	2	1	0
0	0	0	DPB		TXSIZE		
R	R	R	RW		RW		

Address: USBOTGBaseAddress + 0x62

Reset: 0x00

Description: Controls the size of the selected Tx endpoint FIFO.

DPB Double packet buffering defines if double-packet buffering is supported.
0: only single-packet buffering is supported 1: double-packet buffering is supported

TXSIZE Max packet size is the maximum packet size allowed in bytes (before any splitting within the FIFO of bulk/high- bandwidth packets prior to transmission).
0000: 8 0001: 16
0010: 32 0011: 64
0100: 128 0101: 256
0110: 512 0111: 1024
1000: 2048 1001: 4096
If DPB = 0, the Tx FIFO is also this size; if DPB = 1, the Tx FIFO is twice this size.

OTG_RXFSZ OTG Rx FIFO size register

7	6	5	4	3	2	1	0
0	0	0	DPB		RXSIZE		
R	R	R	RW		RW		

Address: USBOTGBaseAddress + 0x63

Reset: 0x00

Description: Controls the size of the selected Rx endpoint FIFO.

DPB Double packet buffering defines whether double-packet buffering supported. 0: only single-packet buffering is supported. 1: double-packet buffering is supported.

RXSIZE Max packet size is the maximum packet size allowed in bytes (after any combination within the FIFO of bulk/high- bandwidth packets following their reception).
0000: 8 0001: 16
0010: 32 0011: 64
0100: 128 0101: 256
0110: 512 0111: 1024
1000: 2048 1001: 4096
If DPB = 0, the Rx FIFO is also this size; if DPB = 1, the Rx FIFO is twice this size.

OTG_TXFA OTG Tx FIFO address register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0							AD						
R	R	R							RW						

Address: USBOTGBaseAddress + 0x64

Reset: 0x0000

Description: Controls the start address of the selected Tx endpoint FIFO.

AD FIFO start address defines the start address of the endpoint FIFO in units of 8 bytes as follows: 0000: 0000, 0001: 0008, 0002: 0010 through to 1FFF: FFF8.

OTG_RXFA OTG Rx FIFO address register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0							AD						
R	R	R							RW						

Address: USBOTGBaseAddress + 0x66

Reset: 0x0000

Description: Controls the start address of the selected Rx endpoint FIFO.

AD FIFO start address defines the start address of the endpoint FIFO in units of 8 bytes as follows: 0000: 0000, 0001: 0008, 0002: 0010 through to 1FFF: FFF8.

OTG_HWVER OTG hardware version number register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0			VMAJ							VMIN					
R			R							RR					

Address: USBOTGBaseAddress + 0x6C

Reset: 0x0590

Description: Returns information about the version of the core that was implemented.

VMAJ Major version number. Returns 000001. The entire version number is VMAJ.VMIN

VMIN Minor version number. Returns 0110010000.

OTG_UVBCTRL OTG ULPI VBUS control register

7	6	5	4	3	2	1	0
0	0	0	0	0	0	EXTVI	EXTVB
R	R	R	R	R	R	RW	RW

Address: USBOTGBaseAddress + 0x70

Reset: 0x00

Description: ULPI PHYs may use an external charge pump to generate the VBUS voltage, rather than an internal charge pump. This register allows selection of the external charge pump. It also allows this selection to be displayed via an external VBUS indicator.

This register is read back from the PHY clock domain. These bits do not return updated values while the PHY is suspended.

EXTVI Use external VBUS indicator. When 1, selects the use of an external VBUS indicator (overcurrent indicator) in the PHYs VbusState determination.

EXTVB Use external VBUS. When set, this bit selects an external charge pump.

OTG_UCKIT OTG ULPI CarKit control register

7	6	5	4	3	2	1	0
0	0	CKAE	RXCMD	CANCK	ALTINT	CKACT	DISU
R	R	R/Clr	R/Clr	RW	R/Clr	R	RW

Address: USBOTGBaseAddress + 0x71

Reset: 0x00

Description: Provides the basic control needed by ULPI-compatible PHYs when interfacing to in-car CarKit systems.

CKAE CarKit active end. Set by link when CarKitActive is cleared. This bit must be cleared by software. It signifies that the core s (synchronous) USB mode has been entered.

RXCMD RX CMD event. Set by link when a RX CMD has been latched. This bit must be cleared by software.

CANCK Cancel CarKit. Set by software to abort CarKit mode and wake up the PHY. This bit is auto-cleared when the PHY has entered synchronous mode.

ALTINT ALT INT event. Set by link when an ALT_INT event occurs. This bit must be cleared by software.

CKACT CarKit active. Set by link when the (asynchronous) CarKit mode is entered after DIR goes high. It is cleared on the falling edge of DIR.

DISU Disable UTMI. Set and cleared by software to decouple the reconstituted UTMI signals from the USB controller prior to entering CarKit mode.

This bit cannot be cleared while the ULPICarKitControl.CarKitActive bit is set.

OTG_UINTMASK OTG ULPI INT mask register

7	6	5	4	3	2	1	0
0	0	0	0	RXCMD_IE	AEND_IE	ALTINT_IE	REGINT_IEI
R	R	R	R	RW	RW	RW	RW

Address: USBOTGBaseAddress + 0x72

Reset: 0x00

Description: Enables the assertion of the CPU interrupt in response to possible interrupt sources.

RXCMD_IE RX CMD interrupt enable. Assert the CPU interrupt if *OTG_UINTSRC*.RXCMDI is set. To clear the interrupt, the software must clear the *OTG_UCKIT*.RXCMD bit.

AEND_IE Active end interrupt enable. Assert the CPU interrupt if OTG_UINTSRC.AENDI is set. To clear the interrupt, the software must clear the *OTG_UCKIT*.CKAE bit.

ALTINT_IE ALT INT interrupt enable. Assert the CPU interrupt if OTG_UINTSRC.ALTINTI is set. To clear the interrupt, the software must clear the *OTG_UCKIT*.ALTINT bit.

REGINT_IE REG INT interrupt enable. Assert the CPU interrupt if OTG_UINTSRC.REGINTI is set. To clear the interrupt, the software must clear the *OTG_UREGCTRL*.REGCMP bit.

OTG_UINTSRC OTG ULPI INT source register

7	6	5	4	3	2	1	0
0				RXCMDI	AENDI	ALTINTI	REGINTI
R	R	R	R	R	R	R	R

Address: USBOTGBaseAddress + 0x73

Reset: 0x00

Description: Shows the unmasked value of the possible sources of interrupt.

RXCMDI RX CMD interrupt. Asserted if the *OTG_UCKIT*.RXCMD bit is set. To clear the interrupt, the software must clear the OTG_UCKIT.RXCMD bit.

AENDI Active end interrupt. Asserted if the *OTG_UCKIT*.CKAE bit is set. To clear the interrupt, the software must clear the OTG_UCKIT.CKAE bit.

ALTINTI ALT INT interrupt. Asserted if the *OTG_UCKIT*.ALTINT bit is set. To clear the interrupt, the software must clear the OTG_UCKIT.RXCMD bit.

REGINTI REG INT interrupt. Asserted if the *OTG_UREGCTRL*.REGCMP bit is set. To clear the interrupt, the software must clear the OTG_UREGCTRL.REGCMP bit.

OTG_UREGDATA OTG ULPI Reg data register

7	6	5	4	3	2	1	0
REGDATA							
RW							

Address: USBOTGBaseAddress + 0x74

Reset: 0x00

Description: Contains the data associated with register read or writes conducted through the ULPI interface.

REGDATA Register data for PHY access.

OTG_UREGADDR OTG ULPI Reg address register

7	6	5	4	3	2	1	0
REGADDR							
RW							

Address: USBOTGBaseAddress + 0x75

Reset: 0x00

Description: Contains the address of the register being read or written through the ULPI interface.

REGADDR Register address for PHY access.

OTG_UREGCTRL OTG ULPI Reg control register

7	6	5	4	3	2	1	0
0	0	0	0	0	URW	REGCMP	REGREQ
R	R	R	R	R	RW	R/Clr	RW

Address: USBOTGBaseAddress + 0x76

Reset: 0x00

Description: Contains control and status bits relating to the register being read or written through the ULPI interface.

URW ULPI read or write. Set by software for register read access and cleared by software for register write access.

REGCMP Register access complete. Set by the ULPI link when register access is complete. This bit must be cleared by software.

REGREQ Register access request. Set by software to initiate register access. This bit is cleared when the REGCMP bit is set.

OTG_URAWDATA OTG ULPI raw data register (async)

7	6	5	4	3	2	1	0
0	0	0	0	D3	D2	D1	D0
R	R	R	R	R	R	R	R

Address: USBOTGBaseAddress + 0x77

Reset: 0x00

Description: Used in asynchronous modes to sample the ULPI bus and in synchronous mode to store the last RX CMD. It has different formats for asynchronous and synchronous modes.

When one of the PHY's asynchronous modes is selected, the OTG_URAWDATA register is used to indicate the present value of the ULPI bus, latched by any transition on INT (that is, as indicated by the D3 bit).

D3 DATA3. Active high interrupt indication (INT).

D2 DATA2. Single-ended zero (SE0).

D1 DATA1. Differential data (DAT).

D0 DATA0. Active high transmit enable (TXEN).

OTG_URAWDATA OTG ULPI raw data register (sync)

7	6	5	4	3	2	1	0
ALTINT	ID	RXEVENT		VBUSSTATE		LINESTATE	
R	R	R		R		R	

Address: USBOTGBaseAddress + 0x77

Reset: 0x00

Description: When synchronous mode is being used, the OTG_URAWDATA register is used to store the last RX CMD.

ALTINT ALTINT event. Asserted when a non-USB interrupt occurs. In particular, it is required to be set if an unmasked event occurs on any bit of the PHY's CarKit interrupt latch register.

ID ID status. This bit reflects the value of the IDDIG (valid 50 ms after IDPULLUP is asserted).

RXEVENT Encoded UTMI event signals:
00: RxActive = 0, RxError = 0, HostDiscon = 0
01: RxActive = 1, RxError = 0, HostDiscon = 0
10: RxActive = 1, RxError = 1, HostDiscon = 0
11: HostDiscon = 1

VBUSSTATE Encoded VBUS voltage state:
00: VBUS < VB_SESS_END
01: VB_SESS_END <= VBUS < VB_SESS_VLD
10: VB_SESS_VLD <= VBUS < VA_VBUS_VLD
11: VA_VBUS_VLD <= VBUS

LINESTATE UTMI+ line state signals:
00: SE0
01: J-state
10: K-state
11: SE1

OTG_EPINFO OTG EP information register

7	6	5	4	3	2	1	0
	RX_EP				TX_EP		
	R				R		

Address: USBOTGBaseAddress + 0x78

Reset: 0xFF

Description: Stores the number of Tx and Rx endpoints implemented in the design. The OTG HS controller has 16 Tx and 16 Rx endpoints.

RX_EP The number of receive endpoints.

TX_EP The number of transmit endpoints.

OTG_RAMINFO OTG RAM information register

7	6	5	4	3	2	1	0
DMA_CH					RAM_BITS		
R					R		

Address: USBOTGBaseAddress + 0x79

Reset: 0x0C

Description: Provides information about the number of DMA channels and the size of the RAM.

DMA_CH The number of DMA channels.

RAM_BITS The width of the RAM data bus.

OTG_LINKINFO OTG link information register

7	6	5	4	3	2	1	0
WTCON					WTID		
RW					RW		

Address: USBOTGBaseAddress + 0x7A

Reset: 0x5C

Description: Reads and reprograms connect or disconnect, and ID-pullup delays.

WTCON Connect/disconnect delay. Sets the wait to be applied to allow for the user's connect and disconnect filter in units of 533.3 ns (the default setting corresponds to 2.667 µs).
When working in FS interface mode (*OTG_TOPCTRL*.MODE_ULPI is zero), the timer values are different (units of 666.63 ns and default value of 3.33 µs).

WTID ID pullup delay. Sets the delay to be applied from IDPULLUP being asserted to IDDIG being considered valid in units of 4.369 ms (the default setting corresponds to 52.43 ms).
When working in FS Interface mode (*OTG_TOPCTRL*.MODE_ULPI is zero) the timer values are different (units of 5.46 ms and default value of 65.54 ms).

OTG_VPLEN OTG VPLEN register

7	6	5	4	3	2	1	0
			VPLEN				
R	RW	RW	RW	RW	RW	RW	RW

Address: USBOTGBaseAddress + 0x7B

Reset: 0x3C

Description: Sets the duration of the VBUS pulsing charge.

VPLEN VBUS pulse length. Sets the duration of the VBus pulsing charge in units of 546.1 µs (the default setting corresponds to 32.77ms).

When working in FS Interface mode (*OTG_TOPCTRL*. MODE_ULPI is zero) the timer values are different (units of 682.62 µs and default value of 40.96 ms).

OTG_HSEOF1 OTG HS time buffer register

7	6	5	4	3	2	1	0
HS_EOF1							
RW							

Address: USBOTGBaseAddress + 0x7C

Reset: 0x80

Description: Sets the minimum time gap that is to be allowed between the start of the last transaction and the EOF for high-speed transactions.

HS_EOF1 HS time buffer. Sets for high-speed transactions in the time before EOF to stop new transactions beginning (in units of 133.3 ns, default setting corresponds to 17.07 µs).

OTG_FSEOF1 OTG FS time buffer register

7	6	5	4	3	2	1	0
FS_EOF1							
RW							

Address: USBOTGBaseAddress + 0x7D

Reset: 0x77

Description: Sets the minimum time gap that is to be allowed between the start of the last transaction and the EOF for full-speed transactions.

FS_EOF1 FS time buffer. Sets for full-speed transactions in the time before EOF to stop new transactions beginning (in units of 533.3 ns, default setting corresponds to 63.46 µs).

OTG_LSEOF1 OTG LS time buffer register

7	6	5	4	3	2	1	0
LS_EOF1							
RW							

Address: USBOTGBaseAddress + 0x7E

Reset: 0x72

Description: Sets the minimum time gap that is to be allowed between the start of the last transaction and the EOF for low-speed transactions.

LS_EPF1 LS time buffer. Sets for low-speed transactions the time before EOF to stop new transactions beginning (in units of 1.067 µs, default setting corresponds to 121.6 µs).

OTG_TX(RX)FAD[0:15] OTG Tx (Rx) function address register (host only)

7	6	5	4	3	2	1	0
0				ADTF			
R				RW			

Address: USBOTGBaseAddress + 0x80 (or 0x84 for Rx) + 8*y, where y = 0 to 15.

Reset: 0x00

Description: Records the address of the target function that is accessed through the associated endpoint.

ADTF Address of target function. Stores the address of target function to be accessed by endpoint.

OTG_TX(RX)HAD[0:15] OTG Tx (Rx) hub address register (host only)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPDIR S15	GPDIR S14	GPDIR S13	GPDIR S12	GPDIR S11	GPDIR S10	GPDIR S9	GPDIR S8	MLT	HUBAD						
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW						

Address: USBOTGBaseAddress + 0x82 (or 0x86 for Rx) + 8*y, where y = 0 to 15.

Reset: 0x00 (not changed by system (PRESETnot) reset)

Description: These registers, like *OTG_TX(RX)HP[0:15]*, only need to be written when a full- or low-speed device is connected via a high-speed USB 2.0 hub which carries out the necessary transaction translation to convert between high-speed transmission and full-/low-speed transmission.

MLT Multiple translators records whether the hub (connected to the core) has multiple transaction translators.

0: single transaction translator 1: multiple transaction translators

HUBAD Hub address.

OTG_TX(RX)HP[0:15] OTG Tx (Rx) hub port register (host only)

7	6	5	4	3	2	1	0
0				HUBPORT			
RW				RW			

Address: USBOTGBaseAddress + 0x83 (or 0x87 for Rx) + 8*y, where y = 0 to 15.

Reset: 0x00

Description: These registers are only written when a full- or low-speed device is connected via a high-speed USB 2.0 hub which carries out the necessary transaction translation.

HUBPORT Hub port contains the port number via which the endpoint is accessed.

OTG_DMASEL OTG DMA selector register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	DMASEL_5						DMASEL_4					DMASEL_3		
R	R	RW						RW					RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMASEL_2							DMASEL_1						DMASEL_0		
RW							RW						RW		

Address: USBOTGBaseAddress + 0x200

Reset: 0x0000 0000

Description: Controls which endpoint's DMA request line is connected to six external DMA request signals.

DMASEL_[0:5] DMA selector denotes the endpoint connected to DMA request line x, according to the following:
00000: EP1 Tx, 00001: EP2 Tx through to 01110: EP15 Tx and
01111: EP1 Rx, 10000: EP2 Rx through to 11101: EP15 Rx.

OTG_TOPCTRL OTG top control register

7	6	5	4	3	2	1	0
0	VBUSLO	HDEV	I2C_OFF	XGATE	SRST	UDDR	MODE_ULPI
R	RW	RW	RW	RW	RW	RW	RW

Address: USBOTGBaseAddress + 0x204

Reset: 0x09

Description: Controls the OTG core interface type, the speed of the ULPI interface (standard versus DDR), soft reset of the IP and transceiver's (XCLK) clock gate.

VBUSLO VBUSLO Control. This bit directly controls the VBUS low indicator of the FS Interface. It should be zero when operating with standard USB transceivers. It is used exclusively for operation with transceivers that do not have an embedded charge pump (such a charge pump is controlled and monitored separately).

HDEV Host/device. This bit serves as a host or device indicator when the core is in the FS mode (MODE_ULPI = 0) and the I2C is switched off (I2C_OFF bit is set). When set, this bit indicates host mode; when reset, device mode. This bit has no effect on the functionality in other modes.

I2C_OFF I2C off switch. When set, this bit switches off the I2C controller. This allows the core to communicate with USB 1.1 compatible transceivers. This bit does not affects the functionality unless FS mode (MODE_ULPI = 0) is chosen.

XGATE XCLK gate controls the gate of the transceiver (XCLK) clock. Writing zero to this bit switches off this clock and terminates communication with a transceiver. XCLK is on by default. The system clock (AHB clock) gate is controlled in power management unit (PMU).

SRST Soft reset. Writing 1 to this bit resets the OTG core. Soft reset terminates all current USB communication and resets all register settings to default. This bit always reads zero.

OTG_DMASEL and OTG_TOPCTRL settings are not affected by this soft reset.

UDDR ULPI DDR active controls whether ULPI DDR (double-data rate) mode is active. It does not affect the functionality of the core if the OTG FS interface is selected in MODE_ULPI.

MODE_ULPI Active transceiver interface. The OTG FS interface is compatible with OTG Transc. Spec. v0.92.

0: OTG FS interface active 1: ULPI transceiver interface active

21 Asynchronous serial ports (UART0,1,2)

21.1 UART register addressing

Register addresses are provided as the UART base address of the relevant UART, UARTnBaseAddress, plus the register offset.

All three UART units are identical blocks, but the UART base address differs.

Table 44. UART base addresses

UART	UART base address
UART0	0x101F D000
UART1	0x101F B000
UART2	0x101F 2000

UARTs 0, 1 and 2 are referred to as UARTn throughout this document.

21.2 UART register summary

The UART must be disabled before any of the control registers are reprogrammed. If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping.

Table 45. UART register list

Offset	Register name	Description	Page
0x000	UARTn_DR	UART RxFIFO and TxFIFO data register	*261*
0x004	UARTn_RSR/UARTn_ECR	UART receive status register (read)/ UART error clear register (write)	*262*
0x018	UARTn_FR	UART flag register	*263*
0x024	UARTn_IBRD	UART integer baud rate divisor register	*264*
0x028	UARTn_FBRD	UART fractional baud rate divisor register	*264*
0x02C	UARTn_LCRH	UART line control register, high byte register	*265*
0x030	UARTn_CR	UART control register	*266*
0x034	UARTn_IFLS	UART interrupt FIFO level select register	*267*
0x038	UARTn_IMSC	UART interrupt mask set/clear register	*268*
0x03C	UARTn_RIS	UART raw interrupt status register	*269*
0x040	UARTn_MIS	UART masked interrupt status register	*270*
0x044	UARTn_ICR	UART interrupt clear register	*271*
0x048	UARTn_DMACR	UART DMA control register	*272*
0x050	UARTn_XFCR	UART XON/XOFF control register	*273*
0x054	UARTn_XON1	UART XON1 register	*274*

Table 45. UART register list (continued)

Offset	Register name	Description	Page
0x058	UARTn_XON2	UART XON2 register	274
0x05C	UARTn_XOFF1	UART XOFF1 register	274
0x060	UARTn_XOFF2	UART XOFF2 register	275
0x100	UARTn_ABCR	UART autobaud control register	275
0x104	UARTn_ABSR	UART autobaud status register	276
0x108	UARTn_ABFMT	UART autobaud format register	276
0x150	UARTn_ABDR	UART autobaud baud divisor register	277
0x154	UARTn_ABDFR	UART autobaud baud divisor fraction register	278
0x158	UARTn_ABMR	UART autobaud baud measurement register	278
0x15C	UARTn_ABIMSC	UART autobaud baud interrupt mask set/clear register	279
0x160	UARTn_ABRIS	UART autobaud baud raw status register	279
0x164	UARTn_ABMIS	UART autobaud baud masked status register	280
0x168	UARTn_ABICR	UART autobaud baud interrupt clear register	280
0xFE0	UARTn_PeriphID0	UART peripheral identification register 0 (bits 7:0)	281
0xFE4	UARTn_PeriphID1	UART peripheral identification register 1 (bits 15:8)	281
0xFE8	UARTn_PeriphID2	UART peripheral identification register 2 (bits 23:16)	281
0xFEC	UARTn_PeriphID3	UART peripheral identification register 3 (bits 31:24)	282
0xFF0	UARTn_PCellID0	UART PCell identification register 0 (bits 7:0)	282
0xFF4	UARTn_PCellID1	UART PCell identification register 1 (bits 15:8)	282
0xFF8	UARTn_PCellID2	UART PCell identification register 2 (bits 23:16)	283
0xFFC	UARTn_PCellID3	UART PCell identification register 3 (bits 31:24)	283

21.3 UART register descriptions

UARTn_DR UART data register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	OE	BE	PE	FE	DATA							
R	R	R	R	R	R	R	R	RW							

Address: UARTnBaseAddress + 0x000

Reset: 0xXXXX XXXX

Description: For words to be transmitted:
– if the FIFOs are enabled, data written to this location is shifted onto the transmit FIFO,
– If the FIFOs are disabled, data is stored in the transmitter holding register (the bottom of the transmit FIFO).

The write operation initiates transmission from the UARTn. The data is prefixed with a start bit. The data byte is then transmitted, LSB first, then the appropriate parity bit (if parity is enabled) and one or two stop bits are appended.
For received words:
– if the FIFOs are enabled, the data byte and the 4-bit status (break, frame, parity or overrun) is pushed onto the 12-bit-wide receive FIFO,
– if the FIFOs are not enabled, the data byte and status are stored in the receiving holding register (the bottom word of the receive FIFO).
The received data byte and corresponding status information is read from the UARTn_DR register. The status information can also be read from the *UARTn_RSR/UARTn_ECR* registers.

OE Overrun error.
0: FIFO is no longer full 1: data is received but receive FIFO is already full

BE Break error. Set to 1 if a break condition is detected, indicating that the received data input was held low for longer than a full-word transmission time (defined as start, data, parity and stop bits).
In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes high (marking state), and the next valid start bit is received.

PE Parity error. Set to 1 if the parity of the received data character does not match the parity selected as defined by bits 2 and 7 (respectively bits EPS and SPS) of the *UARTn_LCRH* register.
In FIFO mode, this error is associated with the character at the top of the FIFO.

FE Framing error. Set to 1 if the received character does not have a valid stop bit (a valid stop bit being 1).
In FIFO mode, this error is associated with the character at the top of the FIFO.

DATA Transmit/receive data byte.
Read: receive data character Write: transmit data character

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_RSR/UARTn_ECR UART receive status/error clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	X	X	X	X	OE	BE	PE	FE
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: UARTnBaseAddress + 0x004

Reset: 0x0000 0000

Description: UARTn_RSR provides receive status information for break, framing and parity, which corresponds to the data character read from *UARTn_DR* prior to reading UARTn_RSR. The status information for overrun is set immediately after an overrun condition occurs.
A write to UARTn_ECR clears the framing, parity, break and overrun errors. All bits are cleared to 0 on reset.
The received data character must be read first from UARTn_DR before reading the error status associated with that data character from UARTn_RSR. This read sequence cannot be reversed because UARTn_RSR is updated only when a read occurs from UARTn_DR. Status information can also be obtained by reading the UARTn_DR register.

OE Overrun error bit. Set to 1 if data is received and the receive FIFO is already full. Cleared to 0 by a write to UARTn_ECR (data value is not important).
The FIFO contents remain valid since no further data is written when the FIFO is full, only the content of the shift register is overwritten. The CPU or DMA must then read the data in order to empty the FIFO.

BE Break error bit. Set to 1 if a break condition is detected, indicating that the received data input was held low for longer than a full-word transmission time (defined as start, data, parity and stop bits).
This bit is cleared to 0 after a write to UARTn_ECR.
In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to high (marking state), and the next valid start bit is received.

PE Parity error bit. Set to 1 if the parity of the received data character does not match the parity selected as defined by bits 2 and 7 of the *UARTn_LCRH* register.
This bit is cleared to 0 after a write to UARTn_ECR.
In FIFO mode, this error is associated with the character at the top of the FIFO.

FE Framing error bit. Set to 1 if the received character does not have a valid stop bit (a valid stop bit being 1).
This bit is cleared to 0 after a write to UARTn_ECR.
In FIFO mode, this error is associated with the character at the top of the FIFO.

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_FR UART flag register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	RTXDIS	TERI	DDCD	DDSR	DCTS	RI	TXFE	RXFF	TXFF	RXFE	BUSY	DCD	DSR	CTS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: UARTnBaseAddress + 0x018

Reset: 0x0000 0090

Description: After reset, TXFF, RXFF and BUSY are cleared to 0, and TXFE and RXFE are set to 1.

RTXDIS Remote transmitter disabled (software flow control). Signals that an Xoff character was sent to the remote transmitter to stop it after the received FIFO passed over its trigger limit.
This bit is cleared when an Xon character is sent to the remote transmitter.

TERI Trailing edge ring indicator bit. Set if the URIxnot pin has changed from low to high since the last read of this register.

DDCD Delta data carrier detect bit. Set if the UDCDxnot pin has changed since the last read of this register.

DDSR Delta data set ready bit. Set if the UDSRxnot pin has changed since the last read of this register.

DCTS Delta clear to send bit. Set if the UCTSxnot pin has changed since the last read of this register.

RI Ring indicator bit. Read as the complement of the URIxnot ring indicator modem status input signal.

TXFE Transmit FIFO empty. If FIFO is disabled (*UARTn_LCRH*.FEN = 0), the transmit FIFO empty bit is set when the transmit holding register is empty.
If the FIFO is enabled (UARTn_LCRH.FEN = 1), the TXFE bit is set when the transmit FIFO is empty.

RXFF Receive FIFO full. If FIFO is disabled (*UARTn_LCRH*.FEN = 0), the receive FIFO full bit is set when the receive holding register is full.
If the FIFO is enabled (*UARTn_LCRH*.FEN = 1), the RXFF bit is set when the receive FIFO is full.

TXFF Transmit FIFO full. If FIFO is disabled (*UARTn_LCRH*.FEN = 0), the transmit FIFO full bit is set when the transmit holding register is full.
If the FIFO is enabled (*UARTn_LCRH*.FEN = 1), the TXFF bit is set when the transmit FIFO is full.

