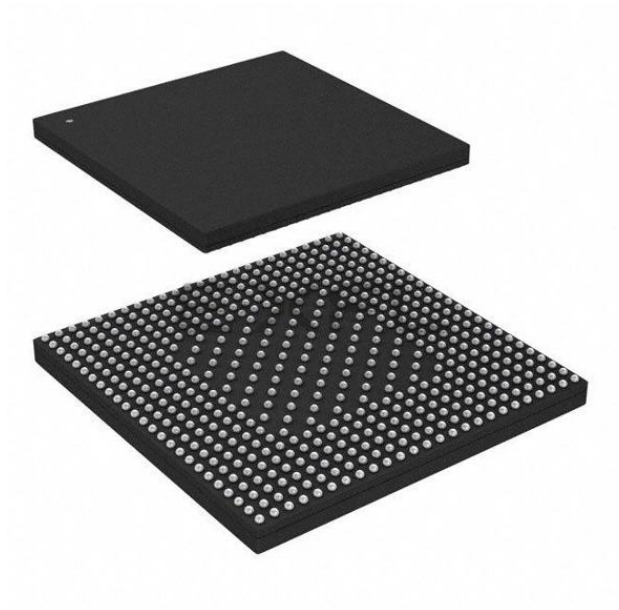


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Active
Core Processor	ARM® Cortex®-A7, ARM® Cortex®-M4
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	800MHz
Co-Processors/DSP	Multimedia; NEON™ MPE
RAM Controllers	LPDDR2, LPDDR3, DDR3, DDR3L
Graphics Acceleration	No
Display & Interface Controllers	Keypad, LCD, MIPI
Ethernet	10/100/1000Mbps (1)
SATA	-
USB	USB 2.0 + PHY (1), USB 2.0 OTG + PHY (1)
Voltage - I/O	1.8V, 3.3V
Operating Temperature	0°C ~ 95°C (TJ)
Security Features	A-HAB, ARM TZ, CAAM, CSU, SJC, SNVS
Package / Case	488-TFBGA
Supplier Device Package	488-TFBGA (12x12)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mcimx7s3dvk08sc

RDC_SEMAPHOREx_RSTGT_R field descriptions (continued)

Field	Description
	If RSTGTN < 64, then reset the single gate defined by RSTGTN, else reset all the gates.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 RSTGSM	Reset Gate Finite State Machine. Reads of the RDC_SEMA42RSTGT register return the encoded state machine value. Note the RSTGSM = 10 state is valid for only a single machine cycle, so it is impossible for a read to return this value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as: 00 Idle, waiting for the first data pattern write. 01 Waiting for the second data pattern write. 10 The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. The "01" state persists for only one clock cycle. Software will never be able to observe this state. 11 This state encoding is never used and therefore reserved.
RSTGMS	Reset Gate Bus Master. This 4-bit read-only field records the logical number of the bus master performing the gate reset function. The reset function requires that the two consecutive writes to this register must be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device, and the logical processor number is SoC-specific. Consult the device reference manual for this information.

Table 5-11. Clock Root Table (continued)

Offset	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			100 - DDR_PLL_DIV2 101 - AUDIO_PLL 110 - VIDEO_PLL 111 - SYS_PLL_PFD3
0xA980	EIM_CLK_ROOT	135	000 - OSC_24M 001 - SYS_PLL_PFD2_DIV2 010 - SYS_PLL_DIV4 011 - DDR_PLL_DIV2 100 - SYS_PLL_PFD2 101 - SYS_PLL_PFD3 110 - ENET_PLL_DIV8 111 - USB_PLL
0xAA00	NAND_CLK_ROOT	533	000 - OSC_24M 001 - SYS_PLL 010 - DDR_PLL_DIV2 011 - SYS_PLL_PFD0 100 - SYS_PLL_PFD3 101 - ENET_PLL_DIV2 110 - ENET_PLL_DIV4 111 - VIDEO_PLL
0xAA80	QSPI_CLK_ROOT	400	000 - OSC_24M 001 - SYS_PLL_PFD4 010 - DDR_PLL_DIV2 011 - ENET_PLL_DIV2 100 - SYS_PLL_PFD3 101 - SYS_PLL_PFD2 110 - SYS_PLL_PFD6 111 - SYS_PLL_PFD7
0xAB00	USDHC1_CLK_ROOT	200	000 - OSC_24M 001 - SYS_PLL_PFD0 010 - DDR_PLL_DIV2 011 - ENET_PLL_DIV2 100 - SYS_PLL_PFD4 101 - SYS_PLL_PFD2 110 - SYS_PLL_PFD6 111 - SYS_PLL_PFD7
0xAB80	USDHC2_CLK_ROOT	200	000 - OSC_24M

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_81A8	Post Divider Register (CCM_POST_ROOT3_CLR)	32	R/W	0000_0000h	5.2.8.20/ 727
3038_81AC	Post Divider Register (CCM_POST_ROOT3_TOG)	32	R/W	0000_0000h	5.2.8.21/ 730
3038_81B