RXFE Receive FIFO empty. If FIFO is disabled (*UARTn_LCRH*.FEN = 0), the receive FIFO empty bit is set when the receive holding register is empty.
If the FIFO is enabled (*UARTn_LCRH*.FEN = 1), the RXFE bit is set when the receive FIFO is empty.

BUSY UARTn busy. Set when the UARTn is busy transmitting data. This bit remains set until the complete byte, including all the stop bits, has been sent from the shift register. However, if the transmit section of the UARTn is disabled in the middle of a transmission, the BUSY bit gets cleared. This bit is set again once the transmit section is re-enabled to complete the remaining transmission.
This bit is set as soon as the transmit FIFO becomes non-empty (regardless of whether UARTn is enabled or not).

DCD Data carrier detect bit. Complement of the UDCDxnot data carrier detect modem status input signal.

DSR Data set ready bit. Complement of the UDSRxnot data set ready modem status input signal.

CTS Clear to send bit. Complement of the UCTSxnot clear to send modem status input signal.

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_IBRD UART integer baud rate divisor register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVINT															
RW															

Address: UARTnBaseAddress + 0x024

Reset: 0x0000 0000

Description: This is the integer part of the baud rate divisor value. The minimum divide ratio possible is 0x1 and the maximum is 0xFFFF (2^{16} - 1). If *UARTn_IBRD* = 0x0 the register is ignored. If UARTn_IBRD = 0xFFFF, then *UARTn_FBRD* cannot be greater than 0. Failure to observe this results in an aborted transmission or reception. UARTn_IBRD, UARTn_FBRD, and *UARTn_LCRH* form a single 29-bit-wide register that is updated on a single write strobe, generated by a write in UARTn_LCRH. Therefore, to internally update the content of UARTn_IBRD or UARTn_FBRD, a write to UARTn_LCRH must always be performed after transmission or reception of the current character has finished.

DIVINT Baud rate integer. The baud rate divisor comprises the integer value (DIVINT) and the fractional value (DIVFRAC). It is calculated as follows:

– baud rate divisor = (frequency [UARTCLK] / [16 x baud rate]) when *UARTn_CR*.OVSFACT = 0,

– baud rate divisor = (frequency [UARTCLK] / [8 x baud rate]) when UARTn_CR.OVSFACT = 1,

where frequency [UARTCLK] is the UART reference clock frequency.

The contents of the UARTn_IBRD and UARTn_FBRD registers are not updated until transmission or reception of the current character has finished (a write to UARTn_LCRH has occurred).

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_FBRD UART fractional baud rate divisor register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	DIVFRAC					
R	R	R	R	R	R	R	R	R	R	RW					

Address: UARTnBaseAddress + 0x028

Reset: 0x0000 0000

Description: This is the fractional part of the baud rate divisor value. The minimum divide ratio possible is 0x1 and the maximum is 0xFFFF 2^{16} - 1). If *UARTn_IBRD* = 0x0, the register is ignored. If UARTn_IBRD = 0xFFFF, then UARTn_FBRD cannot be greater than zero. Failure to observe this results in an aborted transmission or reception.

UARTn_IBRD, UARTn_FBRD and *UARTn_LCRH* form a single 29-bit-wide register that is updated on a single write strobe generated by a write in UARTn_LCRH. Therefore, to internally update the content of UARTn_IBRD or UARTn_FBRD, a write to UARTn_LCRH must always be performed after transmission or reception of the current character has finished.

DIVFRAC Baud rate fraction. The baud rate divisor comprises the integer value (DIVINT) and the fractional value (DIVFRAC). It is calculated as follows:

– baud rate divisor = (frequency [(UARTCLK] / [16 x baud rate]) when *UARTn_CR*.OVSFACT = 0,

– baud rate divisor = (frequency [UARTCLK] / [8 x baud rate]) when UARTn_CR.OVSFACT = 1,

where Frequency [UARTCLK] is the UART reference clock frequency.

The contents of the UARTn_IBRD and UARTn_FBRD registers are not updated until transmission or reception of the current character has finished (a write to UARTn_LCRH has occurred).

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_LCRH UART line control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SPS	WLEN		FEN	STP2	EPS	PEN	BRK
R	R	R	R	R	R	R	R	RW	RW		RW	RW	RW	RW	RW

Address: UARTnBaseAddress + 0x02C

Reset: 0x0000 0000

Description: Registers *UARTn_IBRD*, *UARTn_FBRD*, and UARTn_LCRH form a single 29-bit-wide register that is updated on a single write strobe generated by a write in

UARTn_LCRH. To internally update the content of UARTn_IBRD or UARTn_FBRD, a write to UARTn_LCRH must always be performed at the end.

SPS Stick parity select.

0: stick parity is disabled.

1: if PEN = 1 and EPS = 1, the parity bit is transmitted and checked as a 0, but if PEN = 1 and EPS = 0 (odd parity), the parity bit is transmitted and checked as a 1.

WLEN Word length. Indicates the number of data bits transmitted or received in a frame.

00: 5 bits 01: 6 bits

10: 7 bits 11: 8 bits

FEN Enable FIFOs. Enables/disables the transmit and receive FIFO buffers.

0: FIFOs are disabled (character mode): FIFOs become 1-byte holding register.

1: FIFOs are enabled.

STP2 Two stop bits select. Enables transmission of two stop bits at the end of the frame. The receive logic does not check for two stop bits being received.

0: 1 stop bit transmitted 1: 2 stop bits transmitted

EPS Even parity select. Selects parity generation when parity is enabled (PEN = 1). This bit has no effect when parity is disabled (PEN = 0).

0: odd parity generation and checking is performed during transmission and reception, which check for an odd number of 1s in data and parity bits.

1: even parity generation and checking is performed during transmission and reception, which check for an even number of 1s in data and parity bits.

PEN Parity enable.

0: disabled 1: enabled

BRK Send break bit. Forces a continuous low level on UTXDx output, after completion of the current character. This bit must be asserted for at least one complete frame transmission time in order to generate a break condition. The transmit FIFO contents remain unaffected during a break condition.

0: normal transmission 1: break condition transmission

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_CR UART control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTS EN	RTS EN	0	0	RTS	DTR	RXE	TXE	LBE	0	0	0	OVS FACT	0	0	UART EN
RW	RW	R	R	RW	RW	RW	RW	RW	R	R	R	RW	R	R	RW

Address: UARTnBaseAddress + 0x030

Reset: 0x0000 0300

Description: To enable transmission, both the TXE and UARTEN bits must be set. To enable reception, both the RXE and UARTEN bits must be set.

CTSEN CTS hardware flow control enable.
0: disabled.
1: enabled. Data is only transmitted when the UCTSxnot signal is asserted.

RTSEN RTS hardware flow control enable.
0: disabled
1: enabled. Data is only requested when there is space in the receive FIFO for it to be received.

RTS Request to send. Complement of the URTSxnot modem status output.
0: URTSxnot is high 1: URTSxnot is low

DTR Data transmit ready. Complement of the UDTRxnot modem status output.
0: UDTRxnot is high 1: UDTRxnot is low

RXE Receive enable. Enables the receive section of the UART when the UARTn receiver is disabled in the middle of reception. It completes the current character before stopping.
0: receiver disabled 1: receiver enabled

TXE Transmit enable. Enables the transmit section of the UART. When the UART transmitter is disabled in the middle of transmission, it completes the current character before stopping.
0: transmitter disabled 1: transmitter enabled

LBE Loop-back enable. Enables loop-back mode, which is intended for testing only. In loop-back mode, the UTXDx is internally connected to the URXDx signal, and the control modem output signals are internally connected to the control modem input signals.
0: loop-back mode disabled (normal operation) 1: loop-back mode enabled

OVSFACT UART over-sampling factor. The UARTCLK is 48 MHz in the STn8815. The maximum baud rate is 3 Mbauds when OVSFACT = 0 and 6 Mbauds when OVSFACT = 1.
0: UART bit is 16 UARTCLK clock cycles 1: UART bit is 8 UARTCLK clock cycles

UARTEN UART enable. Enables the UART.
0: UART disabled
1: UART enabled. Data transmission and reception can occur. If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping.

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_IFLS UART interrupt FIFO level select register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	RXIFLSEL			TXIFLSEL		
R	R	R	R	R	R	R	R	R	R	RW			RW		

Address: UARTnBaseAddress + 0x034

Reset: 0x0000 0000

Description: Defines the FIFO level at which the transmit and receive interrupts and burst DMA requests are triggered.

The interrupts are generated based on a transition through a level rather than based on the level, meaning that the interrupt is generated when the fill level progresses through the trigger level.

RXIFLSEL Receive interrupt FIFO level select bit field. Selects the trigger point for receive FIFO interrupt and DMA request.

000: receive FIFO becomes >= 1/64 full (>= 1 character in Rx FIFO)

001: receive FIFO becomes >= 1/32 full (>= 2 characters in Rx FIFO)

010: receive FIFO becomes >= 1/16 full (>= 4 characters in Rx FIFO)

011: receive FIFO becomes >= 1/8 full (>= 8 characters in Rx FIFO)

100: receive FIFO becomes >= 1/4 full (>= 16 characters in Rx FIFO)

101: receive FIFO becomes >= 1/2 full (>= 32 characters in Rx FIFO)

110: receive FIFO becomes >= 3/4 full (>= 48 characters in Rx FIFO)

TXIFLSEL Transmit interrupt FIFO level select bit field. Selects the trigger point for transmit FIFO interrupt and DMA request.

000: transmit FIFO becomes >= 1/64 empty (>= 1 empty location in Tx FIFO)

001: transmit FIFO becomes >= 1/32 empty (>= 2 empty locations in Tx FIFO)

010: transmit FIFO becomes >= 1/16 empty (>= 4 empty locations in Tx FIFO)

011: transmit FIFO becomes >= 1/8 empty (>= 8 empty locations in Tx FIFO)

100: transmit FIFO becomes >= 1/4 empty (>= 16 empty locations in Tx FIFO)

101: transmit FIFO becomes >= 1/2 empty (>= 32 empty locations in Tx FIFO)

110: transmit FIFO becomes >= 3/4 empty (>= 48 empty locations in Tx FIFO)

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_IMSC UART interrupt mask set/clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	XOFFIM	OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM	DSRMIM	DCDMIM	CTSMIM	RIMIM
R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: UARTnBaseAddress + 0x038

Reset: 0x0000 0000

Description: Returns the current value of the mask on the relevant interrupt. A write of 1 to a particular bit sets the mask of that interrupt. A write of 0 clears the corresponding mask. All bits are cleared to 0 when reset.

XOFFIM XOFF interrupt mask. Reading returns the current mask for the XOFFIM interrupt. Writing:
0: clears the mask 1: sets the mask

OEIM Overrun error interrupt mask. Reading returns the current mask for the OEIM interrupt. Writing:
0: clears the mask 1: sets the mask

BEIM Break error interrupt mask. Reading returns the current mask for the BEIM interrupt. Writing:
0: clears the mask 1: sets the mask

PEIM Parity error interrupt mask. Reading returns the current mask for the PEIM interrupt. Writing:
0: clears the mask 1: sets the mask

FEIM Framing error interrupt mask. Reading returns the current mask for the FEIM interrupt. Writing:
0: clears the mask 1: sets the mask

RTIM Receive timeout interrupt mask. Reading returns the current mask for the RTIM interrupt. Writing:
0: clears the mask 1: sets the mask

TXIM Transmit interrupt mask. Reading returns the current mask for the TXIM interrupt. Writing:
0: clears the mask 1: sets the mask

RXIM Receive interrupt mask. Reading returns the current mask for the RXIM interrupt. Writing:
0: clears the mask 1: sets the mask

DSRMIM Data set ready modem interrupt mask. Reading returns the current mask for the DSRMIM interrupt. Writing:
0: clears the mask 1: sets the mask

DCDMIM Data carrier detect modem interrupt mask. Reading returns the current mask for the DCDMIM interrupt. Writing:
0: clears the mask 1: sets the mask

CTSMIM Clear-to-send modem interrupt mask. Reading returns the current mask for the CTSMIM interrupt. Writing:
0: clears the mask 1: sets the mask

RIMIM Ring indicator modem interrupt mask. Reading returns the current mask for the IRMIM interrupt. Writing:
0: clears the mask 1: sets the mask

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_RIS UART raw interrupt register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	XOFF RIS	OE RIS	BE RIS	PE RIS	FE RIS	RT RIS	TX RIS	RX RIS	DSR MRIS	DCD MRIS	CTS MRIS	RI MRIS
R	R	R	R	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH

Address: UARTnBaseAddress + 0x03C

Reset: 0x0000 000X

Description: Indicates the current raw status value of interrupts. A write has no effect. All bits, except for the modem status interrupt bits, are cleared to 0 upon reset. The modem status interrupt bits are undefined after reset.

XOFFRIS XOFF raw interrupt status. A read returns the raw interrupt state (prior to masking) of the Xoff/special character interrupt.

OERIS Overrun error raw interrupt status. A read returns the raw interrupt state (prior to masking) of the overrun error interrupt.

BERIS Break error raw interrupt status. A read returns the raw interrupt state (prior to masking) of the break error interrupt.

PERIS Parity error raw interrupt status. A read returns the raw interrupt state (prior to masking) of the parity error interrupt.

FERIS Framing error raw interrupt status. A read returns the raw interrupt state (prior to masking) of the framing error interrupt.

RTRIS Receive timeout raw interrupt status. A read returns the raw interrupt state (prior to masking) of the receive timeout interrupt. In this case, the raw interrupt cannot be set unless the mask is set, because the mask acts as an *enable* for timeout detection for power saving. The same status can be read from the *UARTn_MIS* and UARTn_RIS registers for the receive timeout interrupt.

TXRIS Transmit raw interrupt status. A read returns the raw interrupt state (prior to masking) of the transmit interrupt.

RXRIS Receive raw interrupt status. A read returns the raw interrupt state (prior to masking) of the receive interrupt.

DSRMRIS Data set ready modem raw interrupt status. A read returns the raw interrupt state (prior to masking) of the data set ready modem interrupt.

DCDMRIS Data carrier detect modem raw interrupt status. A read returns the raw interrupt state (prior to masking) of the data carrier detect modem interrupt.

CTSMRIS Clear to send modem raw interrupt status. A read returns the raw interrupt state (prior to masking) of the clear-to-send modem interrupt.

RIMRIS Ring indicator modem raw interrupt status. A read returns the raw interrupt state (prior to masking) of the ring indicator modem interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_MIS — UART masked interrupt register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	XOFF MIS	OE MIS	BE MIS	PE MIS	FE MIS	RT MIS	TX MIS	RX MIS	DSR MMIS	DCD MMIS	CTS MMIS	RI MMIS
R	R	R	R	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH	RH

Address: UARTnBaseAddress + 0x040

Reset: 0x0000 000X

Description: Indicates the current masked status value of the corresponding interrupt. A write has no effect.

All bits, except for the modem status interrupt bits, are cleared to 0 upon reset. The modem status interrupt bits are undefined after reset.

XOFFMIS XOFF masked interrupt status. A read returns the masked interrupt state of the Xoff/special character interrupt.

OEMIS Overrun error masked interrupt status. A read returns the masked interrupt state of the overrun error interrupt.

BEMIS Break error masked interrupt status. A read returns the masked interrupt state of the break error interrupt.

PEMIS Parity error masked interrupt status. A read returns the masked interrupt state of the parity error interrupt.

FEMIS Framing error masked interrupt status. A read returns the masked interrupt state of the framing error interrupt.

RTMIS Receive timeout masked interrupt status. A read returns the masked interrupt state of the receive timeout interrupt.

TXMIS Transmit masked interrupt status. A read returns the masked interrupt state of the transmit interrupt.

RXMIS Receive masked interrupt status. A read returns the masked interrupt state of the receive interrupt.

DSRMMIS Data set ready modem masked interrupt status. A read returns the masked interrupt state of the data set ready modem interrupt.

DCDMMIS Data carrier detect modem masked interrupt status. A read returns the masked interrupt state of the data carrier detect modem interrupt.

CTSMMIS Clear-to-send modem masked interrupt status. A read returns the masked interrupt state of the clear to-send-modem interrupt

RIMMIS Ring indicator modem masked interrupt status. A read returns the masked interrupt state of the ring indicator modem interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_ICR UART interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	XOFFIC	OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC	DSRMIC	DCDMIC	CTSMIC	RIMIC
R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: UARTnBaseAddress + 0x044

Reset: 0x0000 0000

Description: Always reads as zero. Writing 1 clears the corresponding interrupt.

XOFFIC XOFF interrupt clear.
0: no effect 1: clears the Xoff/special character interrupt

OEIC Overrun error interrupt clear.
0: no effect 1: clears the overrun error interrupt

BEIC Break error interrupt clear.
0: no effect 1: clears the break error interrupt

PEIC Parity error interrupt clear.
0: no effect 1: clears the parity error interrupt

FEIC Framing error interrupt clear.
0: no effect 1: clears the framing error interrupt

RTIC Receive timeout interrupt clear.
0: no effect 1: clears the receive timeout interrupt

TXIC Transmit interrupt clear.
0: no effect 1: clears the transmit interrupt

RXIC Receive interrupt clear.
0: no effect 1: clears the receive interrupt

DSRMIC Data set ready modem interrupt clear.
0: no effect 1: clears the data set ready modem interrupt

DCDMIC Data carrier detect modem interrupt clear.
0: no effect 1: clears the data carrier detect modem interrupt

CTSRIC Clear-to-send modem interrupt clear.
0: no effect 1: clears the clear-to-send modem interrupt

RIMIC Ring indicator modem interrupt clear.
0: no effect 1: clears the ring indicator modem interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_DMACR UART DMA control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	DMAONERR	TXDMAE	RXDMAE
R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Address: UARTnBaseAddress + 0x048

Reset: 0x0000 0000

Description: All bits are cleared to 0 on reset.

DMAONERR DMA on error.
1: DMA receive requests are disabled when the UART error interrupt is asserted.

TXDMAE Transmit DMA enable.
1: transmit FIFO DMA is enabled.

RXDMAE Receive DMA enable.
1: receive FIFO DMA is enabled.

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_XFCR UART XON/XOFF control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	SPE CHR	XON ANY	SFTMOD		SFRMOD		SFEN
R	R	R	R	R	R	R	R	R	R	R	RW		RW		RW

Address: UARTnBaseAddress + 0x050

Reset: 0x0000 0000

Description: Controls software flow. To enable software flow control, both the SFEN and SFRFMOD or SFTMOD bits must be used.

SFEN Software flow control enable.
0: disabled 1: enabled

SFRFMOD Software receive flow control mode.
00: disabled 01: Xon1, Xoff1 characters are used
10: Xon2, Xoff2 characters are used 11: Xon1, Xon2, Xoff1, Xoff2 characters are used

SFTMOD Software transmit flow control mode.
00: disabled 01: Xon1, Xoff1 characters are used
10: Xon2, Xoff2 characters are used 11: Xon1, Xon2, Xoff1, Xoff2 characters are used

XONANY Xon-any bit. Defines when an incoming character is considered as a valid Xon.
0: when it matches Xon programmed value(s) 1: any incoming character is a valid Xon

SPECHR Special character detection bit.
0: detection disabled 1: detection enabled

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_XON1 UART XON1 register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	XON1							
R	R	R	R	R	R	R	R	RW							

Address: UARTnBaseAddress + 0x054

Reset: 0x0000 0000

Description: Stores the Xon1 character used in the software flow control.

UARTn_XON2 UART XON2 register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	XON2							
R	R	R	R	R	R	R	R	RW							

Address: UARTnBaseAddress + 0x058

Reset: 0x0000 0000

Description: Stores the Xon2 character used in the software flow control.

UARTn_XOFF1 UART XOFF1 register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	XOFF1							
R	R	R	R	R	R	R	R	RW							

Address: UARTnBaseAddress + 0x05C

Reset: 0x0000 0000

Description: Stores the Xoff1 character used in the software flow control.

UARTn_XOFF2 UART XOFF2 register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	XOFF2							
R	R	R	R	R	R	R	R	RW							

Address: UARTnBaseAddress + 0x060

Reset: 0x0000 0000

Description: Stores the Xoff2 character used in the software flow control.

UARTn_ABCR UART autobaud control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	UPDAT EN	RE START	ACF GEN
R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW

Address: UARTnBaseAddress + 0x100

Reset: 0x0000 0000

Description: Control the various modes of operation of the autobaud logic.

UPDATEN Update enable. Selects if the autobaud updates the UART registers when a valid sequence is detected.
0: do not update 1: update

RESTART Autobaud restart enable. Enables autoconfig to retry if an invalid sequence is detected.
0: terminate autobaud on error 1: retry until valid autobaud completes

ACFGEN Autoconfig enable. Starts or stops the autoconfig sequence.
0: stop autoconfig 1: start autoconfig

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_ABSR UART autobaud status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	AC DONE	BAUD ERR	CMDERR		VAL FMT	BYTE1S					BYTE2S		
R	R	R	R	R	R		R	R					R		

Address: UARTnBaseAddress + 0x104

Reset: 0x0000 0000

Description: Indicates if autoconfiguration has completed and the status of that process. If the process fails, the register indicates why. Apart from the ACDONE bit, this register is only valid when the process has completed. All bits are cleared after reset.

ACDONE Autoconfig complete status. Set when autoconfig completes.

BAUDERR Invalid baud rate status. Set when invalid baud rate detected. Only valid when ACDONE = 1.

CMDERR Incorrect command sequence detected. Set when the autoformat logic detects an incorrect command sequence. Only valid when ACDONE = 1.

00: no command sequence error detected
01: character (at) OK, format incorrect
10: character (a/) OK, format incorrect
11: invalid character

VALFMT Valid format detected. Set when the autoformat logic has successfully completed, for example an ‘at’ or ‘AT’ sequence received. Only valid when ACDONE = 1.

BYTE1S Byte 1 status indicated by the value of bits [7:11] after the first start bit. Determines the format. Only valid when ACDONE = 1 and VALFMT = 1.

BYTE2S Byte 2 status indicated by the value of bits [7:9] after the second start bit. Determines the format. Only valid when ACDONE = 1 and VALFMT = 1.

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_ABFMT UART autobaud format register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	WLEN		FEN	STP2	EPS	PEN	0
R	R	R	R	R	R	R	R	R	R		R	R	R	R	R

Address: UARTnBaseAddress + 0x108

Reset: 0x0000 007E

Description: Indicates the format decoded by the autoformat logic. It has the same structure as the *UARTn_LCRH* register. Only valid when the autobaud process successfully completed.

WLEN Word length. Number of data bits per character. Only valid when ACDONE= 1 and VALFMT= 1 in *UARTn_ABSR*.
10: 7 bits 11: 8 bits

FEN FIFO enable. Always set (FIFOs enabled).

STP2 Two stop bits. Selects the number of stop bits. Only valid when ACDONE = 1 and VALFMT = 1.
0: 1 stop bit 1: 2 stop bits

EPS Even parity select. Only valid when ACDONE = 1 and VALFMT = 1.
0: odd parity 1: even parity

PEN Parity enable. Only valid when ACDONE = 1 and VALFMT = 1.
0: no parity bit 1: parity bit present

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_ABDR UART autobaud divisor register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUTOBAUDBDR															
RH															

Address: UARTnBaseAddress + 0x150

Reset: 0x0000 0000

Description: Returns the extracted baud divisor value. Only valid when the autobaud process has completed. When *UARTn_CR*.OVSFAT = 1 the software must check that bit 19 of *UARTn_ABMR* = 0, otherwise this register will not provide the correct integer part of the measured baud rate.

AUTOBAUDBDR Autobaud divisor value. Integer part of the measured baud rate.

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_ABDFR UART autobaud divisor fraction register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	AUTOBAUDBDFR					
R	R	R	R	R	R	R	R	R	R	RH					

Address: UARTnBaseAddress + 0x154

Reset: 0x0000 0000

Description: Returns the fractional component of the extracted baud divisor value. Only valid when the autobaud process has completed.

AUTOBAUDBDFR Autobaud divisor fraction value. Fraction part of the measured baud rate.

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_ABMR UART autobaud measurement register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	AUTOBAUDBMR			
R	R	R	R	R	R	R	R	R	R	R	R	RH			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUTOBAUDBMR															
RH															

Address: UARTnBaseAddress + 0x158

Reset: 0x0000 0000

Description: Returns the detected length of the start bit. Only valid when the autobaud process has completed. When *UARTn_CR*.OVSFAT = 1, the software must check that bit 19 of this register = 0, otherwise the *UARTn_ABDR* register will not provide the correct integer part of the measured baud rate.

AUTOBAUDBMR Autobaud measurement value. Raw duration measurement of the length of the start bit, expressed in UARTCLK cycles.

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_ABIMSC UART autobaud interrupt mask set/clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	ABDONEIM	ABERRIM
R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Address: UARTnBaseAddress + 0x15C

Reset: 0x0000 0000

Description: Returns the current value of the mask on the relevant interrupt. A write of 1 to a particular bit sets the mask of that interrupt. A write of 0 clears the corresponding mask.

ABDONEIM Autobaud done interrupt mask. Returns the current mask for the UARTnABDONEINTR interrupt. Writing:

0: clears the mask 1: sets the UARTnABDONEINTR interrupt mask

ABERRIM Autobaud error interrupt mask. Returns the current mask for the UARTnABERRINTR interrupt. Writing:

0: clears the mask 1: sets the UARTnABERRINTR interrupt mask

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_ABRIS UART autobaud raw interrupt register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	ABDONERIS	ABERRRIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	RH	RH

Address: UARTnBaseAddress + 0x160

Reset: 0x0000 0000

Description: Returns the current raw status value of the interrupt. A write has no effect.

ABDONERIS Overrun error interrupt status. Returns the raw interrupt state (prior to masking) of the autobaud done interrupt.

ABERRRIS Break error interrupt status. Returns the raw interrupt state (prior to masking) of the autobaud error interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_ABMIS UART autobaud masked interrupt register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	ABDONEMIS	ABERRMIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	RH	RH

Address: UARTnBaseAddress + 0x164

Reset: 0x0000 0000

Description: Returns the current masked status value of an interrupt. A write has no effect.

ABDONEMIS Autobaud done masked interrupt status. Returns masked interrupt state of autobaud done interrupt.

ABERRMIS Autobaud error masked interrupt status. Returns masked interrupt state of autobaud error interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_ABICR UART autobaud interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	ABDONEIC	ABERRIC
R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Address: UARTnBaseAddress + 0x168

Reset: 0x0000 0000

Description: This register is always read as zero.

ABDONEIC Autobaud done interrupt clear.
0: no effect 1: clears the autobaud done interrupt

ABERRIC Autobaud error interrupt clear.
0: no effect 1: clears the autobaud error interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

UARTn_PeriphID0 UART peripheral identification registers 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: UARTnBaseAddress + 0xFE0

Reset: 0x0000 0002

Description: PartNumber0 reads back as 0x02

UARTn_PeriphID1 UART peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: UARTnBaseAddress + 0xFE4

Reset: 0x0000 0008

Description: Designer0 reads back as 0x0, PartNumber1 reads back as 0x8.

UARTn_PeriphID2 UART peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: UARTnBaseAddress + 0xFE8

Reset: 0x0000 0038

Description: Revision reads back as 0x3, Designer1 reads back as 0x8.

UARTn_PeriphID3 UART peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: UARTnBaseAddress + 0xFEC

Reset: 0x0000 0000

Description: Configuration reads back as 0x00.

UARTn_PCellID0 UART PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	UARTnPCellID0							
R	R	R	R	R	R	R	R	R							

Address: UARTnBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: UARTnPCellID0 reads back as 0x0D.

UARTn_PCellID1 UART PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	UARTnPCellID1							
R	R	R	R	R	R	R	R	R							

Address: UARTnBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: UARTnPCellID1 reads back as 0xF0.

UARTn_PCellID2 UART PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	UARTnPCellID2							
R	R	R	R	R	R	R	R	R							

Address: UARTnBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: UARTnPCellID2 reads back as 0x05.

UARTn_PCellID3 UART PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	UARTnPCellID3							
R	R	R	R	R	R	R	R	R							

Address: UARTnBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: UARTnPCellID3 reads back as 0xB1.

22 Synchronous serial port (SSP)

22.1 L2CC register addressing

Register addresses are provided as the SSP base address, SSPBaseAddress, plus the offset.

The SSPBaseAddress is 0x101F C000.

22.2 L2CC register summary

The following SSP parameters are programmed through the set of registers shown in *Table 46*.

- Master or slave mode.
- Enable operation.
- Frame format.
- Communication baud rate.
- Clock phase and polarity.
- Data widths from 4 to 32 bits wide.
- Interrupt masking.

Table 46. SSP register list

Offset	Register name	Description	Page
0x000	SSP_CR0	SSP control register 0	*285*
0x004	SSP_CR1	SSP control register 1	*286*
0x008	SSP_DR	SSP data register	*287*
0x00C	SSP_SR	SSP status register	*288*
0x010	SSP_CPSR	SSP clock prescale register	*288*
0x014	SSP_IMSC	SSP interrupt mask set and clear register	*289*
0x018	SSP_RIS	SSP raw interrupt status register	*290*
0x01C	SSP_MIS	SSP masked interrupt status register	*290*
0x020	SSP_ICR	SSP interrupt clear register	*291*
0x024	SSP_DMACR	SSP DMA control register	*291*
0xFE0	SSP_PeriphID0	SSP peripheral identification register (bits 7:0)	*292*
OxFE4	SSP_PeriphID1	SSP peripheral identification register (bits 15:8)	*292*
0xFE8	SSP_PeriphID2	SSP peripheral identification register (bits 23:16)	*292*
OxFEC	SSP_PeriphID3	SSP peripheral identification register (bits 31:24)	*293*
0xFF0	SSP_PCellID0	SSP PCell identification register (bits 7:0)	*293*
0xFF4	SSP_PCellID1	SSP PCell identification register (bits 15:8)	*293*

Table 46. SSP register list (continued)

Offset	Register name	Description	Page
0xFF8	SSP_PCellID2	SSP PCell identification register (bits 23:16)	294
0xFFC	SSP_PCellID3	SSP PCell identification register (bits 31:24)	294

22.3 L2CC register descriptions

SSP_CR0 SSP control register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	FRF		CSS				
R	R	R	R	R	R	R	R	R	RW		RW				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR								SPH	SPO	HALF DUP	DSS				
RW								RW	RW	RW	RW				

Address: SSPBaseAddress + 0x000

Reset: 0x0000 0000

Description: The synchronous serial port (SSP) is configured via this register and *SSP_CR1*.

FRF Frame format.
00: Motorola SPI frame format — 01: TI synchronous serial frame format
10: National MicroWire frame format — 11: unidirectional frame format

CSS Command size select +1. Defines the number of bits for the command in Microwire mode.
00000: reserved, undefined operation — 00001: reserved, undefined operation
00010: reserved, undefined operation — 00011: 4-bit data
00100: 5-bit data, through to — 11111: 32-bit data

SCR Serial clock rate. Generates the transmit/receive bit rate of the SSP.
The bit rate is : F_{SSPCLK} / [CPSDVR x (1 + SCR)]
where CPSDVR is an even value from 2 to 254, programmed through register *SSP_CPSR*, and SCR is a value from 0 to 255.

SPH Motorola SPI clock phase (only applicable to Motorola SPI frame format).
0: received data is captured on the rising edge (SPO = 0) or falling edge (SPO = 1) of SSPCLK signal. Transmitted data is sent on the falling edge (SPO = 0) or rising edge (SPO = 1) of SSPCLK.
1: received data is captured on the falling edge (SPO = 0) or rising edge (SPO = 1) of SSPCLK. Transmitted data is sent on rising edge (SPO = 0) or falling edge (SPO = 1) of SSPCLK.

SPO SSPCLK clock polarity (only applicable to Motorola SPI frame format). Defines the clock polarity.
0: the inactive or idle state of SSPCLK is low — 1: the inactive or idle state of SSPCLK is high

HALFDUP Half duplex mode.

0: full duplex (SSPTXD signal is the output and SSPRXD is the input). This configuration must be used for SPI and TI modes. When used in Microwire mode, implies external connection of SSPRXD to SSPTXD pins.

1: half duplex (SSPTXD signal is bidirectional, driven by the SSP when transmitting data, and in input mode when receiving data). This setting should not be used for SPI and TI modes. It can be used for Microwire mode to avoid external connection of SSTRXD to SSPTXD. SSPRX can then be used as a GPIO.

DSS Data size select +1. Defines the number of bits in TI and SPI modes and the number of bits for the returned data in Microwire mode.

00000: reserved, undefined operation — 00001: reserved, undefined operation

00010: reserved, undefined operation — 00011: 4-bit data

00100: 5-bit data; through — 11111: 32-bit data

0 Reserved for future use. Reading returns 0. Must be written with 0.

SSP_CR1 SSP control register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	TXIFLSEL			RXIFLSEL			MWAIT	TENDN	RENDN	SOD	MS	SSE	LBM
R	R	R	RW			RW			RW	RW	RW	RW	RW	RW	RW

Address: SSPBaseAddress + 0x004

Reset: 0x0000 0000

Description: The synchronous serial port (SSP) is configured via this register and *SSP_CR0*.

TXIFLSEL Transmit interrupt FIFO level select. Selects transmit FIFO interrupt and DMA request trigger points.

000: transmit FIFO becomes more than 1/32 empty (more than 1 empty location in Tx FIFO).

001: transmit FIFO becomes more than 1/8 empty (more than 4 empty locations in Tx FIFO).

010: transmit FIFO becomes more than 1/4 empty (more than 8 empty locations in Tx FIFO).

011: transmit FIFO becomes more than 1/2 empty (more than 16 empty locations in Tx FIFO).

100: through 111: reserved.

RXIFLSEL Receive interrupt FIFO level select. Selects receive FIFO interrupt and DMA request trigger points.

000: receive FIFO becomes more than 1/32 full (more than 1 element in Rx FIFO).

001: receive FIFO becomes more than 1/8 full (more than 4 elements in Rx FIFO).

010: receive FIFO becomes more than 1/4 full (more than 8 elements in Rx FIFO).

011: receive FIFO becomes more than 1/2 full (more than 16 elements in Rx FIFO).

100: through 111: reserved.

MWAIT Microwire wait state enable. Determines if a wait state is inserted after the last command bit has been sent (for MicroWire format).

0: no wait state — 1: one wait state

TENDN Transmit endian format. Determines whether the element is transmitted MSB or LSB first.

0: transmitted MSB first — 1: transmitted LSB first

RENDN Receive endian format. Determines whether the element is received MSB or LSB first.
0: received MSB first 1: received LSB first

SOD Slave mode output disable. Only relevant in slave mode (MS = 1). In multiple-slave systems, it is possible for a master SSP to broadcast a message to all slaves in the system while ensuring that only one slave drives data onto its serial output line. In such systems, the SSPRXD lines from multiple slaves can be tied together. To operate in such systems, the SOD bit can be set if the slave SSP is not supposed to drive the SSPTXD line.
0: SSP can drive the SSPTXD output in slave mode.
1: SSP must not drive the SSPTXD output in slave mode.

MS Master or slave mode select.
0: configured as master (default) 1: configured as slave

SSE SSP enable.
0: SSP operation disabled 1: SSP operation enabled

LBM Loop-back mode.
0: normal serial port operation enabled.
1: output of transmit serial shifter is connected internally to input of receive serial shifter.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SSP_DR SSP data register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA, bits [31:16]															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA, bits [15:0]															
RW															

Address: SSPBaseAddress + 0x008

Reset: 0xXXXX XXXX

Description: This register is 32-bits wide. When SSP_DR is read, the entry in the receive FIFO (pointed to by the current FIFO read pointer) is accessed. As data values are removed by the SSP receive logic from the incoming data frame, they are placed in the entry in the receive FIFO (pointed to by the current FIFO write pointer).
When SSP_DR is written to, the entry in the transmit FIFO (pointed to by the write pointer) is written to. Data values are removed from the transmit FIFO one value at a time by the transmit logic. Each value is loaded into the transmit serial shifter, then serially shifted out onto the SSPTXDx pin at the programmed bit rate.
When a data size of less than 32 bits is selected, the user must right-justify data written to the transmit FIFO. The transmit logic ignores the unused bits. Received data less than 32 bits is automatically right-justified in the receive FIFO.
The transmit FIFO and the receive FIFO are not cleared even when the SSE bit is cleared. This allows the software to fill the transmit FIFO before enabling the SSP.

DATA Transmit or receive data.
Read: RxFIFO is read Write: TxFIFO is written

SSP_SR SSP status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	BSY	RFF	RNE	TNF	TFE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: SSPBaseAddress + 0x00C

Reset: 0x0000 0003

Description: Contains bits that indicate the FIFO fill status and the SSP busy status.

BSY SSP busy flag.
0: SSP is idle
1: SSP is currently transmitting and/or receiving a frame or the transmit FIFO is not empty.

RFF Receive FIFO full.
0: receive FIFO is not full 1: receive FIFO is full

RNE Receive FIFO not empty.
0: receive FIFO is empty 1: receive FIFO is not empty

TNF Transmit FIFO not full.
0: transmit FIFO is full 1: transmit FIFO is not full

TFE Transmit FIFO empty.
0: transmit FIFO is not empty 1: transmit FIFO is empty

0 Reserved for future use. Reading returns 0. Must be written with 0.

SSP_CPSR SSP clock prescale register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	CPSDVSR							
R	R	R	R	R	R	R	R	RW							

Address: SSPBaseAddress + 0x010

Reset: 0x0000 0000

Description: Specifies the division factor by which the internal peripheral clock SSPCLK must be divided before further use. The value programmed into this register must be an even number between 2 and 254. The least significant bit of the programmed number is

hard-coded to zero, therefore if an odd number is written to this register, data read back from this register has the LSB as zero.

CPSDVSR Clock prescale divisor. The value is used to generate the transmit and receive bit rate of the SSP. The bit rate is:

$F_{SSPCLKI}$ / [CPSDVR x (1 + SCR)] where SCR = 0 to 255, programmed through *SSP_CR0*.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SSP_IMSC SSP interrupt mask set/clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	TXIM	RXIM	RTIM	RORIM
R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Address: SSPBaseAddress + 0x014

Reset: 0x0000 0000

Description: Returns the current value of the mask on the relevant interrupt. A write of 1 to a particular bit sets the mask, enabling the associated interrupt to be read. A write of 0 clears the corresponding mask. All bits are cleared to 0 when reset.

TXIM Transmit interrupt mask.
0: transmit interrupt is masked 1: transmit interrupt is not masked

RXIM Receive interrupt mask.
0: receive condition interrupt is masked 1: receive interrupt is not masked

RTIM Receive timeout interrupt mask.
0: Rx FIFO is not empty and no read prior to timeout period interrupt is masked.
1: Rx FIFO is not empty and no read prior to timeout period interrupt is not masked.

RORIM Receive overrun interrupt mask.
0: Rx FIFO is written to while full condition interrupt is masked.
1: Rx FIFO is written to while full condition interrupt is not masked.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SSP_RIS SSP raw interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	TXRIS	RXRIS	RTRIS	RORRIS
R	R	R	R	R	R	R	R	R	R	R	R	RH	RH	RH	RH

Address: SSPBaseAddress + 0x018

Reset: 0x0000 0008

Description: Returns the current raw status of the corresponding interrupt prior to masking. A write has no effect.

TXRIS Transmit raw interrupt status. Indicates the raw interrupt state, prior to masking, of the transmit interrupt.

RXRIS Receive raw interrupt status. Indicates the raw interrupt state, prior to masking, of the receive interrupt.

RTRIS Receive timeout raw interrupt status. Indicates the raw interrupt state, prior to masking, of the receive timeout interrupt.

RORRIS Receive overrun raw interrupt status. Indicates the raw interrupt state, prior to masking, of the receive overrun interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SSP_MIS SSP masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	TXMIS	RXMIS	RTMIS	RORMIS
R	R	R	R	R	R	R	R	R	R	R	R	RH	RH	RH	RH

Address: SSPBaseAddress + 0x01C

Reset: 0x0000 0000

Description: Returns the current masked status of the corresponding interrupt. A write has no effect.

TXMIS Transmit masked interrupt status. Indicates the interrupt state after masking of the transmit interrupt.

RXMIS Receive masked interrupt status. Indicates the interrupt state after masking of the receive interrupt.

RTMIS Receive timeout masked interrupt status. Indicates the interrupt state after masking of the receive timeout interrupt.

RORMIS Receive overrun masked interrupt status. Indicates the interrupt state after masking of the receive overrun interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SSP_ICR SSP interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	RTIC	RORIC
R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Address: SSPBaseAddress + 0x020

Reset: 0x0000 0000

Description: On a write of 1, the corresponding interrupt is cleared. A write of 0 has no effect.

RTIC Receive timeout clear interrupt. When set to 1, clears the receive timeout interrupt

RORIC Receive overrun raw clear interrupt. When set to 1, clears the receive overrun interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

SSP_DMACR SSP DMA control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	TX DMAE	RX DMAE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Address: SSPBaseAddress + 0x024

Reset: 0x0000 0000

Description: All bits are cleared to 0 on reset.

TXDMAE Transmit DMA enable. When set to 1, the DMA for the transmit FIFO is enabled.

RXDMAE Receive DMA enable. When set to 1, the DMA for the receive FIFO is enabled.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SSP_PeriphID0 SSP peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: SSPBaseAddress + 0xFE0

Reset: 0x0000 0022

Description: PartNumber0 reads back as 0x22.

SSP_PeriphID1 SSP peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: SSPBaseAddress + 0xFE4

Reset: 0x0000 0000

Description: Designer0 reads back as 0x00. PartNumber1 reads back as 0x00.

SSP_PeriphID2 SSP peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: SSPBaseAddress + 0xFE8

Reset: 0x0000 0008

Description: Revision returns the revision number of the peripheral. Designer1 reads back as 0x08.

SSP_PeriphID3 SSP peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: SSPBaseAddress + 0xFEC

Reset: 0x0000 0001

Description: Configuration reads back as 0x01 (32-bit data path).

SSP_PCellID0 SSP Pcell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SSPPCellID0							
R	R	R	R	R	R	R	R	R							

Address: SSPBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: SSPPCELLID0 reads back as 0x0D.

SSP_PCellID1 SSP Pcell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SSPPCellID1							
R	R	R	R	R	R	R	R	R							

Address: SSPBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: SSPPCELLID1 reads back as 0xF0.

SSP_PCellID2 SSP Pcell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SSPPCellID2							
R	R	R	R	R	R	R	R	R							

Address: SSPBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: SSPPCELLID2 reads back as 0x05.

SSP_PCellID3 SSP Pcell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SSPPCellID3							
R	R	R	R	R	R	R	R	R							

Address: SSPBaseAddress + 0xFFC

Reset: 0x0000 001

Description: SSPPCELLID3 reads back as 0xB1.

23 Multichannel serial ports (MSP0,1,2,3)

23.1 MSP register addressing

Register addresses are provided as the MSP base address, MSPnBaseAddress, plus the register offset.

There are 4 MSP base addresses.

Table 47. MSP base addresses

MSP base address	MSP
0x101F 9000	0
0x101F 1000	1
0x101F 0000	2
0x101C 0000	3

MSP 0, 1, 2 and 3 are referred to as MSPn throughout this document.

23.2 MSP register summary

The device communicates to the MSP via 32-bit wide control registers accessible via the 32-bit width APB). All four MSP units are identical blocks, but their implementation differs as follows:

- MSP0 provides all 7 signals (MSPTXD0, MSPRXD0, MSPTFS0, MSPRFS0, MSPTCK0, MSPRCK0 and MSPSCK0).
- MSP1 provides 5 signals (MSPTXD1, MSPRXD1, MSPTFS1, MSPTCK1 and MSPSCK1), it does not provide MSPRFS1 and MSPRCK1. Inside the device, MSPRFS1 input is wired to MSPTFS1 (input) (MSPRFS1 cannot be programmed as output), and MSPRCK1 input is wired to MSPTCK1 (input) (MSPRCK1 cannot be programmed as output).
- MSP2 provides 4 signals (MSPTXD2, MSPRXD2, MSPTFS2 and MSPTCK2), it does not provide MSPRFS2, MSPRCK2 AND MSPSCK2. Inside the device, MSPRFS2 input is wired to MSPTFS2 (input or output) (MSPRFS2 cannot be programmed as output), and MSPRCK2 input is wired to MSPTCK2 (input) (MSPRCK2 cannot be programmed as output). MSPSCK2 is grounded, thus can not be used to clock the baud rate generator.
- MSP3 does not provide MSPRFS3, MSPRCK3 AND MSPSCK3 signals. Inside the device, MSPRFS3 input is wired to MSPTFS3 (input or output) (MSPRFS3 cannot be programmed as output), and MSPRCK3 input is wired to MSPTCK3 (input) (MSPRCK3 cannot be programmed as output). MSPSCK3 is grounded, thus can not be used to clock the baudrate generator.

To allow simultaneous internal data movement and external data communications, five registers are used that are not directly accessible on the APB. These are the receive buffer register, receive and transmit shift registers, and receive and transmit enable shift registers (MSPn_RBR, MSPn_RSR, MSPn_TSR, MSPn_TSE, MSPn_RSE). An APB bus master reads the received data from the data receive register (*MSPn_DR*, referred to as **MSPn_RDR** in this situation) and writes the data to be transmitted to the data transmit

register (*MSPn_DR*, referred to as **MSPn_TDR** in this situation). Data written to the MSPn_DR is shifted out to MSPTXDx via the transmit shift register (MSPn_TSR). Similarly, receive data on the MSPRXDx pin is shifted into the receive shift register (MSPn_RSR) and copied into the receive buffer register (MSPn_RBR). MSPn_RBR is then copied to MSPn_RDR, which can be read by the APB master. Other registers allow the control mechanism of the MSP to be configured.

The control block provides internal clock generation, frame-synchronization signal generation, multichannel selection and channel enabling. *Table 48* lists the MSP register details.

Table 48. MSP register list

Offset	Register name	Description	Page
0x000	MSPn_DR	Data register	*297*
0x004	MSPn_GCR	Global control register	*298*
0x008	MSPn_TCF	Transmit configuration register	*300*
0x00C	MSPn_RCF	Receive configuration register	*301*
0x010	MSPn_SRG	Sample rate generator control register	*303*
0x014	MSPn_FLR	Flag register	*304*
0x018	MSPn_DMACR	DMA control register	*304*
0x020	MSPn_IMSC	Interrupt mask set and clear register	*305*
0x024	MSPn_RIS	Raw interrupt status register	*306*
0x028	MSPn_MIS	Masked interrupt status register	*307*
0x02C	MSPn_ICR	Interrupt clear register	*308*
0x030	MSPn_MCR	Multichannel control register	*308*
0x034	MSPn_RCV	Receive comparison value register	*310*
0x038	MSPn_RCM	Receive comparison mask register	*310*
0x040	MSPn_TCE0	Transmit channel enable register 0	*311*
0x044	MSPn_TCE1	Transmit channel enable register 1	*311*
0x048	MSPn_TCE2	Transmit channel enable register 2	*312*
0x04C	MSPn_TCE3	Transmit channel enable register 3	*312*
0x060	MSPn_RCE0	Receive channel enable register 0	*312*
0x064	MSPn_RCE1	Receive channel enable register 1	*313*
0x068	MSPn_RCE2	Receive channel enable register 2	*313*
0x06C	MSPn_RCE3	Receive channel enable register 3	*313*
0xFE0	MSPnPeriphID0	Peripheral identification register 0 (bits 7:0)	*314*
0xFE4	MSPnPeriphID1	Peripheral identification register 1 (bits 15:8)	*314*
0xFE8	MSPnPeriphID2	Peripheral identification register 2 (bits 23:16)	*314*
0xFEC	MSPnPeriphID3	Peripheral identification register 3 (bits 31:24)	*315*
0xFF0	MSPnPCellID0	PCell identification register 0 (bits 7:0)	*315*

Table 48. MSP register list

Offset	Register name	Description	Page
0xFF4	MSPnPCellID1	PCell identification register 1 (bits 15:8)	315
0xFF8	MSPnPCellID2	PCell identification register 2 (bits 23:16)	316
0xFFC	MSPnPCellID3	PCell identification register 3 (bits 31:24)	316
	MSPn_RBR MSPn_RSR MSPn_TSR MSPn_TSE MSPn_RSE	Receive buffer register Receive shift register Transmit shift register Transmit enable shift register Receive enable shift register	(1)

1. The MSPn_TSR, MSPn_RSR, MSPn_RBR, MSPn_TSE and MSPn_RSE registers are not directly accessible on the APB bus.

23.3 MSP register descriptions

MSPn_DR MSP data register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA															
RW															

Address: MSPnBaseAddress + 0x000

Reset: 0x0000 0000

Description: When this register is read, the entry in the receive FIFO (pointed to by the current FIFO read pointer) is accessed. As data values are removed by the MSP receive logic from the incoming data buffer, they are placed in the entry in the receive FIFO (pointed to by the current FIFO write pointer).

When this register is written to, the entry in the transmit FIFO (pointed to by the write pointer) is written to. Data values are removed from the transmit FIFO one value at a time by the transmit logic. It is loaded into the transmit serial shifter, then serially shifted out onto the MSPTXDx pin at the programmed bit rate.

When a data size of less than 32 bits is selected, the user must right-justify data written to the Tx FIFO, the transmit logic ignores unused bits. Received data less than 32 bits is automatically right-justified in the Rx FIFO.

The Tx FIFO and Rx FIFO are not cleared even when *MSPn_GCR*.TXEN (transmitter enable) and *MSPn_GCR*.RXEN (receiver enable) are cleared. This allows the software to fill the Tx FIFO before enabling the transmitter.

DATA[31:0] Transmit or receive data.

Read: Rx FIFO is read Write: Tx FIFO is written

MSPn_GCR MSP global configuration register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	SPI BME	SPICKM		FGEN	SCKSEL		SCK POL	SGEN
R	R	R	R	R	R	R	R	RW	RW		RW	RW		RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXDDL	TCK SEL	TCK POL	TFSSEL		TFS POL	TFFEN	TXEN	LBM	RCK SEL	RCK POL	RFS SEL	DCM	RFS POL	RFFE N	RXEN
RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MSPnBaseAddress + 0x004

Reset: 0x0000 0000

Description: The MSP is configured via this 32-bit serial port control register, which also contains MSP status bits.

SPIBME SPI burst mode enable. In SPI mode, MSPTFSx acts as a slave select (SSnot) signal.

0: normal mode enabled (MSPTFSx toggles during back-to-back transfers).

1: burst mode enabled (MSPTFSx always low during back-to-back transfers).

SPICKM SPI clock modes. Enables SPI clock modes. In conjunction with TCKPOL, it provides compatibility with various SPI protocols.

0X: normal clocking mode for non SPI protocols.

10: zero delay SPI mode enabled. The clock starts without delay on the rising edge when TCLKP = 0, or on the falling edge when TCLKP = 1.

11: half cycle delay SPI mode enabled. The clock starts with rising, TCLKP = 0: or falling, TCLKP = 1: edge with a half clock cycle delay.

FGEN Frame generator enable. Enables the frame sync generator.

0: disabled, stopped and in reset 1: enabled, running

SCKSEL Sample rate generator clock selection selects the sample rate generator clock source.

0X: sample rate generator clock is the internal clock signal, MSPCLK signal, 48 MHz.

10: sample rate generator clock is derived from the MSPSCK pin and is free-running.

11: sample rate generator clock is derived from the MSPSCK pin but is resynchronized on MSPRFS, receive frame sync signal, transition to active level. MSPRFS is an input pin, regardless of the value of the RFSSEL bit.

SCKPOL Sample rate generator clock polarity. Used only when MSPSCK clocks the sample rate generator, SCKSEL = 1X. Indicates which edge of MSPSCK clocks the sample rate generator.

0: rising edge 1: falling edge

SGEN Sample rate generator enable. Enables the sample rate generator.

0: disabled, stopped and in reset 1: enabled, running

TXDDL Transmit data extra delay. Enables an extra delay for switching the MSPTXD pin from HiZ to active. It only controls the output enable of the MSPTXD pin, not the data itself.

0: extra delay on MSPTXD output enable is off. 1: extra delay on MSPTXD output enable is on.

TCKSEL Transmit clock selection.

0: if not in SPI mode (SPICKM = 0X) the transmit clock is provided by an external source on the MSPTCK pin. MSPTCK is an input pin. If in SPI mode (SPICKM = 1X) the transmit clock is provided by an external SPI master on the MSPTCK pin, and MSP is then an SPI slave. MSPRCK is internally connected to MSPTCK, which is then an input pin.

1: if not in SPI mode (SPICKM = 0X), the transmit clock is provided by the internal sample rate generator. MSPTCK is then an output pin. If SPI mode (SPICKM = 1X), the transmit clock is provided by the internal sample rate generator, and MSP is then an SPI master. MSPRCK is internally connected to MSPTCK, which is then an output pin.

TCKPOL Transmit clock polarity. Indicates on which edge of MSPTCK the transmit data is driven.

0: rising edge 1: falling edge

TFSSEL Transmit frame synchronization selection.

0X: the frame synchronization signal is provided by an external source, MSPTFS is then an input pin.

10: the frame synchronization signal is provided by frame generator logic each time MSPn_TDR (transmit data register) is copied to MSPn_TSR (transmit shift register). MSPTFS is then an output pin.

11: the frame synchronization signal is provided by frame generator logic, with period and width defined by FRPER and FRWID values in the *MSPn_SRG* register. MSPTFS is then an output pin.

TFSPOL Transmit frame synchronization polarity. Indicates when MSPTFS pulse is active.

0: high 1: low

TFFEN Tx FIFO enable. Enables the Tx FIFO buffer.

0: Tx FIFO is disabled, the Tx FIFO is then a 1-word deep holding register.

1: Tx FIFO is enabled.

TXEN Transmitter enable. Enables the transmitter.

0: transmitter is disabled, stopped and in reset. 1: transmitter is enabled and running.

LBM Loopback mode. Enables loop back mode. In loop back mode, MSPRXD, MSPRFS and MSPRCK are internally connected, through multiplexers, to MSPTXD, MSPTFS and MSPTCK respectively. Loopback mode can be used to test serial port code with a single MSP.

0: loopback mode disabled 1: loopback mode enabled

RCKSEL Receive clock selection.

0: if loopback mode is not set, LBM = 0, the receive clock is provided by an external source on the input pin MSPRCK. If loopback mode is set, LBM = 1, the receive clock is provided by the transmit clock and MSPRCK is an unused pin, in high impedance.

1: if loopback mode is not set, LBM = 0, the receive clock is provided by the sample rate generator and delivered on the output pin MSPRCK. If loopback mode is set, LBM = 1, the receive clock is provided by the transmit clock and delivered on the output pin MSPRCK.

RCKPOL Receive clock polarity. Indicates when the received data is sampled.

0: on falling rising edge of MSPRCK 1: on rising edge of MSPRCK

RFSSEL Receive frame synchronization selection.

0: receive frame synchronization signal is provided by an external source. MSPRFS is an input pin.

1: receive frame synchronization signal is provided by frame generator logic. MSPRFS is an output pin except when SCKSEL = 11.

DCM Direct companding mode. Enables the direct companding mode, which uses the companding hardware as a coprocessor. Both the transmitter and receiver must be disabled, TXEN = 0 and RXEN = 0, or the DCM value is ignored. Data in the Tx FIFO is compressed, sent directly to the receiver expanding logic and is then written in the Rx FIFO.

0: direct companding mode is disabled. 1: enabled if TXEN = RXEN = 0.

RFSPOL Receive frame synchronization polarity. Indicates when the MSPRFS pulse is active.
0: high 1: low

RFFEN Rx FIFO enable. Enables the Rx FIFO buffer.
0: Rx FIFO is disabled (the FIFO becomes a 1-word deep holding register).
1: Rx FIFO is enabled.

RXEN Receiver enable. Enables the receiver.
0: disabled (stopped and in reset) 1: enabled (running)

0 Reserved for future use. Reading returns 0. Must be written with 0.

MSPn_TCF MSP transmit configuration register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	TBSWAP		TP2EN	TP2SM	TP2FLEN							TP2ELEN		
R	R	RW		RW	RW	RW							RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TFSIG	TDDLY		TENDN	TDTYP		TP1FLEN							TP1ELEN		
RW	RW		RW	RW		RW							RW		

Address: MSPnBaseAddress + 0x008

Reset: 0x0000 0000

Description: Configures parameters for transmit operations.

TBSWAP Transmit byte swapping. Configures byte swapping on a word between the Tx FIFO and Tx shift register:
00: no byte swapping.
01: little to big endian byte swap in a word, for example word 0xDCBA produces 0xABCD.
10: little to big endian byte swap in each halfword, for example word 0xDCBA produces 0xCDAB.
11: halfword swap, for example word 0xDCBA produces 0xBADC.

TP2EN Transmit phase 2 enable. Disables or enables the dual phase frame.
0: single phase frame, phase 2 is disabled 1: dual phase frame, phase 2 is enabled

TP2SM Transmit phase 2 start mode. Defines when phase 2 starts. Intended for I2S configuration.
0: immediately after end of phase 1.
1: after the transmit frame synchronization transitions to the opposite edge that started phase 1.

TP2FLEN Transmit phase 2 frame length. Defines the number (- 1) of elements transferred in the phase 2 frame:
TP2FLEN = <number of element in transmit phase 2> - 1.
000 0000: 1 element in phase 2 000 0001: 2 elements in phase 2, until
111 1111: 128 elements in phase 2.

TP2ELEN Transmit phase 2 element length. Defines the number of bits per element in the phase 2 frame:
000: 8 bit-element 001: 10-bit element
010: 12-bit element 011: 14-bit element
100: 16-bit element 101: 20-bit element
110: 24-bit element 111: 32-bit element

TFSIG Transmit frame sync ignore. Indicates a response to transmit frame synchronization pulses.
0: all transmit frame synchronization pulses, even unexpected, restart the transfer.
1: ignore unexpected transmit frame synchronization pulses.

TDDLY Transmit data delay. Defines the number of bit clock cycles between the transmit frame sync activation and the first data bit transmitted.

00: data delay is 0 01: data delay is 1 bit clock cycle

10: data delay is 2 bit clock cycles 11: data delay is 3 bit clock cycles

TENDN Transmit bit endian format determines whether the element is transmitted MSB first or LSB first.

0: MSB first 1: LSB first

TDTYP Transmit data type.

0X: companding disabled.

10: companding enabled, using μ-law. The 32-bit signed data written in the MSPn_TDR register is compressed to 8-bit data, right justified, before it is transmitted. If the 32-bit data is outside the 14-bit μ-law dynamic range, it is automatically compressed to the most positive or most negative value.

11: companding enabled, using A-law. The 32-bit signed data written in the MSPn_TDR register is compressed to 8-bit data, right justified, before it is transmitted. If the 32-bit data is outside the 13-bit A-law dynamic range, it is automatically compressed to the most positive or most negative value.

TP1FLEN Transmit phase 1 frame length. Defines the number (- 1), of elements transferred in phase 1 frame.

TP1FLEN = <number of element in transmit phase 1> - 1

000 0000: 1 element in phase 1, 000 0001: 2 elements in phase 1 until

111 1111: 128 elements in phase 1.

TP1ELEN Transmit phase 1 element length. Defines the number of bits per element in phase 1 frame:

000: 8-bit element length in phase 1 001: 10-bit element length in phase 1

010: 12-bit element length in phase 1 011: 14-bit element length in phase 1

100: 16-bit element length in phase 1 101: 20-bit element length in phase 1

110: 24-bit element length in phase 1 111: 32-bit element length in phase 1

0 Reserved for future use. Reading returns 0. Must be written with 0.

MSPn_RCF MSP receive configuration register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	RBSWAP		RP2EN	RP2SM	RP2FLEN							RP2ELEN		
R	R	RW		RW	RW	RW							RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFSIG	RDDLY		REND N	RDTYP		RP1FLEN							RP1ELEN		
RW	RW		RW	RW		RW							RW		

Address: MSPnBaseAddress + 0x00C

Reset: 0x0000 0000

Description: Configures parameters for receive operations.

RBSWAP Receive halfword swapping. Configure byte swapping on a word between Rx receive buffer and the Rx FIFO:

00: no byte swapping.

01: little to big endian byte swap in a word, word 0xDCBA produces 0xABCD.

10: little to big endian byte swap in each halfword, word 0xDCBA produces 0xCDAB.

11: halfword swap, word 0xDCBA produces 0xBADC.

RP2EN Receive phase 2 enable. Disables or enables dual phase frame.

0: single phase frame, phase 2 is disabled 1: dual phase frame, phase 2 is enabled

RP2SM Receive phase 2 start mode. Defines when phase 2 starts. Mainly intended for i2s configuration.

0: starts immediately after end of phase 1.

1: starts after receive frame synchronization transitions to the opposite edge that started phase 1.

RP2FLEN Receive phase 2 frame length. Defines the number (- 1) of elements received in the phase 2 frame:

RP2FLEN = <number of element in transmit phase 2> - 1

000 0000: 1 element in phase 2, 000 0001: 2 elements in phase 2 and so on until

111 1111: 128 elements in phase 2.

RP2ELEN Receive phase 2 element length. Defines the number of bits per element in the phase 2 frame:

000: 8-bit element length in phase 2 001: 10-bit element length in phase 2

010: 12-bit element length in phase 2 011: 14-bit element length in phase 2

100: 16-bit element length in phase 2 101: 20-bit element length in phase 2

110: 24-bit element length in phase 2 111: 32-bit element length in phase 2

RFSIG Receive frame sync ignore.

0: all receive frame synchronization pulses (even unexpected) restart the transfer.

1: unexpected receive frame synchronization pulses are ignored.

RDDLY Receive data delay. Defines the number of bit clock cycles between the receive frame sync activation and first data bit received.

00: data delay is 0 01: data delay is 1-bit clock cycle

10: data delay is 2-bit clock cycles 11: data delay is 3-bit clock cycles

RENDN Receive bit endian format. Defines whether the element is received MSB first or LSB first.

0: the element is received MSBit first. 1: the element is received LSBit first.

RDTYP Receive data type.

00: companding disabled, received data is right justified, unused MSBits of MSPn_RDR are zero filled.

01: companding disabled, received data is right justified, unused MSBits of MSPn_RDR are signed extended.

10: companding enabled, using μ-law, the data in MSPn_RDR data receive register is right justified, sign extended expanded, with μ-law, value of the 8 LSBits received.

11: companding enabled, using A-law, the data in MSPn_RDR data receive register is right justified, sign extended expanded, with A-law, value of the 8 LSBits received.

RP1FLEN Receive phase 1 frame length. Defines the number (- 1), of elements received in phase 1 frame.
RP1FLEN = <number of element in receive phase 1> - 1.
000 0000: 1 element in phase 1,
000 0001: 2 elements in phase 1, and so on until
111 1111: 128 elements in phase 1.

RP1ELEN Receive phase 1 element length. Defines the number of bits per element in phase 1 frame:
000: 8-bit element length in phase 1
001: 10-bit element length in phase 1
010: 12-bit element length in phase 1
011: 14-bit element length in phase 1
100: 16-bit element length in phase 1
101: 20-bit element length in phase 1
110: 24-bit element length in phase 1
111: 32-bit element length in phase 1

0 Reserved for future use. Reading returns 0. Must be written with 0.

MSPn_SRG MSP sample rate generator register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	FRPER												
R	R	R	RW												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRWID						SCKDIV									
RW						RW									

Address: MSPnBaseAddress + 0x010

Reset: 0x0000 0000

Description: Controls the operation of a range of features of the sample rate generator.

FRPER Frame period. FRPER+1 specifies the frame length, from 1 to 8192 serial data bit clock cycles, when the sample rate generator is used to deliver the transmit or receive frame sync signals (SCKSEL = 10 in *MSPn_GCR* register).

FRWID Frame width. FRWID+1 specifies the active frame pulse width, from 1 to 64 serial data bit clock cycles, when the sample rate generator is used to deliver the transmit or receive frame sync signals (SCKSEL = 1X in MSPn_GCR register).

SCKDIV Sample rate generator clock divide factor. SCKDIV+1 specifies how many times the sample rate generator input clock (SCLK_INT) is divided, to produce the serial data bit clock (MSPSCK). SCLK_INT is either the MSP clock (MSPCLK = 48 MHz) or external clock signal on the MSPSCK pin, depending on the SCKSEL setting in the *MSPn_GCR* register.
MSPSCK frequency = SCLK_INT frequency / (SCKDIV + 1).
SCKDIV valid values are in the range 0 to 1023.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MSPn_FLR MSP flag register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	TFU	TFE	TBUSY	RFU	RFE	RBUSY
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: MSPnBaseAddress + 0x014

Reset: 0x0000 0012

Description: Provides status for the MSP FIFOs and transmit/receive shifters.

TFU Tx FIFO full flag. If Tx FIFO is full and FIFO is disabled, this means that the holding register is full.
0: Tx FIFO not full 1: Tx FIFO full

TFE Tx FIFO empty flag. If the Tx FIFO is not empty and the FIFO is disabled, the holding register is full.
0: Tx FIFO not empty 1: Tx FIFO empty

TBUSY Transmit busy flag. Indicates if shift register (MSPn_TSR) has an element which must be shifted out.
0: transmitter is idle.
1: MSPn_TSR contains an element ready to be shifted out or is in the process of being shifted out.

RFU Rx FIFO full flag.
0: Rx FIFO not full 1: the Rx FIFO full

RFE Rx FIFO empty flag. If the Rx FIFO is not empty and the FIFO is disabled, the holding register is full.
0: Rx FIFO not empty 1: Rx FIFO empty

RBUSY Receiver busy flag.
0: receiver is idle and the Rx FIFO is empty. 1: receiver is currently receiving an element.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MSPn_DMACR MSP DMA control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	TDMAE	RDMAE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW

Address: MSPnBaseAddress + 0x018

Reset: 0x0000 0000

Description: This read/write register is the DMA control register. All bits are cleared to 0 on reset.

TDMAE Transmit DMA enable.
1: DMA is enabled for the Tx FIFO.

RDMAE Receive DMA enable.
1: DMA is enabled for the Rx FIFO.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MSPn_IMSC MSP interrupt mask set/clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	TFOIM	RFOIM	TFSIM	TSEIM	TUEIM	TXIM	RFSIM	RSEIM	ROEIM	RXIM
R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MSPnBaseAddress + 0x020

Reset: 0x0000 0000

Description: On a read, returns the current value of the mask on the associated interrupt. A write of 1 sets the mask, enabling the interrupt to be read. A write of 0 clears the corresponding mask.

TFOIM Tx FIFO not full interrupt mask.
0: masked 1: not masked

RFOIM Rx FIFO not empty interrupt mask.
0: masked 1: not masked

TFSIM Transmit frame sync MSPTFSINTR interrupt mask status.
0: masked 1: not masked

TSEIM Transmit frame synchronization error interrupt mask status.
0: masked 1: not masked

TUEIM Transmit underrun error interrupt mask status.
0: masked 1: not masked

TXIM Transmit MSPTXINTR interrupt mask status.
0: masked 1: not masked

RFSIM Receive frame sync MSPRFSINTR interrupt mask status.
0: masked 1: not masked

RSEIM Receive frame synchronization error interrupt mask status.
0: masked 1: not masked

ROEIM Receive overrun error interrupt mask status.
0: masked 1: not masked

RXIM Receive MSPRXINTR interrupt mask status.
0: masked 1: not masked

0 Reserved for future use. Reading returns 0. Must be written with 0.

MSPn_RIS MSP raw interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	TFO RIS	RFO RIS	TFS RIS	TSE RIS	TUE RIS	TX RIS	RFS RIS	RSE RIS	ROE RIS	RX RIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: MSPnBaseAddress + 0x024

Reset: 0x0000 0210

Description: On a read, returns the current raw status of the corresponding interrupt prior to masking. A write has no effect.

TFORIS TxFIFO not full raw interrupt status. Returns the raw interrupt state, prior to masking, of the TxFIFO not full interrupt.

RFORIS RxFIFO not empty raw interrupt status. Returns the raw interrupt state, prior to masking, of the RxFIFO not empty interrupt.

TFSRIS Transmit frame sync raw interrupt status. Returns the raw interrupt state, prior to masking, of the transmit frame sync interrupt.

TSERIS Transmit frame synchronization error raw interrupt status. Returns the raw interrupt state, prior to masking, of the transmit frame sync error interrupt.

TUERIS Transmit underrun error raw interrupt status. Returns the raw interrupt state, prior to masking, of the transmit underrun error interrupt.

TXRIS TxFIFO service raw interrupt status. Returns the raw interrupt state, prior to masking, of the TxFIFO service interrupt.

RFSRIS Receive frame sync raw interrupt status. Returns the raw interrupt state, prior to masking, of the receive frame sync interrupt.

RSERIS Receive frame synchronization error raw interrupt status. Returns the raw interrupt state prior to masking of the receive frame sync error interrupt.

ROERIS Receive overrun error raw interrupt status. Returns the raw interrupt state, prior to masking, of the receive overrun error interrupt.

RXRIS RxFIFO service raw interrupt status. Returns the raw interrupt state, prior to masking, of the RxFIFO service interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MSPn_MIS MSP masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	TFOMIS	RFOMIS	TFSMIS	TSEMIS	TUEMIS	TXMIS	RFSMIS	RSEMIS	ROEMIS	RXMIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: MSPnBaseAddress + 0x028

Reset: 0x0000 0000

Description: Returns the current masked status of the corresponding interrupt. A write has no effect.

TFOMIS Tx FIFO not full masked interrupt status. Returns the masked interrupt state of the Tx FIFO not full interrupt.

RFOMIS Rx FIFO not empty masked interrupt status. Returns the masked interrupt state of the Rx FIFO not empty interrupt

TFSMIS Transmit frame sync masked interrupt status. Returns the masked interrupt state of the transmit frame sync interrupt.

TSEMIS Transmit frame synchronization error masked interrupt status. Returns the masked interrupt state of the transmit frame sync error interrupt.

TUEMIS Transmit underrun error masked interrupt status. Returns the masked interrupt state of the transmit underrun error interrupt.

TXMIS Tx FIFO service masked interrupt status. Returns the masked interrupt state of the transmit interrupt.

RFSMIS Receive frame sync masked interrupt status. Returns the masked interrupt state of the receive frame sync interrupt.

RSEMIS Receive frame synchronization error masked interrupt status. Returns the masked interrupt state of the receive frame sync error interrupt.

ROEMIS Receive overrun error masked interrupt status. Returns the masked interrupt state of the receive overrun error interrupt.

RXMIS Rx FIFO service masked interrupt status. Returns the masked interrupt state of the receive interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MSPn_ICR MSP interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	TFSIC	TSEIC	TUEIC	0	RFSIC	RSEIC	ROEIC	0
R	R	R	R	R	R	R	R	RW	RW	RW	R	RW	RW	RW	R

Address: MSPnBaseAddress + 0x02C

Reset: 0x0000 0000

Description: A write of 0 has no effect. Reads return zero.

TFSIC Transmit frame sync interrupt clear.
Writing a 1 clears the transmit frame sync interrupt, reading returns 0.

TSEIC Transmit frame synchronization error interrupt clear.
Writing a 1 clears the transmit frame sync error interrupt, reading returns 0.

TUEIC Transmit underrun error interrupt clear.
Writing a 1 clears the transmit underrun error interrupt, reading returns 0.

RFSIC Receive frame sync interrupt clear.
Writing a 1 clears the receive frame sync interrupt, reading returns 0.

RSEIC Receive frame synchronization error interrupt clear.
Writing a 1 clears the receive frame sync error interrupt, reading returns 0.

ROEIC Receive overrun error interrupt clear.
Writing a 1 clears the receive overrun error interrupt, reading returns 0.

0 Reserved for future use. Reading returns 0. Must be written with 0.

MSPn_MCR MSP multichannel control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	TMCSF		TMCEN	RMCMP		RMCSF		RMCEN
R	R	R	R	R	R	R	R	R		RW	RW		R		RW

Address: MSPnBaseAddress + 0x030

Reset: 0x0000 0000

Description: Enables multichannel operations for the transmitter and receiver when they are configured to single-phase frame. *MSPn_TCF*.TP2EN/ RP2EN must be set to 0 for the transmitter/receiver to operate in multichannel mode.

TMCSF Transmit multichannel current subframe. Indicates the number of the currently transmitting subframe when TMCEN = 1. Otherwise it is not defined.

00: subframe 0 is current (element 0 to 31) 01: subframe 1 is current (element 32 to 63)

10: subframe 2 is current (element 64 to 95) 11: subframe 3 is current (element 96 to 127)

TMCEN Transmit multichannel mode enable. Enables the transmitter to operate in multichannel mode, providing that *MSPn_TCF*.TP2EN = 0; if TP2EN = 1, then TMCEN is not relevant, and multichannel mode is disabled.

0: multichannel mode disabled, all channels are enabled, registers *MSPn_TCE0,1,2,3* are not used. The MSPTXD pin is always driven during transmission of data.

1: multichannel mode enabled, channels are enabled or disabled according to corresponding bits in the *MSPn_TCEx* registers. The MSPTXD pin is only driven during transmission of channels that are enabled and corresponding bit TCEx = 1.

RMCMP Receive multichannel comparison mode. Controls the mode for the receive multichannel bit-to-bit comparison, between the received data and *MSPn_RCV*. A bit is masked if the corresponding bit is set in register *MSPn_RCM*.

0X: receive multichannel comparison is disabled. The channels for the current subframe are enabled or disabled according to the corresponding bits in the *MSPn_RCE* register.

10: receive multichannel comparison is enabled. For the current subframe, the enabled channels in MSPn_RCE are accepted if the bit-to-bit comparison of the received data with MSPn_RCV (for bits not masked by MSPn_RCM) is false.

11: receive multichannel comparison is enabled. For the current subframe, the enabled channels in MSPn_RCE are accepted if the bit-to-bit comparison of the received data with MSPn_RCV (for bits not masked by MSPn_RCM) is true.

RMCSF Receive multichannel current subframe reveals the number of the currently receiving subframe when RMCEN = 1. Otherwise, it is not defined.

00: current is subframe 0 (element 0 to 31). 01: current is subframe 1 (element 32 to 63).

10: current is subframe 2 (element 64 to 95). 11: current is subframe 3 (element 96 to 127).

RMCEN Receive multichannel mode enable. The receiver is in multichannel mode if *MSPn_RCF*.RP2EN = 0. If MSPn_RCF.RP2EN = 1, RMCEN is not relevant and multichannel mode is disabled.

0: receive multichannel mode disabled, all receive channels enabled, registers MSPn_RCEx not used.

1: receive multichannel mode enabled, channels are enabled or disabled according to the corresponding bits in MSPn_RCEx registers. In addition, enabled channels can be received only if their value matches the content of the multichannel comparison value (MSPn_RCV) for the bits that are not masked by the multichannel comparison mask (MSPn_RCM).

0 Reserved for future use. Reading returns 0. Must be written with 0.

MSPn_RCV MSP receive compare value register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCV31	RCV30	RCV29	RCV28	RCV27	RCV26	RCV25	RCV24	RCV23	RCV22	RCV21	RCV20	RCV19	RCV18	RCV17	RCV16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCV15	RCV14	RCV13	RCV12	RCV11	RCV10	RCV9	RCV8	RCV7	RCV6	RCV5	RCV4	RCV3	RCV2	RCV1	RCV0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MSPnBaseAddress + 0x034

Reset: 0x0000 0000

Description: In multichannel mode (*MSPn_MCR*.RMCEN = 1), if receive multichannel comparison mode is enabled (MSPn_MCR.RMCMP = 1X), the received data for any enabled channel (that is, with its corresponding bit set in MSPn_RCEx) is compared bit-to-bit with the data in this register (*MSPn_RCV*) for bits that are not masked in *MSPn_RCM*. When the comparison result is false (MSPn_MCR.RMCMP = 10), or true (MSPn_MCR.RMCMP = 11), the received data is accepted, MSPn_RBR is copied to MSPn_RDR, the RFU and RFE flags are updated, and the receive interrupt is set.

```
Comparison result (TRUE or FALSE):=
   [(<MSP_RBR> AND NOT<MSP_RCM>) == (<MSP_RCV> AND
NOT<MSP_RCM>]
```

Note: Comparison occurs on the received data before expansion if the companding hardware is enabled.

RCV[31:0] Receive compare value bit x. Only used when receive comparison mode is enabled (*MSPn_MCR*.RMCMP = 1X). Defines the value of bit x to which bit x of MSPn_RBR (receive buffer register) is compared to, if not masked (*MSPn_RCM*.RCMx = 0). Comparison does not occur (always true) for the masked bit (*MSPn_RCM*.RCMx = 1).

MSPn_RCM MSP receive compare mask register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCM31	RCM30	RCM29	RCM28	RCM27	RCM26	RCM25	RCM24	RCM23	RCM22	RCM21	RCM20	RCM19	RCM18	RCM17	RCM16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCM15	RCM14	RCM13	RCM12	RCM11	RCM10	RCM9	RCM8	RCM7	RCM6	RCM5	RCM4	RCM3	RCM2	RCM1	RCM0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MSPnBaseAddress + 0x038

Reset: 0x0000 0000

Description: In multichannel mode (*MSPn_MCR*.RMCEN = 1), if receive multichannel comparison mode is enabled (MSPn_MCR.RMCMP = 1X), the received data for any enabled channel (that is, with its corresponding bit set in MSPn_RCEx) is compared bit-to-bit

with the data in the *MSPn_RCV*, for the bits that are not masked in *MSPn_RCM*. When the comparison result is false (MSPn_MCR.RMCMP = 10), or true (MSPn_MCR.RMCMP = 11), the received data is accepted, MSPn_RBR is copied to MSPn_RDR, the *MSPn_FLR*.RFU and RFE flags are updated, and the receive interrupt is set.

```
Comparison result (TRUE or FALSE):=
   [(<MSPn_RBR> AND NOT<MSPn_RCM>) == (<MSPn_RCV> AND
NOT<MSPn_RCM>]
```

Note: MSPn_RCM value must be set according to the defined element length (via RP1ELEN in MSPn_RCF): non significant bits of MSPn_RBR must be masked (corresponding bits set to 1 in MSPn_RCM) for a correct comparison result.

RCM[31:0] Receive compare mask bit x. Indicates bit x of MSPn_RBR is masked for comparison to *MSPn_RCV*. Only used when receive comparison mode is enabled (*MSPn_MCR*.RMCMP = 1X).

0: not masked 1: masked

MSPn_TCE0 MSP transmit channel enable register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCE 31	TCE 30	TCE 29	TCE 28	TCE 27	TCE 26	TCE 25	TCE 24	TCE 23	TCE 22	TCE 21	TCE 20	TCE 19	TCE 18	TCE 17	TCE 16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCE 15	TCE 14	TCE 13	TCE 12	TCE 11	TCE 10	TCE 9	TCE 8	TCE 7	TCE 6	TCE 5	TCE 4	TCE 3	TCE 2	TCE 1	TCE 0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MSPnBaseAddress + 0x040

Reset: 0x0000 0000

Description: If multichannel mode is enabled for the transmitter, 0 to 128 elements can be enabled for transmission, via the transmit channel enable registers *MSPn_TCEx*

TCE[127:0] Transmit channel enable. Enables transmission of elements in the current frame. Only used when transmit multichannel mode is enabled.

0: disabled (MSPTXDx signal goes HiZ) 1: enabled (MSPTXDx signal drives the data)

MSPn_TCE1 MSP transmit channel enable register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCE 63	TCE 62	TCE 61	TCE 60	TCE 59	TCE 58	TCE 57	TCE 56	TCE 55	TCE 54	TCE 53	TCE 52	TCE 51	TCE 50	TCE 49	TCE 48
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCE 47	TCE 46	TCE 45	TCE 44	TCE 43	TCE 42	TCE 41	TCE 40	TCE 39	TCE 38	TCE 37	TCE 36	TCE 35	TCE 34	TCE 33	TCE 32
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MSPnBaseAddress + 0x044

Reset: 0x0000 0000

Description: See description of *MSPn_TCE0 on page 311*.

MSPn_TCE2 MSP transmit channel enable register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCE 95	TCE 94	TCE 93	TCE 92	TCE 91	TCE 90	TCE 89	TCE 88	TCE 87	TCE 86	TCE 85	TCE 84	TCE 83	TCE 82	TCE 81	TCE 80
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCE 79	TCE 78	TCE 77	TCE 76	TCE 75	TCE 74	TCE 73	TCE 72	TCE 71	TCE 70	TCE 69	TCE 68	TCE 67	TCE 66	TCE 65	TCE 64
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MSPnBaseAddress + 0x048

Reset: 0x0000 0000

Description: See MSPn_TCE0 description.

MSPn_TCE3 MSP transmit channel enable register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TCE 127	TCE 126	TCE 125	TCE 124	TCE 123	TCE 122	TCE 121	TCE 120	TCE 119	TCE 118	TCE 117	TCE 116	TCE 115	TCE 114	TCE 113	TCE 112
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCE 111	TCE 110	TCE 109	TCE 108	TCE 107	TCE 106	TCE 105	TCE 104	TCE 103	TCE 102	TCE 101	TCE 100	TCE 99	TCE 98	TCE 97	TCE 96
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MSPnBaseAddress + 0x04C

Reset: 0x0000 0000

Description: See MSPn_TCE0 description.

MSPn_RCE0 MSP receive channel enable register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCE 31	RCE 30	RCE 29	RCE 28	RCE 27	RCE 26	RCE 25	RCE 24	RCE 23	RCE 22	RCE 21	RCE 20	RCE 19	RCE 18	RCE 17	RCE 16
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCE 15	RCE 14	RCE 13	RCE 12	RCE 11	RCE 10	RCE 9	RCE 8	RCE 7	RCE 6	RCE 5	RCE 4	RCE 3	RCE 2	RCE 1	RCE 0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MSPnBaseAddress + 0x060

Reset: 0x0000 0000

Description: If multichannel mode is enabled for the receiver, 0 to 128 elements can be enabled for reception, via the receive channel enable registers MSPn_RCEx.

RCE[127:0] Receive channel enable. Enables reception of elements in the current frame. Only used when receive multichannel mode is enabled.

0: reception disabled 1: enabled

MSPn_RCE1 MSP receive channel enable register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCE 63	RCE 62	RCE 61	RCE 60	RCE 59	RCE 58	RCE 57	RCE 56	RCE 55	RCE 54	RCE 53	RCE 52	RCE 51	RCE 50	RCE 49	RCE 48
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCE 47	RCE 46	RCE 45	RCE 44	RCE 43	RCE 42	RCE 41	RCE 40	RCE 39	RCE 38	RCE 37	RCE 36	RCE 35	RCE 34	RCE 33	RCE 32
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MSPnBaseAddress + 0x064

Reset: 0x0000 0000

Description: See description of *MSPn_RCE0*.

MSPn_RCE2 MSP receive channel enable register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCE 95	RCE 94	RCE 93	RCE 92	RCE 91	RCE 90	RCE 89	RCE 88	RCE 87	RCE 86	RCE 85	RCE 84	RCE 83	RCE 82	RCE 81	RCE 80
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCE 79	RCE 78	RCE 77	RCE 76	RCE 75	RCE 74	RCE 73	RCE 72	RCE 71	RCE 70	RCE 69	RCE 68	RCE 67	RCE 66	RCE 65	RCE 64
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MSPnBaseAddress + 0x068

Reset: 0x0000 0000

Description: See description of *MSPn_RCE0*.

MSPn_RCE3 MSP receive channel enable register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RCE 127	RCE 126	RCE 125	RCE 124	RCE 123	RCE 122	RCE 121	RCE 120	RCE 119	RCE 118	RCE 117	RCE 116	RCE 115	RCE 114	RCE 113	RCE 112
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCE 111	RCE 110	RCE 109	RCE 108	RCE 107	RCE 106	RCE 105	RCE 104	RCE 103	RCE 102	RCE 101	RCE 100	RCE 99	RCE 98	RCE 97	RCE 96
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Address: MSPnBaseAddress + 0x06C

Reset: 0x0000 0000

Description: See description of *MSPn_RCE0*.

MSPnPeriphID0 MSP peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: MSPnBaseAddress + 0xFE0

Reset: 0x0000 0021

Description: PartNumber0 returns 0x21.

MSPnPeriphID1 MSP peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: MSPnBaseAddress + 0xFE4

Reset: 0x0000 0000

Description: Designer0 returns 0x0, PartNumber1 returns 0x0.

MSPnPeriphID2 MSP peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: MSPnBaseAddress + 0xFE8

Reset: 0x0000 0028

Description: Revision returns 0x2, Designer1 returns 0x8.

MSPnPeriphID3 MSP peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: MSPnBaseAddress + 0xFEC

Reset: 0x0000 0000

Description: Configuration returns 0x00.

MSPnPCellID0 MSP PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MSPPCellID0							
R	R	R	R	R	R	R	R	R							

Address: MSPnBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: MSPPCellID0 returns 0x0D.

MSPnPCellID1 MSP PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MSPPCellID1							
R	R	R	R	R	R	R	R	R							

Address: MSPnBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: MSPPCellID1 returns 0xF0.

MSPnPCellID2 MSP PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MSPPCellID2							
R	R	R	R	R	R	R	R	R							

Address: MSPnBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: MSPPCellID2 returns 0x05.

MSPnPCellID3 MSP PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MSPPCellID3							
R	R	R	R	R	R	R	R	R							

Address: MSPnBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: MSPPCellID3 returns 0xB1.

24 Fast IrDA controller (FIrDA)

24.1 FIrDA register addressing

Register addresses are provided as the FIrDA base address, IrDABaseAddress, plus the register offset.

The IrDA base address is 0x101F A000.

24.2 FIrDA register summary

The communication between the FIrDA controller and the APB bus is performed by means of 32-bit registers.

Table 49. IrDA register list

Offset	Register name	Description	Page
0x000	IrDA_CR	IrDA control register	318
0x004	IrDA_CFR	IrDA configuration register	318
0x008	IrDA_PAR	IrDA parameter register	319
0x00C	IrDA_DVR	IrDA divider register	320
0x010	IrDA_SR	IrDA status register	321
0x014	IrDA_TFS	IrDA transmission frame size register	321
0x018	IrDA_RFS	IrDA reception frame size register	322
0x01C	IrDA_TXB	IrDA transmission buffer register	322
0x020	IrDA_RXB	IrDA reception buffer register	323
0x024	IrDA_IMSC	IrDA interrupt mask set/clear register	323
0x028	IrDA_RIS	IrDA raw interrupt status register	324
0x02C	IrDA_MIS	IrDA masked interrupt status register	324
0x030	IrDA_ICR	IrDA interrupt clear register	325
0x034	IrDA_ISR	IrDA interrupt set register	326
0x038	IrDA_DMACR	IrDA DMA control register	326
0xFE0	IrDA_PeriphID0	IrDA peripheral identification register 0 (bits 7:0)	327
0xFE4	IrDA_PeriphID1	IrDA peripheral identification register 1 (bits 15:8)	327
0xFE8	IrDA_PeriphID2	IrDA peripheral identification register 2 (bits 23:16)	328
0xFEC	IrDA_PeriphID3	IrDA peripheral identification register 3 (bits 31:24)	328
0xFF0	IrDA_PCellID0	IrDA PCell identification register 0 (bits 7:0)	328
0xFF4	IrDA_PCellID1	IrDA PCell identification register 1 (bits 15:8)	329
0xFF8	IrDA_PCellID2	IrDA PCell identification register 2 (bits 23:16)	329
0xFFC	IrDA_PCellID3	IrDA PCell identification register 3 (bits 31:24)	329

24.3 FIrDA register descriptions

IrDA_CR IrDA control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RUN
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	RW

Address: IrDABaseAddress + 0x000

Reset: 0x0000 0000

Description: The IrDA controller is controlled by way of this serial port control register.

RUN Enable FIrDA controller.
0: inactive 1: listening state

0 Reserved for future use. Reading returns 0. Must be written with 0.

IrDA_CFR IrDA configuration register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	SIPEN	TXPOL	RXPOL	BS		
										RW	RW	RW	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	RATV												
			RW												

Address: IrDABaseAddress + 0x004

Reset: 0x0002 0EA6

Description: Configures the IrDA controller. It must only be modified when the FIrDA controller is disabled via *IrDA_CR*.RUN.

SIPEN Automatic SIP (serial infrared interaction pulse) generation enable.
0: no SIP is generated (default)
1: SIP generated automatically after each frame transmission

TXPOL Polarity of Tx pulses.
0: active low (default) 1: active high

RXPOL Polarity of Rx pulses.
0: active low (default) 1: active high

BS Burst size. Burst size of DMA transfers (the DMA controller does not support a burst size of 2 words).
000: 1 word 001: 2 words
010: 4 words (default) 011 to 111: reserved

RATV Reception abort timer value. A frame with a single pair of characters, with a time gap greater than 10 ms, is considered as an invalid frame. The reception abort timer of the synchronization must be programmed with RATV according to the following equation:

RATV = (T_{abort} * $f_{IRDACLK}$) / 128, (where T_{abort} is in seconds and $f_{IRDACLK}$ in Hz)

RATV = 0x0EA6 when T_{abort}=10ms at $f_{IRDACLK}$ = 48 MHz (default)

0 Reserved for future use. Reading returns 0. Must be written with 0.

IrDA_PAR IrDA parameter register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	MNRB											
R	R	R	R	RW											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	ABF						MODE	
R	R	R	R	R	R	R	R	RW						RW	

Address: IrDABaseAddress + 0x008

Reset: 0x0046 0000

Description: Specifies transmission parameters. It must only be modified when the FIrDA controller is disabled via *IrDA_CR*.RUN.

MNRB Maximum number of received bytes. Must be programmed according to the negotiated data size (see IrLAP specification 6.6.5). In FIR mode the effective maximum number of received bytes is data size + 2 + 4 (information + address and control + 4 CRC bytes).

0x046: maximum of 70 bytes (default) N: maximum of N bytes

ABF Number of additional beginning flags. Must be programmed according to negotiated XBOFs (see IrLAP specification 6.6.7).

00 0000: no additional beginning flag

00 0001: 1 additional beginning flag until 11 0000: 48 additional beginning flags

11 0001 to 11 1111: reserved

MODE Infrared mode.

00: SIR mode (default) 01: MIR mode

10: FIR mode 11: reserved

0 Reserved for future use. Reading returns 0. Must be written with 0.

IrDA_DVR IrDA divider register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0						DEC					
R	R	R	R	R						RW					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			INC								N				
			RW								RW				

Address: IrDABaseAddress + 0x00C

Reset: 0x0000 0000

Description: The enable signal EN_PULSE is derived from the IRDACLK by means of a fractional divider. The frequency of EN_PULSE depends on the increment value (INC) and the decrement value (DEC) of this register. The enable signal EN_SYMB is derived from EN_PULSE by means of an integer divider. The denominator N + 1 depends on the 8-bit field N. This register must only be modified when the IrDA controller is disabled by *IrDA_CR*. RUN.

DEC Decrement. Decrement value of fractional divider.
DEC = L - K. The values of K and L are listed in *Table 50*, *Table 51* and *Table 52*.

INC Increment. Increment value of fractional divider. INC = K. K must always be less than L.
K = L is not allowed, except for K = L = 0. In this case, clk_pulse is equal to IRDACLK.

N Denominator. N + 1 is the denominator of integer divider.
N = 0 to 255. N is set according to *Table 50*, *Table 51* and *Table 52*.

O Reserved for future use. Reading returns 0. Must be written with 0.

Table 50. Setting of K, L and N + 1 for SIR ($f_{IRDACLK}$ = 48 MHz)

Bit rate (Kbit/s)	f_{EN_PULSE} (kHz)	f_{EN_SYMB} (kHz)	K	L	N + 1
9.6	576	9.6	3	250	60
19.2	576	19.2	3	250	30
38.4	576	38.4	3	250	15
57.6	576	57.6	3	250	10
115.2	576	115.2	3	250	5

Table 51. Setting of K, L and N + 1 for MIR ($f_{IRDACLK}$ = 48 MHz)

Bit rate (Kbit/s)	f_{EN_PULSE} (kHz)	f_{EN_SYMB} (kHz)	K	L	N + 1
576	2304	576	6	125	4
1152	4608	1152	12	125	4

Table 52. Setting of K, L and N + 1 for FIR ($f_{IRDACLK}$ = 48 MHz)

Bit rate (Kbit/s)	f_{EN_PULSE} (kHz)	f_{EN_SYMB} (kHz)	K	L	N + 1
4000	8000	2000	1	6	4

IrDA_SR IrDA status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	TXS	RXS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	RH	RH

Address: IrDABaseAddress + 0x010

Reset: 0x0000 0000

Description: Indicates the status of the FIrDA controller.

TXS Transmission state. 1: FIrDA controller is in transmission state

RXS Reception state. 1: FIrDA controller is in reception state

0 Reserved for future use. Reading returns 0. Must be written with 0.

IrDA_TFS IrDA transmission frame size register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	TFS											
R	R	R	R	W											

Address: IrDABaseAddress + 0x014

Reset: 0x0000 0000

Description: Indicates the size of the frame to be transmitted.

TFS Transmission frame size.
Number of bytes transmitted is data size + 2 (information + address and control bytes).
0x000: reset value
0x001: transmit frame with 2 data bytes
0x002: transmit frame with 3 data bytes, up to 0x801: transmit frame with 2050 data bytes
0x802 to 0xFFF: reserved

0 Reserved for future use. Reading returns 0. Must be written with 0.

IrDA_RFS IrDA reception frame size register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	RFS											
R	R	R	R	RH											

Address: IrDABaseAddress + 0x018

Reset: 0x0000 0000

Description: Indicates the size of the received frame.

RFS Reception frame size.
In SIR and MIR, the number of received bytes is:
– data size + 2 + 2 (information + address and control + 2 CRC bytes).
In FIR the number of received bytes is:
– data size + 2 + 4 (information + address and control + 4 CRC bytes).
0x000: reset value
0x001 to 0x003: reserved
0x004: 4 data bytes to 0x806: 2054 data bytes
0x807 to 0xFFF: reserved

0 Reserved for future use. Reading returns 0. Must be written with 0.

IrDA_TXB IrDA transmission buffer register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXD															
W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXD															
W															

Address: IrDABaseAddress + 0x01C

Reset: 0x0000 0000

Description: Contains the data to be transmitted in transmission mode. There must be a pause of one clock cycle between two write accesses.

TXD Transmission data. Bytes to be transmitted.

IrDA_RXB IrDA reception buffer register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXD RH															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXD RH															

Address: IrDABaseAddress + 0x020

Reset: 0x0000 0000

Description: Contains the received data bytes in reception mode.

RXD Reception data. Received bytes.

IrDA_IMSC IrDA interrupt mask set/clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	FD IM	FI IM	SD IM	FT IM	BREQ IM	LBREQ IM	SREQ IM	LSREQ IM
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: IrDABaseAddress + 0x024

Reset: 0x0000 0000

Description: Interrupt mask set or clear register. On a read, it gives the current value of the mask on the relevant interrupt. A write of 1 to the particular bit sets the mask, enabling the interrupt to be read. A write of 0 clears the corresponding mask.

FDIM Frame detected interrupt mask.
0: interrupt masked 1: interrupt not masked

FIIM Frame invalid interrupt mask.
0: interrupt masked 1: interrupt not masked

SDIM Signal detected interrupt mask.
0: interrupt masked 1: interrupt not masked

FTIM Frame detected interrupt mask.
0: interrupt masked 1 interrupt not masked

BREQIM BREQ burst request interrupt mask.
0: interrupt masked 1: interrupt not masked

LBREQIM LBREQ last burst request interrupt mask.
0: interrupt masked 1: interrupt not masked

SREQIM SREQ single word request interrupt mask.
0: interrupt masked 1: interrupt not masked

LSREQIM LSREQ last single word request interrupt mask.
0: interrupt masked 1: interrupt not masked

0 Reserved for future use. Reading returns 0. Must be written with 0.

IrDA_RIS IrDA raw interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	FD RIS	FI RIS	SD RIS	FT RIS	BREQ RIS	LBREQ RIS	SREQ RIS	LSREQ RIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: IrDABaseAddress + 0x028

Reset: 0x0000 0008

Description: Indicates the current raw status of interrupts prior to masking.

FDRIS Frame detected raw interrupt status (prior to masking).

FIRIS Frame invalid raw interrupt status (prior to masking).

SDRIS Signal detected raw interrupt status (prior to masking).

FTRIS Frame transmitted raw interrupt status (prior to masking).

BREQRIS Receive frame sync raw interrupt status (prior to masking).

LBREQRIS LBREQ raw interrupt status (prior to masking).

SREQRIS SREQ raw interrupt status (prior to masking).

LSREQRIS LSREQ raw interrupt status (prior to masking).

0 Reserved for future use. Reading returns 0. Must be written with 0.

IrDA_MIS IrDA masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	FD MIS	FI MIS	SD MIS	FT MIS	BREQ MIS	LBREQ MIS	SREQ MIS	LSREQ MIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: IrDABaseAddress + 0x02C

Reset: 0x0000 0000

Description: Contains the current, masked status of the corresponding interrupt.

FDMIS Frame detected masked interrupt status.

FIMIS Frame invalid masked interrupt status.

SDMIS Signal detected masked interrupt status.

FTMIS Frame transmitted masked interrupt status.

BREQMIS Receive frame sync masked interrupt status.

LBREQMIS LBREQ masked interrupt status.

SREQMIS SREQ masked interrupt status.

LSREQMIS LSREQ masked interrupt status.

0 Reserved for future use. Read returns 0. Must be written with 0.

IrDA_ICR IrDA interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	FDIC	FIIC	SDIC	FTIC	BREQIC	LBREQIC	SREQIC	LSREQIC
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: IrDABaseAddress + 0x030

Reset: 0x0000 0000

Description: Clears interrupts.

FDIC Writing 1 clears the frame detected interrupt. A read returns 0.

FIIC Writing 1 clears the frame invalid interrupt. A read returns 0.

SDIC Writing 1 clears the signal detected interrupt. A read returns 0.

FTIC Writing 1 clears the frame transmitted interrupt. A read returns 0.

BREQIC Writing 1 clears the frame sync interrupt. A read returns 0.

LBREQIC Writing 1 clears the LBREQ interrupt. A read returns 0.

SREQIC Writing 1 clears the SREQ interrupt. A read returns 0.

LSREQIC Writing 1 clears the LSREQ interrupt. A read returns 0.

0 Reserved for future use. Reading returns 0. Must be written with 0.

IrDA_ISR — IrDA interrupt set register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	FDIS	FIIS	SDIS	FTIS	BREQIS	LBREQIS	SREQIS	LSREQIS
R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Address: IrDABaseAddress + 0x034

Reset: 0x0000 0000

Description: Sets interrupts. A write of 0 has no effect.

FDIS Frame detect interrupt set.
Writing 1 sets the frame detected interrupt. A read returns 0.

FIIS Frame invalid interrupt set.
Writing 1 sets the frame invalid interrupt. A read returns 0.

SDIS Signal detected interrupt set.
Writing 1 sets the signal detected interrupt. A read returns 0.

FTIS Frame transmitted interrupt set.
Writing 1 sets the frame transmitted interrupt. A read returns 0.

BREQIS Burst request interrupt set.
Writing 1 sets the BREQ interrupt. A read returns 0.

LBREQIS Last burst request interrupt set.
Writing 1 sets the LBREQ interrupt. A read returns 0.

SREQIS Single request interrupt set.
Writing 1 sets the SREQ interrupt. A read returns 0.

LSREQIS Last single request interrupt set.
Writing 1 sets the LSREQ interrupt. A read returns 0.

0 Reserved for future use. Reading returns 0. Must be written with 0.

IrDA_DMACR — IrDA DMA control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	BREQEN	LBREQEN	SREQEN	LSREQEN
R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Address: IrDABaseAddress + 0x038

Reset: 0x0000 0000

Description: Controls DMA requests.

BREQEN Burst request DMA enable.
0: clears pending DMA request and disables further requests
1: enabled

LBREQEN Last burst request DMA enable.
0: clears pending DMA request and disables further requests
1: enabled

SREQEN Single request DMA enable.
0: clears pending DMA request and disables further requests
1: enabled

LSREQEN Last single request DMA enable.
0: clears pending DMA request and disables further requests
1: enabled

0 Reserved for future use. Reading returns 0. Must be written with 0.

IrDA_PeriphID0 IrDA peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: IrDABaseAddress + 0xFE0

Reset: 0x0000 0010

Description: Part Number0 reads back as 0x10.

IrDA_PeriphID1 IrDA peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: IrDABaseAddress + 0xFE4

Reset: 0x0000 0000

Description: Designer0 and Part Number1 both read back as 0x0.

IrDA_PeriphID2 IrDA peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: IrDABaseAddress + 0xFE8

Reset: 0x0000 0008

Description: Revision returns the peripheral revision (0x0) and Designer1 reads back as 0x8.

IrDA_PeriphID3 IrDA peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: IrDABaseAddress + 0xFEC

Reset: 0x0000 0000

Description: Configuration reads back as 0x00.

IrDA_PCellID0 IrDA PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	IrDAPCellID0							
R	R	R	R	R	R	R	R	R							

Address: IrDABaseAddress + 0xFF0

Reset: 0x0000 000D

Description: IrDAPCellID0 reads back as 0x0D.

IrDA_PCellID1 IrDA PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	IrDAPCellID1							
R	R	R	R	R	R	R	R	R							

Address: IrDABaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: IrDAPCellID1 reads back as 0xF0.

IrDA_PCellID2 IrDA PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	IrDAPCellID2							
R	R	R	R	R	R	R	R	R							

Address: IrDABaseAddress + 0xFF8

Reset: 0x0000 0005

Description: IrDAPCellID2 reads back as 0x05.

IrDA_PCellID3 IrDA PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	IrDAPCellID3							
R	R	R	R	R	R	R	R	R							

Address: IrDABaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: IrDAPCellID3 reads back as 0xB1.

25 I2C high-speed controllers (I2C0,1)

25.1 I2C register addressing

Register addresses are provided as the I2C base address, I2CnBaseAddress, plus the register offset.

There are 2 I2C base addresses.

Table 53. I2C base addresses

I2C base address	I2C
0x101F 8000	0
0x101F 7000	1

I2C0 and I2C1 are referred to as I2Cn throughout this document.

25.2 I2C register summary

Table 54. I2C register list

Offset	Register	Description	Page
0x000	I2Cn_CR	I2C control register	*331*
0x004	I2Cn_SCR	I2C slave control register	*333*
0x008	I2Cn_HSMCR	I2C HS master code register	*334*
0x00C	I2Cn_MCR	I2C master control register	*334*
0x010	I2Cn_TFR	I2C transmit FIFO register	*335*
0x014	I2Cn_SR	I2C status register	*336*
0x018	I2Cn_RFR	I2C receive FIFO register	*338*
0x01C	I2Cn_TFTR	I2C transmit FIFO threshold register	*338*
0x020	I2Cn_RFTR	I2C receive FIFO threshold register	*339*
0x024	I2Cn_DMAR	I2C DMA register	*339*
0x028	I2Cn_BRCR	I2C baud rate counter register	*340*
0x02C	I2Cn_IMSCR	I2C interrupt mask set and clear register	*341*
0x030	I2Cn_RISR	I2C raw interrupt status register	*342*
0x034	I2Cn_MISR	I2C masked interrupt status register	*344*
0x038	I2Cn_ICR	I2C interrupt set and clear register	*344*
0xFE0	I2Cn_PeriphID0	I2C peripheral identification register 0 (bits 7:0)	*327*
0xFE4	I2Cn_PeriphID1	I2C peripheral identification register 1 (bits 15:8)	*327*
0xFE8	I2Cn_PeriphID2	I2C peripheral identification register 2 (bits 23:16)	*328*
0xFEC	I2Cn_PeriphID3	I2C peripheral identification register 3 (bits 31:24)	*328*
0xFF0	I2Cn_PCellID0	I2C PCell identification register 0 (bits 7:0)	*328*

Table 54. I2C register list (continued)

Offset	Register	Description	Page
0xFF4	I2Cn_PCellID1	I2C PCell identification register 1 (bits 15:8)	329
0xFF8	I2Cn_PCellID2	I2C PCell identification register 2 (bits 23:16)	329
0xFFC	I2Cn_PCellID3	I2C PCell identification register 3 (bits 31:24)	329

Note: All the I2C configuration registers (I2Cn_CR [excluding PE, DMA_TX_EN, DMA_RX_EN, FTX and FRX bits]) can be modified only when the device is disabled (I2Cn_CR.PE is reset) to avoid synchronization problems with the different clock domains (system, I2C clocks).

25.3 I2C register descriptions

I2Cn_CR I2C control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	FON		LM	DMA_SLE	DMA_RX_EN	DMA_TX_EN	FRX	FTX	SGCM	SM		SAM	OM		PE
R	RW		RW	RW	RW	RW	RW	RW	RW	RW		RW	RW		RW

Address: I2CnBaseAddress + 0x0000

Reset: 0x0000 0000

Description: Control register.

FON Filtering on. Sets the digital filters on the SDA and SCL lines, according to the I2C-bus requirements, when standard open-drain pads are used.
00: no digital filters inserted
01: digital filters (1 clock-wide-spikes) inserted
10: digital filters (2 clock-wide-spikes) inserted
11: digital filters (4 clock-wide-spikes) inserted

LM Loopback mode. Sets the loopback operating mode for the I2C controller.
0: normal mode
1: loopback,Tx FIFO internally directed to Rx FIFO

DMA_SLE DMA synchronization logic enable. Select/bypass synchronization logic (double stage register) on signals DMA_TC_TX, DMA_CLR_TX, DMA_TC_RX, DMA_CLR_RX generated by the DMA controller. Must be programmed according to DMACSynch (PL080) register.
0: disabled on DMA input interface
1: enabled on DMA input interface

DMA_RX_EN DMA Rx enable. Enables the DMA Rx interface to generate burst/single requests for a single DMA descriptor until a master-read (MR) or write-to-slave (WTS) operation is executed. The DMA Rx channel must be programmed for a peripheral-to-memory transfer where the flow controller is DMA.

When an MR operation is scheduled (by writing to the *I2Cn_MCR* register), the DMA Rx interface (if enabled) generates a sequence of burst/single requests, with a known transfer length of I2Cn_MCR.LENGTH.

When a WTS operation is received (interrupt bit *I2Cn_RISR*.WTSR asserted), the application must enable the DMA Rx interface to generate a sequence of burst/single requests. Since the DMA transfer length is unknown, it must be programmed to the maximum value.

A burst request (for master and slave operations) is automatically triggered when the Rx FIFO contains a number of words greater than or equal to the programmed source burst size (*I2Cn_DMAR*.SBSIZE_RX).

Once the DMA transfer is complete, the DMA Rx interface is automatically turned off, clearing this bit, and the terminal count interrupt bit in register *DMACn_TCRIS* is asserted. The application must always verify the status of the executed I2C transaction (success or abort, and transfer length). See also *I2Cn_DMAR on page 339* for more explanation. The CPU cannot read the Rx FIFO register (*I2Cn_RFR*) when the DMA Rx interface is enabled.

0: idle state, DMA Rx interface disabled 1: run state, DMA Rx interface enabled

DMA_TX_EN DMA Tx enable. Enables the DMA Tx interface to generate burst/single requests for a single DMA descriptor until a master write (MW) or read-from-slave (RFS) operation is executed. The DMA Tx channel must be programmed for a memory-to-peripheral transfer where the flow controller is a DMA.

When a master write operation is scheduled (writing to the *I2Cn_MCR* register), the DMA Tx interface (if enabled) generates a sequence of burst/single requests, and transfer length is I2Cn_MCR.LENGTH.

When an read-from-slave operation is received (interrupt bit *I2Cn_RISR*.RFSR assertion), the application enables the DMA Tx interface to generate a sequence of burst or single requests (depending on the *I2Cn_DMAR*.BURST_TX setting). Since the DMA transfer length is unknown, it must be programmed to the maximum value.

When a NACK bit is received from the I2C line, the Tx FIFO may contain some pending data because the transfer length is unknown. The pending data in the Tx FIFO is automatically discarded, flushing the FIFO, at the beginning of the next read-from-slave operation.

The burst request (for master and slave operations) is automatically triggered when the Tx FIFO contains a number of available entries greater than or equal to the programmed destination burst size (I2Cn_DMAR.DBSIZE_TX).

Once the DMA transfer is completed, the DMA Tx interface is automatically turned off, clearing this bit, and the terminal count interrupt bit in the *DMACn_TCRIS* register is asserted. The application always verifies the status of the executed I2C transaction (success, abortion and transfer length). See *I2Cn_DMAR on page 339* for more explanation. The CPU cannot write to the Tx FIFO register (*I2Cn_TFR*) when the DMA Tx interface is enabled.

0: idle state, DMA Tx interface disabled 1: run state, DMA Tx interface enabled

FRX Flush receive. Shows the flush status of the receive circuitry (FIFO and FSM). The I2C node configuration (register setting) is not affected by flushing. Flushing is performed on modules working on different clock domains (system and I2C clocks) and takes several system clock cycles to complete. On completion, the I2C node (internal logic) clears this bit. The application must not access the Rx FIFO during flushing and should poll on this bit while waiting for flushing to end.

0: completed (I2C controller clears the bit) 1: started and in progress (set by application)

FTX Flush transmit. Shows the transmit circuitry flush status (FIFO, FSM). The I2C node configuration (register setting) is not affected by flushing. Flushing is performed on modules working on different clock domains (system and I2C clocks) and takes several system clock cycles to complete. On completion, the I2C node (internal logic) clears this bit. The application must not access the Tx FIFO during flushing and should poll on this bit while waiting for flushing to end.

0: completed (I2C controller clears the bit) 1: started and in progress (set by application)

SGCM Slave general call mode. Defines the slave controller operating mode when a general call is received (*I2Cn_SR*.TYPE=01). A hardware general call is *always* managed in transparent mode.

0: transparent mode. The slave receiver recognizes the general call but any action is taken by software after decoding the Rx FIFO message.

1: direct mode. The slave receiver recognizes the general call and executes the related actions directly (without software intervention). Only the status is stored in I2Cn_SR for application notification.

SM Speed mode. Defines the speed mode related to the serial bit rate.

00: standard mode (up to 100 Kb/s) 01: fast mode (up to 400 Kb/s)

10: high-speed (up to 3.4 Mb/s) 11: reserved

SAM Slave addressing mode. Defines the slave addressing mode when the peripheral works in slave or master/slave mode. The received address is compared with the content of the register *I2Cn_SCR*.

0: 7-bit addressing mode 1: 10-bit addressing mode

OM Operating mode.

00: slave mode (peripheral can only respond (Tx/Rx) when addressed by a master device).

01: master mode (peripheral works in a multi-master system where it cannot be addressed by another master device, it can only initiate a new transfer as master device).

10: master/slave mode (the peripheral works in a multi-master system where it can be addressed by another master device and can initiate a transfer as master device).

11: reserved.

PE Peripheral enable. Enables the peripheral to work in the operating mode set by *I2Cn_CR*.OM.

0: disabled 1: enabled

0 Reserved for future use. Reading returns 0. Must be written with 0.

I2Cn_SCR I2C slave control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLSU															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	ESA10						SA7			
R	R	R	R	R	R	RW						RW			

Address: I2CnBaseAddress + 0x0004

Reset: 0x000F 0055

Description:

SLSU Slave data setup time. Defines data setup time after SCL clock stretching in terms of I2CnCLK cycles. Data setup time = SLSU-1 clock cycles. This value depends on the mode selected (standard, fast or high-speed) and on the I2CnCLK frequency. The needed setup time for the three modes is 250 ns, 100 ns and 10 ns respectively, thus leading to typical values of 14, 6 and 2 for a 48 MHz I2CnCLK.

ESA10 Extended 10-bit slave address. Includes the extension (MSBs) to the SA7 register field for when the slave addressing mode is 10-bit (*I2Cn_CR*.SAM = 1).

SA7 7-bit slave address. Includes the 7-bit slave address or the LSBs of the 10-bit slave address. The slave address mode is set according to the *I2Cn_CR*.SAM setting.

0 Reserved for future use. Reading returns 0. Must be written with 0.

I2Cn_HSMCR I2C high-speed master code register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0		MC	
R	R	R	R	R	R	R	R	R	R	R	R	R		RW	

Address: I2CnBaseAddress + 0x0008

Reset: 0x0000 0000

Description: Contains the master code used by the master during configuration in high-speed mode.

MC Master code. Defines the last three bits of the master code. This bit is only relevant when the controller is configured in high-speed mode (*I2Cn_CR*.SM = 10).

0 Reserved for future use. Reading returns 0. Must be written with 0.

I2Cn_MCR I2C master control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15
0	0	0	0	0	0						LENGTH					
R	R	R	R	R	R						RW					

-	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	-
	P	AM		SB		EA10					A7				OP	
	RW	RW		RW		RW					RW				RW	

Address: I2CnBaseAddress + 0x000C

Reset: 0x0000 0000

Description: Defines the transfer features. A typical transfer is defined as:

- start condition,
- start byte procedure (optional),

– address (7- or 10-bit) and read/write bit,
– data transmission/reception.

The master read or write operations (MR/MW) are performed sequentially one at a time (no queuing mode). Writing to this register triggers the related operation, and the Tx FIFO is preloaded otherwise the I2C controller cannot start the operation. Each operation is initiated by the start command and can terminate by the stop command (I2C line). When the operation is not terminated by the stop command, a repeated start follows on the next required operation, otherwise the I2C line stalls.

When the MR/MW operation is complete, the *I2Cn_RISR*.MTD is asserted and the related status of the operation is stored in the *I2Cn_SR* register. If a write operation fails (transaction aborted) the application flushes the Tx FIFO, asserting *I2Cn_CR*.FTX.

For 10-bit addressing, an MR operation must be preceded by an MW to the same slave, due to the partial slave addressing used in 10-bit read operations.

LENGTH Transaction length. Defines the length, in number of bytes, to be transmitted or received. For a write operation, the payload is stored in the Tx FIFO. A transaction can be larger than the size of the Tx FIFO. For a read operation, the length refers to the number of bytes to be received before generating a NACK response. A transaction can be larger than the Rx FIFO size. The I2C clock line is stretched low until the data in the Rx FIFO is processed.

P Stop condition. Indicates if a stop condition terminates the current transaction. If it is not terminated by a stop condition, a repeated start condition is generated to avoid stalling the I2C line.
0: not terminated 1: terminated

AM Address type. Indicates what initiates the transaction. If 00, then OP, A7 and EA10 become irrelevant.
00: general call command 01: 7-bit address from the A7 field
10: 10-bit address from EA10 and A7 11: reserved

SB Start byte. Indicates if the start byte procedure is prefixed to the current transaction.
0: not applied 1: prefixed

EA10 Extended address. MSB bits of A7. Valid only when the addressing mode is 10 bits (AM = 10).

A7 Address. 7-bit address or the LSBs of the10-bit address used to initiate the current transaction.

OP Operation. Indicates if the operation is a master write or master read.
0: master write 1: master read

0 Reserved for future use. Reading returns 0. Must be written with 0.

I2Cn_TFR I2C transmit FIFO register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	TDATA							
R	R	R	R	R	R	R	R	W							

Address: I2CnBaseAddress + 0x0010

Reset: 0x0000 0000

Description:

TDATA Transmission data. Contains the payload related to a master-write or read-from-slave operation to be written in the Tx FIFO. TDATA(0) is the first LSB transmitted over the I2C line. For a master-write operation, the Tx FIFO is preloaded, otherwise the I2C controller cannot start the operation until data is available.

For a read-from-slave operation, when the slave is addressed, *I2Cn_RISR*.RFSR is asserted and the CPU downloads the data in the FIFO. If the FIFO is empty and the I2C master still requires data, a new request (I2Cn_RISR.RFSE) is asserted to request the additional data from the CPU. The slave controller stretches the I2C clock line when no data is available for transmission. Since the Tx FIFO could contain some pending data related to the previous transfer (the transfer length may be unknown to the slave controller), the Tx FIFO is self-flushed beforehand to assert I2Cn_RISR.RFSR. Once the read-from-slave operation is completed, I2Cn_RISR.STD is asserted and the related status of the operation is stored in the *I2Cn_SR* register.

In CPU mode, FIFO management is based on the assertion of I2Cn_RISR.TXFNE near-empty threshold.

In DMA mode, single/burst requests are automatically executed based on the number of entries available in the Tx FIFO and the related destination burst size programmed in *I2Cn_DMAR*.DBSIZE_TX. The DMA requests are terminated at the end of the I2C read operation (NACK received by the master) by a dummy last single/burst request (also refer to *I2Cn_DMAR on page 339*).

0 Reserved for future use. Reading returns 0. Must be written with 0.

I2Cn_SR I2C status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	LENGTH			
R	R	R	R	R	R	R	R	R	R	R	R	R			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LENGTH							TYPE		CAUSE			STATUS		OP	
R							R		R			R		R	

Address: I2CnBaseAddress + 0x0014

Reset: 0x0000 0000

Description: Contains the status of the transfer in terms of operation type, status, cause of abort, type and length of transaction (in bytes).

On completion of a master or slave operation, *I2Cn_RISR*.MTD or I2Cn_RISR.STD is asserted and the related status of the operation is stored in the current register.

LENGTH Transfer length.

MR and WTS operations: size of the subsequent payload, in bytes.

MW and RFS operations: number of bytes transferred by the master/slave device.

WTS operations: if the transfer length exceeds 2047 bytes, the operation is stopped by the slave returning a NACK and the OVFL flag is set (via the CAUSE field).

RFS operations: if transfer length exceeds 2047 bytes, operation continues but length is reset to 0.

TYPE Receive type. Valid only for WTS operations.

00: FRAME: slave received a normal frame.

01: GCALL: slave received a general call. If *I2Cn_CR*.SGCM=1, the general call is executed directly without software intervention and only the control code-word is reported in the FIFO (LENGTH =0).

10: HW_GCALL: slave received a hardware general call.

11: reserved.

CAUSE Abort cause. Valid only when the STATUS field contains the ABORT tag.

000: NACK_ADDR: master received NACK after address transmission. Valid for MW/MR operations.

001: NACK_DATA: master received NACK during data phase of MW operation. Valid for MW operations.

010: ACK_MCODE: master received ACK after master code transmission in HS-mode. Valid for MW/MR operations in HS-mode.

011: ARB_LOST: master lost arbitration during MW/MR operation. Valid for MW/MR operations.

100: BERR_START: slave restart.

101: BERR_STOP: slave reset.

110: OVFL: slave received a WTS operation related frame longer than 2047 bytes. The slave device returns a NACK to complete the data transfer. Valid for WTS operations.

Others: reserved.

STATUS Controller status. Valid for MW, MR, WTS and RFS operations.

00: NOP: no operation in progress

01: ON_GOING: an operation is ongoing

10: OK: operation completed successfully

11: ABORT: abort due to event detailed in CAUSE

OP Operation.

00: MW: master write

01: MR: master read

10: RFS: read from slave

11: WTS: write to slave

0 Reserved for future use. Reading returns 0. Must be written with 0.

I2Cn_RFR I2C receive FIFO register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	RDATA							
R	R	R	R	R	R	R	R	R							

Address: I2CnBaseAddress + 0x0018

Reset: 0x0000 0000

Description:

RDATA Receive data. Received payload from an MR or WTS operation, to be read from the Rx FIFO. RDATA(0) is the first LSB bit received over the I2C line. If the FIFO is full, the I2C controller stretches the I2C clock line automatically until a new entry is available. For a WTS operation, when the slave is addressed, the *I2Cn_RISR*.WTSR is asserted to notify the CPU.

In CPU mode, FIFO management is based on the assertion of the interrupt bit I2Cn_RISR.RXFNF, related to the nearly-full threshold.

In DMA mode, the single/burst requests are automatically executed based on the number of entries in the Rx FIFO and the related source burst size programmed in *I2Cn_DMAR*.SBSIZE_RX.

0 Reserved for future use. Reading returns 0. Must be written with 0.

I2Cn_TFTR I2C transmit FIFO threshold register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	THRESHOLD_TX		
R	R	R	R	R	R	R	R	R	R	R	R	R	RW		

Address: I2CnBaseAddress + 0x001C

Reset: 0x0000 0000

Description:

THRESHOLD_TX Threshold Tx. Contains the threshold value of the Tx FIFO (in bytes). When the number of entries of the Tx FIFO is less than or equal to the threshold value, *I2Cn_RISR*.TXFNE is set to request the loading of data to the application (in CPU mode).

0 Reserved for future use. Reading returns 0. Must be written with 0.

I2Cn_RFTR I2C receive FIFO threshold register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	THRESHOLD_RX		
R	R	R	R	R	R	R	R	R	R	R	R	R	RW		

Address: I2CnBaseAddress + 0x0020
Reset: 0x0000 0000

Description: Threshold Rx contains the threshold value of the Rx FIFO (in bytes). When the number of entries in the Rx FIFO is greater than or equal to the threshold value, *I2Cn_RISR*.RXFNF is set to request downloading of received data from the application. The application (in CPU mode) downloads the received data based on the threshold.

I2Cn_DMAR I2C DMA register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	BURST_TX	DBSIZE_TX			0	0	0	0	BURST_RX	SBSIZE_RX		
R	R	R	R	RW	RW			R	R	R	R	RW	RW		

Address: I2CnBaseAddress + 0x0024
Reset: 0x0000 0000

Description:

BURST_TX Burst Tx. Defines the type of DMA requests generated by the DMA Tx interface. On completion of the DMA transfer, the DMA Tx interface is turned off, thus clearing *I2Cn_CR*.DMA_TX_EN.
DMA transfer cannot be used if the system memory is organized as a FIFO.

0: single request mode. A single request transfers a single data word in the Tx FIFO (one byte).

1: burst mode. A burst request transfers a programmed burst transfer of words according to *I2Cn_DMAR*.DBSIZE_TX. In burst mode, DMA transfers can also be completed by one or more single requests as required.

DBSIZE_TX Destination burst size Tx. Indicates the number of transfers related to a source burst. This register field is valid only if I2Cn_DMAR.BURST_TX is set. It must be programmed according to the *DMACn_CxCR*.DBSize setting. The destination burst size must be less than the transaction length, otherwise only single requests are generated.

BURST_RX Burst Rx. Defines the type of DMA requests generated by the DMA Rx interface. On completion of the DMA transfer, the DMA Rx interface is turned off, thus clearing I2Cn_CR.DMA_RX_EN.

0: single request mode. A single request transfers a single data word from the Rx FIFO.

1: burst mode. A burst request transfers a programmed burst transfer of words according to I2Cn_DMAR.SBSize. In burst mode, DMA transfers can also be completed by one or more single requests as required.

SBSIZE_RX Source burst size Rx. Indicates the number of transfers related to a source burst. This register field is valid only if I2Cn_DMAR.BURST_RX is set. It must be programmed according to the *DMACn_CxCR*.SBSize setting. The source burst size must be less than the transaction length, otherwise only single requests are generated.

I2Cn_BRCR I2C baud rate counter register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRCNT1															
RW															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRCNT2															
RW															

Address: I2CnBaseAddress + 0x0028

Reset: 0x0000 0008

Description: I

BRCNT1 Baud rate counter 1. Defines the counter value used to setup the I2C baud rate in high-speed mode, when the peripheral is operating in master mode, as described in the equation:

$$Baudrate(high) = \frac{f_{i2cclk}}{BRCNT1 ¥ 1.5}$$

The minimum value allowed is to be determined. *Table 55* includes some examples of settings related to the I2C clock frequency, baud rate and this field.

BRCNT2 Baud rate counter 2. Defines the counter value used to set-up the I2C baud rate in standard and fast mode, when the peripheral is operating in master mode, as described in the equation:

$$Baudrate(fast) = \frac{f_{i2cclk}}{BRCNT2 ¥ 1.5}$$

$$Baudrate(standard) = \frac{f_{i2cclk}}{BRCNT2 ¥ 2}$$

The minimum value allowed is to be determined. *Table 55* includes some setting examples related to the I2C clock frequency, baud rate and this field.

0 Reserved for future use. Reading returns 0. Must be written with 0.

Table 55. I2C baud rate, BRCNTx value, and I2C clock frequency setting

I2CnCLK frequency	Baud rate	BRCNT1 value (decimal)	BRCNT2 value (decimal)
48 MHz	Standard mode 100 kb/s	N.A.	240
26 MHz			130
19.2 MHz			96
48 MHz	Fast mode 400 kb/s	N.A.	80
26 MHz			44
19.2 MHz			32
48 MHz	High-speed mode 3.4 Mb/s	10	not applicable
26 MHz		6	
19.2 MHz		4	

I2Cn_IMSCR I2C interrupt mask set/clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	BERRM	MALM	0	0	0	STDM	MTDM	WTSRM	RFSEM	RFSRM
R	R	R	R	R	R	RW	RW	R	R	R	RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	RXFFM	RXFNFM	RXFEM	TXFOVRM	TXFFM	TXFNEM	TXFEM
R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW

Address: I2CnBaseAddress + 0x002C
Reset: 0x0000 0000

Description: Controls the interrupt bits in the *I2Cn_RISR* register.

BERRM Bus error mask. Enables the interrupt bit BERR.
0: BERR interrupt bit enabled 1: BERR interrupt bit disabled

MALM Master arbitration lost mask. Enables the interrupt bit MAL.
0: MAL interrupt bit enabled 1: MAL interrupt bit disabled

STDM Slave transaction done mask. Enables the interrupt bit STD.
0: STD interrupt bit enabled 1: STD interrupt bit disabled

MTDM Master transaction done mask. Enables the interrupt bit MTD.
0: MTD interrupt bit enabled 1: MTD interrupt bit disabled

WTSRM Write-to-slave request mask. Enables the interrupt bit WTSR.
0: WTSR interrupt bit enabled 1: WTSR interrupt bit disabled

RFSEM Read-from-slave empty mask. Enables the interrupt bit RFSE.
0: RFSE interrupt bit enabled 1: RFSE interrupt bit disabled

RFSRM Read-from-slave request mask. Enables the interrupt bit RFSR.
0: RFSR interrupt bit enabled 1: RFSR interrupt bit disabled

RXFFM Rx FIFO full mask. Enables the interrupt bit RXFF.
0: RXFF interrupt bit enabled 1: RXFF interrupt bit disabled

RXFNFM Rx FIFO nearly full mask. Enables the interrupt bit RXFNF.
0: RXFNF interrupt bit enabled 1: RXFNF interrupt bit disabled

RXFEM Rx FIFO empty mask. Enables the interrupt bit RXFE.
0: RXFE interrupt bit enabled 1: RXFE interrupt bit disabled

TXFOVRM Tx FIFO overrun mask. Enables the interrupt bit TFXOVR.
0: TXFOVR interrupt bit enabled 1: TXFOVR interrupt bit disabled

TXFFM Tx FIFO full mask. Enables the interrupt bit TXFF.
0: TXFF interrupt bit enabled 1: TXFF interrupt bit disabled

TXFNEM Tx FIFO nearly empty mask. Enables the interrupt bit TXFNE.
0: TXFNE interrupt bit enabled 1: TXFNE interrupt bit disabled

TXFEM Tx FIFO empty mask. Enables the interrupt bit TXFE.
0: TXFE interrupt bit enabled 1: TXFE interrupt bit disabled

0 Reserved for future use. Reading returns 0. Must be written with 0.

I2Cn_RISR I2C raw interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	BERR	MAL	0	0	0	STD	MTD	WTSR	RFSE	RFSR
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	RXFF	RXFNF	RXFE	TXFOVR	TXFF	TXFNE	TXFE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: I2CnBaseAddress + 0x0030

Reset: 0x0000 0013

Description: Indicates the interrupt sources prior to masking.

BERR Bus error. Set when an unexpected start/stop condition occurs during a transaction. The action depends on the type of operation in progress. *I2Cn_SR*.CAUSE contains an error tag for this error. This interrupt is cleared by setting the related bit of the *I2Cn_ICR* register.

0: no bus error detection 1: bus error detection

MAL Master arbitration lost. Set when the master loses the arbitration (only for debugging). I2Cn_SR.CAUSE contains an error tag for this error. A collision occurs when two stations transmit two opposite values simultaneously on the serial line. The station pulling up the line identifies the collision reading a 0 value on the SDA input signal, stops transmission, leaves the bus and waits for the idle state (stop condition received) on the bus line before trying the same transaction again. The station that transmits the first unique 0 wins the bus arbitration.
This interrupt is cleared by setting the related bit of the I2Cn_ICR register.

0: no master arbitration lost 1: master arbitration lost

STD Slave transaction done. Set when a slave operation (write-to or read-from) is executed. The application reads the related transaction status (in the I2Cn_SR register), the pending data in the Rx FIFO (only for a write-to-slave operation) and clears the interrupt (transaction ACK). A subsequent slave operation is notified (I2Cn_RISR.WTSR and I2Cn_RISR.RFSR interrupt bit assertions) after the interrupt is cleared, and the I2C clock line is stretched low. A subsequent master operation can also be issued (writing to the *I2Cn_MCR* register) after the interrupt is cleared.
This interrupt is cleared by setting the related bit of the I2Cn_ICR register.

0: slave transaction ACK 1: slave transaction done (ready for ACK)

MTD Master transaction done. Set when a master operation (write or read) is executed. The application reads the related transaction status (I2Cn_SR register), the pending data in the Rx FIFO (only for a master read operation) and clears the interrupt (transaction ACK). A subsequent master operation can be issued (writing the I2Cn_MCR register) after the interrupt is cleared. A subsequent slave operation is notified (I2Cn_RISR.WTSR and I2Cn_RISR.RFSR interrupt bit assertions) after the interrupt is cleared, and the I2C clock line is stretched low.
This interrupt is cleared by setting the related bit of the I2Cn_ICR register.

0: master transaction ACK 1: master transaction done (ready for ACK)

WTSR Write-to-slave request. Set when a write-to-slave operation is received (I2C slave is addressed) from the I2C line. This notification can be used to program the DMA descriptor when required.
This interrupt is cleared by setting the related bit of the I2Cn_ICR register.

0: no write-to-slave request pending 1: write-to-slave request pending

RFSE Read-from-slave empty. Set when a read-from-slave operation is in progress and the Tx FIFO is empty. On assertion of this interrupt, the CPU downloads the data required for the slave operation into the Tx FIFO. This bit is self-cleared by writing in the Tx FIFO. At the end of the read-from-slave operation, this bit is cleared even though the Tx FIFO is empty.

0: Tx FIFO is not empty 1: empty, read-from-slave operation in progress

RFSR Read-from-slave request. Set when a read-from-slave request is received (I2C slave is addressed) from the I2C line. On assertion of this interrupt the Tx FIFO is flushed (pending data is cleared) and the CPU downloads the data required for the slave operation into the Tx FIFO. This bit is self-cleared by writing in the FIFO. If the FIFO is empty before the read operation completes, *I2Cn_RISR*.RFSE is set. This interrupt is cleared by setting the related bit of the *I2Cn_ICR* register.

0: Read-from-slave request is served 1: Read-from-slave request is pending

RXFF Rx FIFO full. Self-cleared when the data is read from the Rx FIFO.

0: Rx FIFO not full 1: Rx FIFO full (only for debugging purpose)

RXFNF Rx FIFO nearly full. Self-cleared when the threshold level is below that of the programmed threshold.

0: number of entries in the Rx FIFO less than *I2Cn_RFTR*.THRESHOLD_RX.

1: number of entries in the Rx FIFO greater than or equal to I2Cn_RFTR.THRESHOLD_RX.

RXFE Rx FIFO empty. Self-cleared when the slave Rx FIFO is not empty.
0: Rx FIFO not empty 1: Rx FIFO empty (only for debugging purpose)

TXFOVR Tx FIFO overrun. Set when a write operation in the Tx FIFO is performed and the Tx FIFO is full. Overflow conditions must be avoided by controlling the data flow. When there is an overrun, the application flushes the transmitter (*I2Cn_CR*.FTX bit set) since the Tx FIFO content is corrupted (at least one word is lost in the FIFO). This interrupt is cleared by setting the related I2Cn_ICR bit.
0: no overrun condition in Tx FIFO 1: overrun condition in Tx FIFO

TXFF Tx FIFO full. Self-cleared when the Tx FIFO is not full.
0: Tx FIFO not full 1: Tx FIFO full (only for debugging purpose)

TXFNE Tx FIFO nearly empty. Self-cleared when the threshold level is over the programmed threshold.
0: number of entries in the Tx FIFO greater than *I2Cn_TFTR*.THRESHOLD_TX.
1: number of entries in the Tx FIFO less than or equal to I2Cn_TFTR.THRESHOLD_TX.

TXFE Tx FIFO empty. Self-cleared by writing in the Tx FIFO.
0: Tx FIFO not empty 1: Tx FIFO empty (only for debugging purpose)

0 Reserved for future use. Reading returns 0. Must be written with 0.

I2Cn_MISR I2C masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	BERR MIS	MAL MIS	0	0	0	STD MIS	MTD MIS	WTSR MIS	RFSE MIS	RFSR MIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	RXFF MIS	RXFNF MIS	RXFE MIS	TXFOV RMIS	TXFF MIS	TXFNE MIS	TXFE MIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: I2CnBaseAddress + 0x0034

Reset: 0x00000000

Description: Indicates the interrupt sources after masking. For bit descriptions, refer to *I2Cn_RISR*.
An output signal is asserted when at least one interrupt source of this register is pending.

I2Cn_ICR I2C interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	BERR IC	MAL IC	0	0	0	STD IC	MTD IC	WTSR IC	RFSE IC	RFSR IC
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	TXFOV RIC	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: I2CnBaseAddress + 0x0038

Reset: 0x0000 0000

Description: Indicates the interrupt sources after masking. For bit descriptions, refer to *I2Cn_RISR*. Writing 1 to a bit in this register clears the corresponding bit in the status register.
Data bits set to 0 have no effect on their corresponding bit in the status register.

I2Cn_PeriphID0 I2C peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: I2CnBaseAddress + 0xFE0

Reset: 0x0000 0024

Description: Part Number0 reads back as 0x24.

I2Cn_PeriphID1 I2C peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: I2CnBaseAddress + 0xFE4

Reset: 0x0000 0000

Description: Designer0 and Part Number1 both read back as 0x0.

I2Cn_PeriphID2 I2C peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: I2CnBaseAddress + 0xFE8

Reset: 0x0000 0018

Description: Revision returns the peripheral revision (0x1) and Designer1 reads back as 0x8.

I2Cn_PeriphID3 I2C peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: I2CnBaseAddress + 0xFEC

Reset: 0x0000 0000

Description: Configuration reads back as 0x00.

I2Cn_PCellID0 I2C PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	I2CPCellID0							
R	R	R	R	R	R	R	R	R							

Address: I2CnBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: I2CPCellID0 reads back as 0x0D.

I2Cn_PCellID1 I2C PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	I2CPCellID1							
R	R	R	R	R	R	R	R	R							

Address: I2CnBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: I2CPCellID1 reads back as 0xF0.

I2Cn_PCellID2 I2C PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	I2CPCellID2							
R	R	R	R	R	R	R	R	R							

Address: I2CnBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: I2CPCellID2 reads back as 0x05.

I2Cn_PCellID3 I2C PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	I2CPCellID3							
R	R	R	R	R	R	R	R	R							

Address: I2CnBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: I2CPCellID3 reads back as 0xB1.

26 1-Wire master (OWM)

26.1 OWM register addressing

Register addresses are provided as the 1-Wire[c] master base address, OWMBaseAddress, plus the register offset.

The OWMBaseAddress is 0x101E A000.

26.2 OWM register summary

Table 56. OWM register list

Offset	Register	Description	Page
0x000	OWM_CR	OWM control register	349
0x004	OWM_DR	OWM data register	350
0x008	OWM_IMSC	OWM interrupt mask set/clear register	351
0x00C	OWM_RIS	OWM raw interrupt status register	352
0x010	OWM_MIS	OWM masked interrupt status register	353
0x014	OWM_ICR	OWM interrupt clear register	353
0xFE0	OWM_PeriphID0	OWM peripheral identification register 0 (bits 7:0)	354
0xFE4	OWM_PeriphID1	OWM peripheral identification register 1 (bits 15:8)	354
0xFE8	OWM_PeriphID2	OWM peripheral identification register 2 (bits 23:16)	355
0xFEC	OWM_PeriphID3	OWM peripheral identification register 3 (bits 31:24)	355
0xFF0	OWM_PCellID0	OWM PCell identification register 0 (bits 7:0)	355
0xFF4	OWM_PCellID1	OWM PCell identification register 1 (bits 15:8)	356
0xFF8	OWM_PCellID2	OWM PCell identification register 2 (bits 23:16)	356
0xFFC	OWM_PCellID3	OWM PCell identification register 3 (bits 31:24)	356

c. 1-Wire is a registered trademark of Dallas Semiconductor.

26.3 OWM register descriptions

OWM_CR OWM control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PDR	0	0	TSLOT		1BIT	1WMOD	1WRST
R	R	R	R	R	R	R	R	R	R	R	R		RW	RW	RW

Address: OWMBaseAddress + 0x000

Reset: 0x0000 0000

Description: Configures the 1-wire master (OWM).

PDR Presence detect read. Occurs on a presence detect interrupt. If a presence pulse is detected after a reset pulse a slave device is detected.

0: no slave device detected 1: slave device detected

TSLOT Bit time slot. Sets the bit slot timing durations. See *Table 57* and the datasheet for timing details.

1BIT Single bit mode. Defines if read or write operations are for a single bit or a byte.

0: byte mode, master operates in full byte boundaries. First bit is LSB and last bit is MSB of the byte.

1: single bit mode, only the least significant bit of the transmit/receive register would be sent/received before enabling the interrupt flags that signal the end of the transmission.

1WMOD 1-wire mode. Defines the mode of operation for reading a bit from a slave.

0: master does not pull the bus line low to initiate a bit-read from the slave. Presence detection does not occur after a reset pulse, and PDR is forced to 1.

1: master pulls the bus line low to initiate a bit-read from the slave. Presence detection phase occurs after a reset pulse. The result of this presence detection phase is stored in PDR.

1WRST 1-wire reset. Writing 1 generates a reset on the 1-wire bus. This bit is automatically cleared as soon as the 1-wire reset completes. The 1-wire master sets the presence detect interrupt flag (OWMPDINTR) when the reset is complete and sufficient time has passed for a presence detect to occur. The result of the presence detect is placed in the raw interrupt status bit PDRIS. If a presence detect pulse was received, DPRIS is cleared, otherwise it is set.

0 Reserved for future use. Reading returns 0. Must be written with 0.

Table 57. Time slots (τ = 1 μs)

TSLOT	Bit time slot (tSLOT)	Write 0 time (tW0)	Write 1 time (tW1)	Read start (tR0)(1)	Read strobe (tRD)	Reset time high (tRSTH)	Reset time low (tRSTL)	Presence detect strobe (tPDS)(2)	Time-out(3)
00	11 τ	8 τ	1 τ	1 τ 1)	2 τ	50 τ	61 τ	3 τ 3)	6 τ 3)
01	73 τ	63 τ	6 τ	1 τ 1)	15 τ	500 τ	488 τ	30 τ 3)	60 τ 3)
10	223 τ	127 τ	6 τ	1 τ 1)	79 τ	500 τ	488 τ	30 τ 3)	60 τ 3)
11	223 τ	127 τ	6 τ	1 τ 1)	79 τ	50 τ	223 τ	30 τ 3)	360 τ 3)

1. Read start phase occurs if OWM_CR.1WMOD = 1.
2. Presence detect phase occurs if OWM_CR.1WMOD = 1.

3. If OWM_CR.1WMOD = 1, the 1-wire master waits for a falling edge on the OWMDQ line to be detected after a reset for up to this time-out amount of time. If OWM_CR.1WMOD = 0, the 1-wire master waits for a falling edge.

OWM_DR OWM data register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	DATA							
R	R	R	R	R	R	R	R	RW							

Address: OWMBaseAddress + 0x004

Reset: 0x0000 0000

Description: Data exchanged with the 1-wire master passes through this data register.

The 1-wire master is double buffered with separate transmit and receive buffers. Writing to this location connects the transmit buffer to the data bus, while reading connects the receive buffer to the data bus.

To send a byte on the 1-wire bus, the user writes data to the transmit buffer. This data is then moved to the transmit shift register where it is shifted serially onto the bus LSB first. A new byte of data can then be written to the transmit buffer. As soon as the transmit shift register is empty the data is transferred from the transmit buffer and the process repeats.

Each of these registers has a flag that may be used as interrupt sources. The transmit buffer empty (TBE) flag is set when the transmit buffer is empty and ready to accept a new byte. As soon as a byte is written into the transmit buffer, TBE is cleared. The transmit shift register empty (TEMT) flag is set when the shift register has no data in it and is ready to accept a new byte. As soon as a byte of data is transferred from the transmit buffer, TEMT is cleared and TBE is set. Note that the 1-wire protocol requires a reset before any bus communication.

Before reading data from a slave device, the device must be ready to transmit data depending on commands already received from the CPU. Data is retrieved from the bus in a similar fashion to a write operation.

The host initiates a read by writing to the transmit buffer.

The data that is then shifted into the receive shift register is the wired-AND of the written data and the data from the slave device.

Therefore in order to read a byte from a slave device the host must write 0xFF.

When the receive shift register is full the data is transferred to the receive buffer where it can be accessed by the host. Additional bytes can now be read by sending 0xFF again.

If the slave device is not ready to transmit, the data received is identical to that which was transmitted.

The receive buffer register can also generate interrupts. The receive buffer flag (RBF) is set when data is transferred from the receive shift register and cleared when the host reads the register. If RBF is set, no further transmissions should be made on the 1-wire bus or data may be lost as the

byte in the receive buffer is overwritten by the next received byte.
See the datasheet for operational details. Generating a 1-wire reset on the bus is covered under command operations.

DATA Transmit/receive data.
Read: receive byte (or bit 0 in 1-bit mode) is read.
Write: transmit byte (or bit 0 in 1-bit mode) is written.

0 Reserved for future use. Reading returns 0. Must be written with 0.

OWM_IMSC OWM interrupt mask set/clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	OWLIM	OWSIM	RSRFIM	RBFIM	TSREIM	TBEIM	PDIM
R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW

Address: OWMBaseAddress + 0x008

Reset: 0x0000 0000

Description: Interrupt mask set or clear register. Returns the current value of the mask on the relevant interrupt. A write of 1 to the particular bit sets the mask, enabling the interrupt to be read. A write of 0 clears the corresponding mask. All the bits are cleared to 0 when reset.

OWLIM 1-wire low interrupt (OWLINT) mask.
0: masked 1: not masked

OWSIM 1-wire short interrupt (OWSINT) mask.
0: masked 1: not masked

RSRFIM Receive shift register full interrupt (OWRSRFINT) mask.
0: masked 1: not masked

RBFIM Receive buffer full interrupt (OWRBFINT) mask.
0: masked 1: not masked

TSREIM Transmit shift register empty interrupt (OWTSREINT) mask.
0: masked 1: not masked

TBEIM Transmit buffer empty interrupt (OWTBEINT) mask.
0: masked 1: not masked

PDIM Presence detect interrupt (OWPDINT) mask.
0: masked 1: not masked

0 Reserved for future use. Reading returns 0. Must be written with 0.

OWM_RIS OWM raw interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	OWL RIS	OWS RIS	RSRF RIS	RBF RIS	TSRE RIS	TBE RIS	PD RIS
R	R	R	R	R	R	R	R	R	RH	RH	RH	RH	RH	RH	RH

Address: OWMBaseAddress + 0x00C

Reset: 0x0000 0046

Description: Returns the raw status of the interrupt prior to masking. A write has no effect.

OWLRIS 1-wire low raw interrupt status. Set after reset.

1: 1-wire line is low while the master is in idle, indicating that a slave device has issued a presence pulse on the 1-wire (OWMDQ) line.

OWSRIS 1-wire short raw interrupt status. Set after reset because GPIO forces the OWM input line low.

1: 1-wire line was low before the master sent out the beginning of a reset or a time slot.

0: 1-wire line was high as expected prior to all resets and time slots.

RSRFRIS Receive shift register full raw interrupt status.

1: a byte of data is waiting in the receive shift register.

0: receive shift register is either empty or currently receiving data. This bit is cleared by the hardware when data in the receive shift register is transferred to the receive buffer.

RBFRIS Receive buffer full raw interrupt status.

1: a byte of data is waiting to be read in the receive buffer.

0: receive buffer has no new data to be read. This bit is cleared when the byte is read from the receive buffer, except if another byte is waiting in the receive shift register when bit RSRFRIS = 1.

TSRERIS Transmit shift register empty raw interrupt status. Set after reset.

1: transmit shift register is empty and is ready to receive the next byte of data from the transmit buffer.

0: transmit shift register is busy sending out data. This bit is cleared when data is transferred from the transmit buffer to the transmit shift register.

TBERIS Transmit buffer empty raw interrupt status. Set after reset.

1: transmit buffer is empty and is ready to receive the next byte of data.

0: transmit buffer is waiting for the transmit shift register to finish sending its current data before updating it. This bit is cleared when data is written to the transmit buffer.

PDRIS Presence detect raw interrupt status.

1: 1-wire reset has been issued and an appropriate amount of time has passed for a presence detect pulse to have occurred.

0: 1-wire reset has not been issued by the master since the previous read of the interrupt register.

0 Reserved for future use. Reading returns 0. Must be written with 0.

OWM_MIS OWM masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	OWL MIS	OWS MIS	RSRF MIS	RBF MIS	TSRE MIS	TBE MIS	PD MIS
R	R	R	R	R	R	R	R	R	RH	RH	RH	RH	RH	RH	RH

Address: OWMBaseAddress + 0x010

Reset: 0x0000 0000

Description: Returns the masked status of the interrupt after masking. A write has no effect.

OWLMIS 1-wire low masked interrupt status: masked interrupt state of the 1-wire low interrupt.

OWSMIS 1-wire short masked interrupt status: masked interrupt state of the 1-wire short interrupt.

RSRFMIS Receive shift register full masked interrupt status: masked interrupt state of the receive shift register full interrupt.

RBFMIS Receive buffer full masked interrupt status: masked interrupt state of the receive buffer full interrupt.

TSREMIS Transmit shift register empty masked interrupt status: masked interrupt state of the transmit shift register empty interrupt.

TBEMIS Transmit buffer empty masked interrupt status: masked interrupt state of the transmit buffer empty interrupt.

PDMIS Presence detect masked interrupt status: masked interrupt state of the presence detect interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

OWM_ICR OWM interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	OWLIC	OWSIC	0	0	0	0	PDIC
R	R	R	R	R	R	R	R	R	W	W	R	R	R	R	W

Address: OWMBaseAddress + 0x014

Reset: 0x0000 0000

Description: Clears the corresponding interrupt.

OWLIC 1-wire low interrupt clear.
0: no effect 1: corresponding 1-wire low interrupt is cleared

OWSIC 1-wire short interrupt clear.
0: no effect 1: corresponding 1-wire short interrupt is cleared

PDIC Presence detect interrupt clear.
0: no effect 1: corresponding presence detect interrupt cleared

0 Reserved for future use. Reading returns 0. Must be written with 0.

OWM_PeriphID0 OWM peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: OWMBaseAddress + 0xFE0

Reset: 0x0000 0023

Description: PartNumber0 reads back as 0x23.

OWM_PeriphID1 OWM peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: OWMBaseAddress + 0xFE4

Reset: 0x0000 0000

Description: Designer0 reads back as 0x00. PartNumber1 reads back as 0x00.

OWM_PeriphID2 OWM peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: OWMBaseAddress + 0xFE8

Reset: 0x0000 0008

Description: Revision returns the peripheral revision. Designer1 reads back as 0x08.

OWM_PeriphID3 OWM peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: OWMBaseAddress + 0xFEC

Reset: 0x0000 0000

Description: Configuration reads back as 0x00.

OWM_PCellID0 OWM PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	OWMPCellID0							
R	R	R	R	R	R	R	R	R							

Address: OWMBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: Returns 0x0D.

OWM_PCellID1 OWM PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	OWMPCellID1							
R	R	R	R	R	R	R	R	R							

Address: OWMBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: Returns 0xF0.

OWM_PCellID2 OWM PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	OWMPCellID2							
R	R	R	R	R	R	R	R	R							

Address: OWMBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: Returns 0x05.

OWM_PCellID3 OWM PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	OWMPCellID3							
R	R	R	R	R	R	R	R	R							

Address: OWMBaseAddress + 0xFFC

Reset: 0x0000 00B1

Description: Returns 0xB1.

27 SD card interface (SDI)

The device communicates to the system via 32-bit-wide control registers accessible via AMBA rev. 2.0 Peripheral Bus (APB). These registers are listed in *Table 58* and are described in details in the following pages.

27.1 SDI register summary

Table 58. SDI register list

Offset	Register name	Description	Page
0x00	SDI_PWR	SDI Power Control Register	*358*
0x04	SDI_CLKCR	SDI Clock Control Register	*359*
0x08	SDI_ARG	SDI Argument Register	*360*
0x0C	SDI_CMD	SDI Command Register	*361*
0x10	SDI_RESPCMD	SDI Command Response Register	*362*
0x14	SDI_RESP0	SDI Response Registers 0..3	*362*
0x18	SDI_RESP1	SDI Response Registers 0..3	*362*
0x1C	SDI_RESP2	SDI Response Registers 0..3	*362*
0x20	SDI_RESP3	SDI Response Registers 0..3	*362*
0x24	SDI_DTIMER	SDI Data Timer Register	*363*
0x28	SDI_DLEN	SDI Data Length Register	*363*
0x2C	SDI_DCTRL	SDI Data Control Register	*363*
0x30	SDI_DCOUNT	SDI Data Counter Register	*364*
0x34	SDI_STA	SDI Status Register	*365*
0x38	SDI_ICR	SDI Interrupt Clear Register	*366*
0x3C	-	Reserved	*367*
0x40	-	Reserved	*367*
0x48	SDI_FIFOCNT	SDI FIFO Counter Register	*368*
0x04C to 0x07C	-	Reserved	-
0x80	SDI_FIFO	SDI Data FIFO Register	*368*
0x0C0 to 0xFDC	-	Reserved	-
0xFE0	SDIPeriphID0	SDI Peripheral Identification Register 0	*369*
0xFE4	SDIPeriphID1	SDI Peripheral Identification Register 1	*369*
0xFE8	SDIPeriphID2	SDI Peripheral Identification Register 2	*369*
0xFEC	SDIPeriphID3	SDI Peripheral Identification Register 3	*369*

Table 58. SDI register list (continued)

Offset	Register name	Description	Page
0xFF0	SDIPCellID0	SDI PCell Identification Register 0	370
0xFF4	SDIPCellID1	SDI PCell Identification Register 1	370
0xFF8	SDIPCellID2	SDI PCell Identification Register 2	370
0xFFC	SDIPCellID3	SDI PCell Identification Register 3	371

27.2 SDI register descriptions

SDI_PWR — SDI Power Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							DAT74DIREN	FBCLKEN	OPD	DAT31DIREN	DAT0DIREN	CMDDIREN	DAT2DIREN	PWRCTRL	
R																							R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Address: SDIBaseAddress + 0x00

Type: R/W

Reset: 0x00000000

Description: The bit assignments of the SDIPWR register are shown below

Note: After a data write, data cannot be written to this register for three MCLK (48 MHz clock that drives the SDI logic) clock periods plus two PCLK (Peripheral bus clock) clock periods.

[8] **DAT74DIREN:** SDIDAT74DIR Feedback Enable bit. Defines if the direction of SDIDAT[7:4] signals are using or not the SDIDAT74DIR signal:

0: the SDIDAT74DIR signal is not used to control SDIDAT[7:4] signal directions. The SDIDAT74DIR pin can be used in GPIO function.

1: the SDIDAT74DIR signal is used to control SDIDAT[7:4] signal directions. GPIO[?] must be configured in alternate function A for correct behavior.

[7] **FBCLKEN:** Feedback Clock Enable bit. Determines which signal is used to internally latch the card inputs:

0: the SDICLK signal is used to latch the card inputs

1: the SDIFBCLK signal is used to latch the card inputs. This must be used when the external buffers on card signals have a propagation time of more than 4 ns. The SDIFBCLK pin must then be driven by the SDICLK signal delayed by the in and out propagation times of external buffer.

[6] **OPD:** SDICMD Output Control bit. Control the output mode of the SDICMD signal:

0: the SDICMD signal is in Push-Pull output mode

1: the SDICMD signal is in Open-Drain output mode

[5] **DAT31DIREN:** SDIDAT31DIR Feedback Enable bit. Defines if the direction of SDIDAT[3][1] signals are using or not the SDIDAT31DIR signal:

0: the SDIDAT31DIR signals not used to control SDIDAT[3][1] signal directions. The SDIDAT31DIR pin can be used in GPIO function.

1: the SDIDAT31DIR signal is used to control SDIDAT[3][1] signal directions. GPIO[16] must be configured in alternate function A for correct behavior.

[4] **DAT0DIREN:** SDIDAT0DIR Feedback Enable bit. Defines if the direction of SDIDAT[0] signal is using or not the SDIDAT0DIR signal:

0: the SDIDAT0DIR signal is not used to control SDIDAT[0] signal direction. The SDIDAT0DIR pin can be used in GPIO function.

1: the SDIDAT0DIR signal is used to control SDIDAT[0] signal directions. GPIO[15] must be configured in alternate function A for correct behavior.

[3] **CMDDIREN:** SDICMDDIR Feedback Enable bit. Defines if the direction of SDICMD signal is using or not the SDICMDDIR signal:

0: the SDICMDDIR signal is not used to control SDICMD signal direction. The SDICMDDIR pin can be used in GPIO function.

1: the SDICMDDIR signal is used to control SDICMD signal directions. GPIO[10] must be configured in alternate function A for correct behavior.

[2] **DAT2DIREN:** SDIDAT2DIR Feedback Enable bit. Defines if the direction of SDIDAT[2] signals are using or not the SDIDAT2DIR signal

0: the SDIDAT2DIR signal is not used to control SDIDAT[2] signal direction. The SDIDAT2DIR pin can be used in GPIO function.

1: the SDIDAT2DIR signal is used to control SDIDAT[2] signal directions. GPIO[?] must be configured in alternate function A for correct behavior.

[1:0] **PWRCTRL:** Power Supply Control bits. These bits allow to define the current functional state of the card clock:

00: Power Off: the clock to card is stopped.

01: reserved

10: reserved

11: Power On: the card is clocked.

Note: After a data write, data cannot be written to this register for three MCLK (48 MHz clock that drives the SDI logic) clock periods plus two PCLK (Peripheral bus clock) clock periods.

SDI_CLKCR SDI Clock Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		NEGEDG	WIDBUS		BYPASS	PWRSAV	CLKEN	CLKDIV							
R																		R/W	R/W		R/W	R/W	R/W	R/W							

Address: SDIBaseAddress + 0x04

Type: R/W

Reset: 0x00000000

Description: The SDI_CLKCR register controls the SDI output clock. While the SD Card or MultiMediaCard is in identification mode, the SDICLK frequency must be less than

400 kHz. YThe clock frequency can be changed to the maximum card bus frequency when relative card addresses are assigned to all cards. After a data write, data cannot be written to this register for 3 MCLK plus 2 PCLK clock periods.

[13] **NEGEDG:** MCICLKOUT dephasing selection bit

0: Command and Data are generated on falling edge of MCICLKOUT (expect at 48MHz)

1: Command and Data are generated on rising edge of MCICLKOUT (this bit must be set only at 48MHz to generate Command and Data on falling edge)

[12:11] **WIDBUS:** Wide Bus Mode Enable bit

00: Default bus mode: SDIDAT[0] used.

01: 4Wide bus mode: SDIDAT[3:0] used.

10: 8Wide bus mode: SDIDAT[7:0] used

[10] **BYPASS:** Clock Divider Bypass enable bit

0: Disable bypass: MCLK is divided according CLKDIV value before driving the SDICLK output signal.

1: Enable bypass: MCLK drive directly the SDICLK output signal.

[9] **PWRSAV:** Power Saving configuration bit. For power saving, the SDICLK clock output can be disabled when the bus is idle by setting this bit:

0: SDICLK clock is always enabled.

1: SDICLK is only enabled when bus is active.

[8] **CLKEN:** Clock Enable bit

0: SDICLK is disabled.

1: SDICLK is enabled.

[7:0] **CLKDIV:** Clock Divide factor. This field defines the divide factor between the input clock (48 MHz) and the SDICLK output clock:SDICLK frequency = 48 MHz / [CLKDIV + 2].

SDI_ARG SDI Argument Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMDARG																															
R/W																															

Address: SDIBaseAddress + 0x08

Type: R/W

Reset: 0x00000000

Description: The SDI_ARG register contains a 32-bit command argument, which is sent to a card as part of a command message. Table585 shows the bit assignment of the SDI_ARG register.

[31:0] **CMDARG:** Command Argument. Command Argument sent to a card as part of a command message.If a command contains an argument, it must be loaded into this register before writing a command to the command register.

SDI_CMD SDI Command Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	CE-ATACMD	nIEN	ENCMDcompl	RESERVED	CPSMEN	WAITPEND	WAITINT	LONGRESP	WAITRESP	CMDINDEX						
R																	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W						

Address: SDIBaseAddress + 0x0C

Type: R/W

Reset: 0x00000000

Description: The SDI_CMD register contains the command index and command type bits.The command index is sent to a card as part of a command message.The command type bits control the Command Path State Machine (CPSM). *Table 59* shows the response types:

Note: After a data write, data cannot be written to this register for three MCLK (48 MHz clock that drives the SDI logic) clock periods plus two PCLK (Peripheral bus clock) clock periods.

[14] **ATACMD:** CE-ATA command. If set CPSM is transferring CMD61

[13] **nIEN:** not Interrupt ENABLEif ‘0 interrupt in CE-ATA device are enabled

[12] **ENCMDcompl:** Enable CMD completion. If set command completion signal is enabled

[11] RESERVED

[10] **CPSMEN:** Command Path State Machine (CPSM) Enable bit. If set, CPSM is enabled.

[9] **WAITPEND:** CPSM Waits for ends of data transfer (CmdPend internal signal)If set, CPSM waits for end of data transfer before it starts sending a command.

[8] **WAITINT:** CPSM Waits for Interrupt Request. If set, CPSM disables command timeout and waits for an interrupt request.

[7] **LONGRESP:** Long Response configuration bit. If set, CPSM receives a 136-bit long response.

[6] **WAITRESP:** CPSM Waits for a Response. If set, CPSM waits for a response.

[5:0] **CMDINDEX:** Command Index. Command Index is sent to the card as part of a command message.

Table 59. Command response types

WAITRESP	LONGRESP	Description
0	0	No response, expect CmdSent flag
0	1	No response, expect CmdSent flag
1	0	Short response, expect CmdRespEnd or CmdCrcFail flag
1	1	Long response, expect CmdRespEnd or CmdCrcFail flag

Note: MultiMediaCard can send two kinds of response: short response, 48 bits long, or long response,136 bits long (type R2). SD-Card can send only short responses. Short response format is as showed in Table 60 but the argument can vary according to the type of

response(R1 or R1b, R3, R4, R5, R6 or R6 modified): the software will distinguish the type of response according to the sent command. CE-ATA Devices send only short responses, type R1 or R1b according to the direction of the transfer.

SDI_RESPCMD SDI Command Response Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										RESPCMD					
R																										R					

Address: SDIBaseAddress + 0x10

Type: R

Reset: 0x00000000

Description: The SDI_RESPCMD register contains the command index field of the last command response received. Table588 shows the bit assignment of the SDI_RESPCMD register.

[5:0] **RESPCMD:** Response Command Index. Read-only bit field.Contains the command index of the last command response received.

SDI_RESPx SDI Response Registers 0..3 (x= 0 to 3)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDI_RESP0	CARDSTATUS																															
	R																															

Address: SDIBaseAddress + 0x14 + 0x4*x (x=0 to 3)

Type: R

Reset: 0x00000000

Description: The SDI_RESP0..3 registers contain the status of a card, which is part of the receive response.The Card Status size is 32 or 127 bits, depending on the response type.

SDI_RESP0: [31:0] **CARDSTATUS:** Card status. See *Table 60*

Table 60. Response Type and SDI_RESPx registers

Register	Short Response	Long Response
SDI_RESP0	Card Status[31:0]	Card Status[127:96]
SDI_RESP1	Unused	Card Status[95:64]
SDI_RESP2	Unused	Card Status[63:32]
SDI_RESP3	Unused	Card Status[31:1] | 0b

Note: The most significant bit of the card status is received first. The SDIRESP3 register LSB is always 0b.

SDI_DTIMER SDI Data Timer Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATATIME																															
R/W																															

Address: SDIBaseAddress + 0x24

Type: R/W

Reset: 0x00000000

Description: The SDI_DTIMER register contains the data timeout period, in card bus clock periods. Table590 shows the bit assignment of the SDI_DTIMER register.
A counter loads the value from the SDI_DTIMER register, and starts decrementing when the *Data Path State Machine* (DPSM) enters the WAIT_R or BUSY state. If the timer reaches 0 while the DPSM is in either of these states, the timeout status flag is set.
A data transfer must be written to the data timer register and the data length register before being written to the data control register.

[31:0] **DATATIME:** Data Timeout Period. Data timeout period expressed in card bus clock periods.

SDI_DLEN SDI Data Length Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							DATALENGTH																								
R							R																								

Address: SDIBaseAddress + 0x28

Type: R

Reset: 0x00000000

Description: The SDI_DLEN register contains the number of data bytes to be transferred. The value is loaded into the data counter when data transfer starts. Table591 shows the bit assignment of the SDI_DLEN register
For a block data transfer, the value in the data length register must be a multiple of the block size (see *Section : SDI_DCTRL*).
A data transfer must be written to the data timer register and the data length register before being written to the data control register.

[24:0] **DATALENGTH:** Data Length Value. Number of data bytes to be transferred.

SDI_DCTRL SDI Data Control Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				RESERVED	RWMOD	RWSTOP	RWSTART	DBLOCKSIZE				DMAEN	DTMODE	DTDIR	DTEN
R																				R/W	R/W	R/W	R/W	R/W				R/W	R/W	R/W	R/W

Address: SDIBaseAddress + 0x2C

Type: R/W

Reset: 0x00000000

Description: The SDI_DCTRL register control the Data Path State Machine (DPSM). Table592 shows the bit assignment of the SDI_DCTRL register.

Note: After a data write, data cannot be written to this register for three MCLK (48 MHz clock that drives the SDI logic) clock periods plus two PCLK (Peripheral bus clock) clock periods.

[10] **RWMOD:** Read Wait Mode
0: Read Wait control by stopping SDICLK
1: Read Wait control using SDIDAT[2]

[9] **RWSTOP:** Read Wait Stop
0: Read Wait in progress if RW Start bit is set
1: Enable for Read Wait stop if RW Start bit is set

[8] **RWSTART:** Read Wait Start. If set read wait operation starts

[7:4] **DBLOCKSIZE:** Data Block Size. Defines the data block length when the block data transfer mode is selected:..
0000: (0) Block length = 20 = 1 byte
0001: Block length = 21 = 2 bytes
1011: (11) Block length = 211 = 2048 bytes
1100: (12) Block length = 212 = 4096 bytes
1101: (13) Block length = 213 = 8192 bytes
1110: (14) Block length = 214 = 16384 bytes
1111: (15) reserved

[3] **DMAEN:** DMA Enable bit
0: DMA disabled.
1: DMA enabled.

[2] **DTMODE:** Data Transfer Mode selection
0: Block data transfer.
1: Stream data transfer.

[1] **DTDIR:** Data Transfer Direction selection
0: From controller to card.
1: From card to controller.

[0] **DTEN:** Data Transfer Enabled bit. Data transfer starts if 1b is written to the DTEN bit. Depending on the direction bit, DTDIR, the DPSM moves to the WAIT_S,WAIT_R state or READWAIT if RW Start is set immediately at the beginning of transfer.You do not need to clear the enable bit after the end of a data transfer but you need to update the MCIDataCtrl to enable a new data transfer

SDI_DCOUNT SDI Data Counter Register

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
RESERVED	DATACOUNT
R	R

Address: SDIBaseAddress + 0x30

Type: R

Reset: 0x00000000

Description: The SDI_DCOUNT register loads the value from the data length register (see section 2758: SDI Data Length Register SDI_DLEN on page15) when the DPSM moves from IDLE state to the WAIT_R or WAIT_S state. As data is transferred, the counter decrements the value until it reaches 0. The DPSM then moves to the IDLE state and the data status end flag, DATAEND, is set. Table593 shows the bit assignment of the SDI_DCOUNT register.

Note: This register should be read only when the data transfer is complete.

[24:0] **DATACOUNT:** Data Count Value. When read, returns the number of remaining data bytes to be transferred.Write has no effect.

SDI_STA SDI Status Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CEATAEND	RESERVED	RXDAVL	TXDAVL	RXFIFOE	TXFIFOE	RXFIFOF	TXFIFOF	RXFIFOBR	TXFIFOBW	RXACT	TXACT	CMDACT	DBCKEND	STBITERR	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL
R								R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: SDIBaseAddress + 0x34

Type: R

Reset: 0x00000000

Description: The SDI_STA register is a read-only register. It contains two type of flag:

- Static (bits [23:22,10:0]): These bits remain asserted until they are cleared by writing to the SDICLR register (see section 27512: SDI Interrupt Clear Register SDI_ICR on page22).
- Dynamic (bits [21:11]): These bits change state depending on the state of the underlying logic (for example, FIFO full and empty flags are asserted and deasserted as data while written to the FIFO).

Note: The status bits RXFIFOBR *and* TXFIFOBW *replace the status bits* RXFIFOHF *and* TXFIFOHE*. They have the same meaning, ie there are at least 8 words in the FIFO and at least 8 words can be written in the FIFO*.

[23] **CEATAEND:** CE-ATA command completion signal received for CMD61

[21] **RXDAVL:** Data available in receive FIFO

[20] **TXDAVL:** Data available in transmit FIFO

[19] **RXFIFOE:** Receive FIFO Empty

[18] **TXFIFOE:** Transmit FIFO Empty

[17] **RXFIFOF:** Receive FIFO Full

[16] **TXFIFOF:** Transmit FIFO Full

[15] **RXFIFOBR:** Receive FIFO Burst Readable: there is at least a burst (8 words) in the FIFO.

[14] **TXFIFOBW:** Transmit FIFO Burst Writable: at least a burst (8 words) can be written in the FIFO.

[13] **RXACT:** Data receive in progress

[12] **TXACT:** Data transmit in progress

[11] **CMDACT:** Command transfer in progress

[10] **DBCKEND:** Data Block sent/received (CRC check passed)

[9] **STBITERR:** Start Bit not detected on all data signals in wide bus mode

[8] **DATAEND:** Data End (Data Counter, SDIDCOUNT, is zero)

[7] **CMDSENT:** Command Sent (no response required)

[6] **CMDREND:** Command Response received (CRC check passed)

[5] **RXOVERR:** Received FIFO Overrun error

[4] **TXUNDERR:** Transmit FIFO Underrun error

[3] **DTIMEOUT:** Data Timeout

[2] **CTIMEOUT:** Command Response Timeout

[1] **DCRCFAIL:** Data block sent/received (CRC check failed)

[0] **CCRCFAIL:** Command Response received (CRC check failed)

SDI_ICR SDI Interrupt Clear Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CEATAENDC	RESERVED	RESERVED											DBCKENDC	STBITERRC	DATAENDC	CMDSENTC	CMDRENDC	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAIL	CCRCFAIL
R								R/W	R/W	R											R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: SDIBaseAddress + 0x38

Type: R/W

Reset: 0x00000000

Description: The SDI_ICR register is a write-only register. Writing a bit with 1b clears the corresponding bit in the Status register, SDI_STA. Table595 shows the bit assignment of the SDI_ICR register.

Note: The mask bits for RXFIFOBR *and* TXFIFOBW *replace the mask bits for* RXFIFOHF *and* TXFIFOHE*. They have the same meaning.*

[23] **CEATAENDC:** CEATAEND flag Clear bit

[10] **DBCKENDC:** DBCKEND flag Clear bit

[9] **STBITERRC:** STBITERR flag Clear bit

[8] **DATAENDC:** DATAENDC flag Clear bit

[7] **CMDSENTC:** CMDSENT flag Clear bit

[6] **CMDRENDC:** CMDREND flag Clear bit

[5] **RXOVERRC:** RXOVERR flag Clear bit

[4] **TXUNDERRC:** TXUNDERR flag Clear bit

[3] **DTIMEOUTC:** DTIMEOUT flag Clear bit

[2] **CTIMEOUTC:** CTIMEOUT flag Clear bit

[1] **DCRCFAIL:** DCRCFAIL flag Clear bit

[0] **CCRCFAIL:** CCRCFAIL flag Clear bit

SDI_MASK SDI Mask Register

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDI_MASK0	RESERVED								MASK23	RESERVED	MASK21	MASK20	MASK19	MASK18	MASK17	MASK16	MASK15	MASK14	MASK13	MASK12	MASK11	MASK10	MASK9	MASK8	MASK7	MASK6	MASK5	MASK4	MASK3	MASK2	MASK1	MASK0
	R								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: SDIBaseAddress + 0x3C + 0x4*x (x=0 to 1)

Type: R/W

Reset: 0x00000000

Description: There are two interrupt mask registers, SDI_MASK0 and SDI_MASK1, one for each interrupt request signal. The interrupt mask registers determines which status flags generate an interrupt request by setting the corresponding bit to 1b. Table596 shows the bit assignment of the SDI_MASK register.

Note:

SDI_MASK0: [23] **MASK23:** Mask CEATAEND flag

SDI_MASK0: [21] **MASK21:** Mask RXDAVL flag

SDI_MASK0: [20] **MASK20:** Mask TXDAVL flag

SDI_MASK0: [19] **MASK19:** Mask RXFIFOE flag

SDI_MASK0: [18] **MASK18:** Mask TXFIFOE flag

SDI_MASK0: [17] **MASK17:** Mask RXFIFOF flag

SDI_MASK0: [16] **MASK16:** Mask TXFIFOF flag

SDI_MASK0: [15] **MASK15:** Mask RXFIFOBR flag

SDI_MASK0: [14] **MASK14:** Mask TXFIFOBW flag

SDI_MASK0: [13] **MASK13:** Mask RXACT flag

SDI_MASK0: [12] **MASK12:** Mask TXACT flag

SDI_MASK0: [11] **MASK11:** Mask CMDACT flag

SDI_MASK0: [10] **MASK10:** Mask DBCKEND flag

SDI_MASK0: [9] **MASK9:** Mask STBITERR flag

SDI_MASK0: [8] **MASK8:** Mask DATAEND flag

SDI_MASK0: [7] **MASK7:** Mask CMDSENT flag

SDI_MASK0: [6] **MASK6:** Mask CMDREND flag

SDI_MASK0: [5] **MASK5:** Mask RXOVERR flag

SDI_MASK0: [4] **MASK4:** Mask TXUNDERR flag

SDI_MASK0: [3] **MASK3:** Mask DTIMEOUT flag

SDI_MASK0: [2] **MASK2:** Mask CTIMEOUT flag

SDI_MASK0: [1] **MASK1:** Mask DCRCFAIL flag

SDI_MASK0: [0] **MASK0:** Mask CCRCFAIL flag

SDI_FIFOCNT SDI FIFO Counter Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DATACOUNT																							
R								R																							

Address: SDIBaseAddress + 0x48

Type: R

Reset: 0x00000000

Description: The SDI_FIFOCNT register contains the remaining number of words to be written to or read from the FIFO. The FIFO counter loads the value from the data length register (see section 2758: SDI Data Length Register SDI_DLEN on page15) when the Data Transfer Enable bit, DTEN, is set in the Data Control register (SDIDCTRL register). If the data length is not word aligned (multiple of 4), the remaining 1 to 3 bytes are regarded as a word. Table597 shows the bit assignment of the SDI_FIFOCNT register.

[23:0] **DATACOUNT:** Remaining number of words to be written to or read from the FIFO.

SDI_FIFO SDI Data FIFO Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFO_DATA																															
R/W																															

Address: SDIBaseAddress + 0x80

Type: R/W

Reset: 0x00000000

Description: The FIFO contains 32 words. The same FIFO is used in receive and transmit.The receive and transmit FIFOs can be read or written through a 32-bit wide register.This register can be accessed by 16 sequential addresses. This allows the CPU to use its load and store multiple operands to read/write to the FIFO (burst). This doesn't allow to access to a specific word in the FIFO.Table598 shows the bit assignment of the SDI_FIFO register.

[31:0] **FIFO_DATA:** Receive and Transmit FIFO data. The FIFO data register can be seen by 16 entries of 32-bit words, from address SDI base + 0x080 to SDI base + 0xBC.

SDIPeriphID0 SDI Peripheral Identification Register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PartNumber0							
R																								R							

Address: SDIBaseAddress + 0xFE0

Type: R

Reset: 0x00000080

Description: The SDIPeriphID0-3 registers are four 8-bit registers, that span address location 0xFE0 to 0xFEC. The registers are read-only.

[7:0] **PartNumber0:** These bits read back as 0x80

SDIPeriphID1 SDI Peripheral Identification Register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								Designer0				PartNumber1			
R																								R				R			

Address: SDIBaseAddress + 0xFE4

Type: R

Reset: 0x00000001

Description: SDI Peripheral Identification Register 1

[7:4] **Designer0:** These bits read back as 0x0

[3:0] **PartNumber1:** These bits read back as 0x1

SDIPeriphID2 SDI Peripheral Identification Register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								Revision				Designer1			
R																								R				R			

Address: SDIBaseAddress + 0xFE8

Type: R

Reset: 0x00000028

Description: .

[7:4] **Revision:** These bits read back as 0x2

[3:0] **Designer1:** These bits read back as 0x8

SDIPeriphID3 SDI Peripheral Identification Register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								Configuration							
R																								R							

Address: SDIBaseAddress + 0xFEC

Type: R

Reset: 0x00000000

Description: SDI Peripheral Identification Register 3

[7:0] **Configuration:** These bits read back as 0x00

SDIPCellID0 SDI PCell Identification Register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SDIPCellID0							
R																								R							

Address: SDIBaseAddress + 0xFF0

Type: R

Reset: 0x0000000D

Description: The SDIPCellID0-3 registers are four 8-bit registers, that span address location 0xFF0 to 0xFFC. The registers are read-only.

[7:0] **SDIPCellID0:** These bits read back as 0x0D

SDIPCellID1 SDI PCell Identification Register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SDIPCellID1							
R																								R							

Address: SDIBaseAddress + 0xFF4

Type: R

Reset: 0x000000F0

Description: SDI PCell Identification Register 1

[7:0] **SDIPCellID1:** These bits read back as 0xF0

SDIPCellID2 SDI PCell Identification Register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SDIPCellID2							
R																								R							

Address: SDIBaseAddress + 0xFF8

Type: R

Reset: 0x00000005

Description: SDI PCell Identification Register 2

[7:0] **SDIPCellID2:** These bits read back as 0x05

SDIPCellID3 SDI PCell Identification Register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SDIPCellID3							
R																								R							

Address: SDIBaseAddress + 0xFFC

Type: R

Reset: 0x000000B1

Description: SDI PCell Identification Register 3

[7:0] **SDIPCellID3:** These bits read back as 0xB1

28 Scroll key and keypad encoder (SKE)

28.1 SKE register addressing

Register addresses are provided as the SKE base address, SKEBaseAddress, plus the register offset.

The SKEBaseAddress is 0x101E B000.

28.2 SKE register summary

The device communicates to the system via registers accessible via a 32-bit width AMBA rev. 2.0 Peripheral Bus (APB).

Table 61. SKE register list

Offset	Register	Description	Page
0x000	SKE_CR	Control register	373
0x004	SKE_VAL0	Scroll key 0 value register	374
0x008	SKE_VAL1	Scroll key 1 value register	374
0x00C	SKE_DBCR	Scroll key and keypad debounce register	375
0x010	SKE_IMSC	Interrupt mask set and clear register	376
0x014	SKE_RIS	Raw interrupt status register	376
0x018	SKE_MIS	Masked interrupt status register	377
0x01C	SKE_ICR	Interrupt clear register	378
0x020	SKE_ASR0	Keypad autoscan result register 0	378
0x024	SKE_ASR1	Keypad autoscan result register 1	379
0x028	SKE_ASR2	Keypad autoscan result register 2	379
0x02C	SKE_ASR3	Keypad autoscan result register 3	380
0xFE0	SKE_PeriphID0	Peripheral identification register (bits 7:0)	380
0xFE4	SKE_PeriphID1	Peripheral identification register (bits 15:8)	380
0xFE8	SKE_PeriphID2	Peripheral identification register (bits 23:16)	381
0xFEC	SKE_PeriphID3	Peripheral identification register (bits 31:24)	381
0xFF0	SKE_PCellID0	PCell identification register (bits 7:0)	381
0xFF4	SKE_PCellID1	PCell identification register (bits 15:8)	382
0xFF8	SKE_PCellID2	PCell identification register (bits 23:16)	382
0xFFC	SKE_PCellID3	PCell identification register (bits 31:24)	382

28.3 SKE register descriptions

SKE_CR SKE control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KPOC7	KPOC6	KPOC5	KPOC4	KPOC3	KPOC2	KPOC1	KPOC0	KPASON	KPMLT	KPCN			KPASEN	SKEN1	SKEN0
RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW			RW	RW	RW

Address: SKEBaseAddress + 0x000

Reset: 0x0000 0000

Description: Contains the enable bit and the debounce interval value.

KPOC[0:7] Keypad output KPCx control. Drives the matrix keypad output signal, KPOx. The KPOCx bits for used columns must be set to 1 to detect the key usage that starts the autoscan sequence or software sequence.

0: KPCx output signal is in HiZ. 1: KPCx output signal is driven low.

KPASON Keypad autoscan on-going flag. Reveals that keypad autoscan process is on going.

0: scan of matrix is completed or key pas autoscan is disabled. Registers *SKE_ASRx* can be read.

1: scan of matrix is on going, SKE_ASRx registers are not valid.

KPMLT Keypad multi key pressed detection enable. Defines if autoscan waits for all keys to be released between matrix scan (to ignore multiple key presses).

0: autoscan waits for all keys to be released before starting a new matrix scan.

1: autoscan restart without waiting until all keys are released.

KPCN Keypad column number. Defines the number of matrix keypad columns to be scanned by the autoscan process. Autoscan process scans asserted low output KPC0 first, and continues up to KPCx, with x = (KPCN -1):

000: autoscan asserts low KPC[0] only (1 column),

001: autoscan asserts low sequentially KPC[0] and KPC[1] (2 columns) and so on until

111: autoscan asserts low sequentially KPC[0], then KPC[1], and so on until KPC[7] (8 columns).

KPASEN Keypad autoscan enable. A read returns the value of the KPASEN bit.

0: all SKE activity is stopped to reduce power consumption, interrupts are cleared, registers SKE_ASRx remain unchanged. When KPASEN is cleared, the on-going key-pad autoscan is completed. KPASON must be cleared before the keypad autoscan is re-enabled.

1: keypad autoscan is enabled.

SKEN1 Scroll key encoder 1 enable.

0: all activity in scroll key encoder 1 is stopped to reduce power consumption, interrupts are cleared, *SKE_VAL1* remains unchanged. A read returns the value of the SKEN1 bit.

1: scroll key encoder 1 is enabled.

SKEN0 Scroll key encoder 0 enable.

0: all activity in scroll key encoder 0 is stopped to reduce power consumption, interrupts are cleared, *SKE_VAL0* remains unchanged. A read returns the value of the SKEN0 bit.

1: scroll key encoder 0 is enabled.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SKE_VAL0 SKE value register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVF0	UDF0	0	0	0	0	0	0	SKEVAL0							
RW	RW	R	R	R	R	R	R	RW							

Address: SKEBaseAddress + 0x004

Reset: 0x0000 0000

Description: Contains overflow and underflow flags and the counter value for scroll key 0.

OVF0 Scroll key 0 overflow flag. Cleared when SKE_VAL0 register is read or written.
When SKEVAL0 is decrementing from 255 to 0, and if UDF0 flag is cleared, then OVF0 flag is set;
When SKEVAL0 is incrementing from 0 to 255, OVF0 is clear.

UDF0 Scroll key 0 underflow flag. Cleared when SKE_VAL0 register is read or written.
When SKEVAL0 is incrementing from 0 to 255, and if OVF0 flag is cleared, then UDF0 flag is set;
When SKEVAL0 is decrementing from 255 to 0, UDF0 is clear.

SKEVAL0 Scroll key 0 count value. Count value between 0 and 255 for scroll key 0. Every PCLK clock cycle, this counter value is incremented, decremented or unchanged, depending on the SKA0/SKB0 level change detected after the debouncing logic. An interrupt occurs on each increment/decrement of this counter value (no interrupt when it is changed by software).
When written, the underflow/overflow flags are cleared.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SKE_VAL1 SKE value register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVF1	UDF1	0	0	0	0	0	0	SKEVAL1							
RW	RW	R	R	R	R	R	R	RW							

Address: SKEBaseAddress + 0x008

Reset: 0x0000 0000

Description: Contains overflow and underflow flags and the counter value for scroll key 1.

OVF1 Scroll key 1 overflow flag. Cleared when SKE_VAL1 register is read or written.
When SKEVAL1 is decrementing from 255 to 0, and if UDF1 flag is clear, then OVF1 flag is set;
When SKEVAL1 is incrementing from 0 to 255, OVF1 is clear.

UDF1 Scroll key 1 underflow flag. Cleared when SKE_VAL1 register is read or written.
When SKEVAL1 is incrementing from 0 to 255, and if OVF1 flag is clear, then UDF1 flag is set;
When SKEVAL1 is decrementing from 255 to 0, UDF1 is clear.

SKEVAL1 Scroll key 1 count value. Count value between 0 and 255 for scroll key 1. Every CLK32K clock cycle, this counter value is incremented, decremented or unchanged, depending on the SKA1/SKB1 level change detected after the debouncing logic. An interrupt occurs on each increment/decrement of this counter value (no interrupt when it is changed by software).
When written, the underflow/overflow flags are cleared.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SKE_DBCR SKE debounce register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KPDBC								SKEDBC							
RW								RW							

Address: SKEBaseAddress + 0x00C

Reset: 0x0000 6666

Description: Contains the debounce interval values for the scroll key encoder and the keypad keypress detection.

KPDBC Keypad debounce interval. Defines the debounce interval for both scroll keys in 32/32.768 ms steps (almost 1 ms).
0x00: by-passed 0x01: 1 x (32/32.768) ms (almost 1 ms)
0x66: 102 x (32/32.768) ms (almost 100 ms) 0xFF: 255 x (32/32.768) ms (almost 250 ms)

SKEDBC Scroll key debounce interval. Defines the debounce interval for both scroll keys in 32/32.768 ms steps (almost 1 ms).
0x00: by-passed 0x01: 1 x (32/32.768) ms (almost 1 ms)
0x66: 102 x (32/32.768) ms (almost 100 ms) 0xFF: 255 x (32/32.768) ms (almost 250 ms)

0 Reserved for future use. Reading returns 0. Must be written with 0.

SKE_IMSC SKE interrupt mask register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	KPIMS	KPIMA	SKEIM1	SKEIM0
R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Address: SKEBaseAddress + 0x010

Reset: 0x0000 0000

Description: Controls the masking of the interrupt generated by the SKE. Reading this register returns the current value of the mask on the SKE interrupt. This register is cleared upon a power-on reset only (PORn).

KPIMS Keypad interrupt enable. Enables generation of interrupt to VIC for matrix keypad in software scan mode (*SKE_CR*.KPASEN=0).

KPIMA Keypad interrupt enable. Enables generation of interrupt to VIC for matrix keypad in autoscan mode (SKE_CR.KPASEN=0).

SKEIM1 Scroll key 1 interrupt enable. Enables generation of interrupt to VIC for scroll key 1. Any change of (debounced) SKA1/SKB1 inputs causing a change in register *SKE_VAL0/1* triggers an interrupt to the VIC when this bit is set to 1.

SKEIM0 Scroll key 0 interrupt enable. Enables generation of interrupt to VIC for scroll key 0. Any change of (debounced) SKA0/SKB0 inputs causing a change in register SKE_VAL triggers an interrupt to the VIC when this bit is set to 1.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SKE_RIS SKE raw interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	KPRISS	KPRISA	SKERIS1	SKERIS0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: SKEBaseAddress + 0x014

Reset: 0x0000 0000

Description: Returns the current raw status value of the corresponding interrupt prior to masking. A write has no effect.

KPRISS Keypad raw interrupt status software mode. The raw interrupt state (prior to masking) of matrix keypad in software mode (*SKE_CR*.KPASEN=0). This bit is set to 1 as long as keypad activity is detected (after debouncing). Keypad interrupt clear bit *SKE_ICR*.KPIC does not clear this bit. KPRISS and KPRISA interrupts are mutually exclusive

KPRISA Keypad raw interrupt status autoscan mode. The raw interrupt state (prior to masking) of matrix keypad in software mode (SKE_CR.KPASEN=0). This bit is set to 1 by each keyboard scan cycle which ends with a stable key pad.This bit remains set as long as keypad activity is detected. Keypad interrupt clear bit SKE_ICR.KPIC does not clear this bit.

SKERIS1 Scroll key 1 raw interrupt status. The raw interrupt state (prior to masking) of scroll key 1 interrupt.

SKERIS0 Scroll key 0 raw interrupt status. The raw interrupt state (prior to masking) of scroll key 0 interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SKE_MIS SKE masked interrupt status register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	KPMISS	KPMISA	SKEMIS1	SKEMIS0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Address: SKEBaseAddress + 0x018

Reset: 0x0000 0000

Description: Returns the current masked status value (after masking) of the corresponding interrupt. A write has no effect.

KPMISS Keypad masked interrupt status software mode. The masked interrupt state of matrix keypad in software mode (*SKE_CR*.KPASEN=0). Clearing the keypad interrupt with *SKE_ICR*.KPICS clears KPMISS bit, while KPRISS can remain set as long as a key is pressed.

KPMISA Keypad masked interrupt status autoscan mode. The masked interrupt state of matrix keypad in autoscan mode (SKE_CR.KPASEN=0). Clearing the keypad interrupt with SKE_ICR.KPICA clears KPMISA bit, while KPRISA can remain set as long as a key is pressed.

SKEMIS1 Scroll key 1 masked interrupt status. The masked interrupt status of scroll key 1 interrupt.

SKEMIS0 Scroll key 0 masked interrupt status. The masked interrupt status of scroll key 0 interrupt.

0 Reserved for future use. Reading returns 0. Must be written with 0.

SKE_ICR SKE interrupt clear register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	KPICS	KPICA	SKEIC1	SKEIC0
R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Address: SKEBaseAddress + 0x01C

Reset: 0x0000 0000

Description: Writing a bit with 1 clears the corresponding interrupt. Writing with 0 has no effect.

KPICS Keypad interrupt clear. Clears the masked interrupt state of matrix keypad in software mode (*SKE_MIS*.KPMIS), but does not clear KPRIS bit that remains set as long as a key is pressed. Reading returns 0.
Writing 0: no effect 1: clears the interrupt

KPICA Keypad interrupt clear. Clears the masked interrupt state of matrix keypad in autoscan mode (SKE_MIS.KPMIS), but does not clear KPRIS bit that remains set as long as a key is pressed. Reading returns 0.
Writing 0: no effect 1: clears the interrupt

SKEIC1 Scroll key 1 interrupt clear clears the scroll key 1 interrupt. Reading returns 0.
Writing 0: no effect 1: clears the interrupt

SKEIC0 Scroll key 0 interrupt clear clears the scroll key 0 interrupt. Reading returns 0.
Writing 0: no effect 1: clears the interrupt

0 Reserved for future use. Reading returns 0. Must be written with 0.

SKE_ASR0 SKE autoscan result register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KPRVAL1								KPRVAL0							
RW								RW							

Address: SKEBaseAddress + 0x020

Reset: 0x0000 0000

Description: SKE_ASR[0:3] registers contain the row values captured from the autoscan process. Each register *SKE_ASRx* contains two row values, first one (lower byte), is the row

value for column (2*x), the second one (upper byte) is the row value for column (2*x+1).

KPRVAL[0:7] Autoscan keypad row defines the inverted levels captured on KPI[7:0] signals when matrix keypad column y (y = 0 to 7), that is, KPO[y] signal, was asserted low during the autoscan process.
The LSB correspond to inverted level on KPI[0], the MSB correspond to inverted level on KPI[7].

0 Reserved for future use. Reading returns 0. Must be written with 0.

SKE_ASR1 SKE autoscan result register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KPRVAL3								KPRVAL2							
RW								RW							

Address: SKEBaseAddress + 0x024

Reset: 0x0000 0000

Description: See register *SKE_ASR0* description.

SKE_ASR2 SKE autoscan result register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KPRVAL5								KPRVAL4							
RW								RW							

Address: SKEBaseAddress + 0x028

Reset: 0x0000 0000

Description: See register *SKE_ASR0* description.

SKE_ASR3 SKE autoscan result register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KPRVAL7								KPRVAL6							
RW								RW							

Address: SKEBaseAddress + 0x02C

Reset: 0x0000 0000

Description: See register *SKE_ASR0* description.

SKE_PeriphID0 SKE peripheral identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PartNumber0							
R	R	R	R	R	R	R	R	R							

Address: SKEBaseAddress + 0xFE0

Reset: 0x0000 0070

Description: PartNumber0 returns 0x70.

SKE_PeriphID1 SKE peripheral identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Designer0				PartNumber1			
R	R	R	R	R	R	R	R	R				R			

Address: SKEBaseAddress + 0xFE4

Reset: 0x0000 0000

Description: Designer0 returns 0x00. PartNumber1 returns 0x00.

SKE_PeriphID2 SKE peripheral identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Revision				Designer1			
R	R	R	R	R	R	R	R	R				R			

Address: SKEBaseAddress + 0xFE8

Reset: 0x0000 0018

Description: Revision returns 0x01. Designer1 returns 0x08.

SKE_PeriphID3 SKE peripheral identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	Configuration							
R	R	R	R	R	R	R	R	R							

Address: STn8815 Base + 0xFEC

Reset: 0x0000 0000

Description: Configuration returns 0x00.

SKE_PCellID0 SKE PCell identification register 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SKEPCellID0							
R	R	R	R	R	R	R	R	R							

Address: SKEBaseAddress + 0xFF0

Reset: 0x0000 000D

Description: SKEPCellID0 returns 0x0D.

SKE_PCellID1 SKE PCell identification register 1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SKEPCellID1							
R	R	R	R	R	R	R	R	R							

Address: SKEBaseAddress + 0xFF4

Reset: 0x0000 00F0

Description: SKEPCellID1 returns 0xF0.

SKE_PCellID2 SKE PCell identification register 2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SKEPCellID2							
R	R	R	R	R	R	R	R	R							

Address: SKEBaseAddress + 0xFF8

Reset: 0x0000 0005

Description: SKEPCellID2 returns 0x05.

SKE_PCellID3 SKE PCell identification register 3

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SKEPCellID3							
R	R	R	R	R	R	R	R	R							

Address: SKEBaseAddress + 0xFFC

Reset: 0x0000 00B1

SKEPCellID3 returns 0xB1.

29 Revision history

Table 62. Document revision history

Date	Revision	Changes
12-Oct-2009	1	Initial release.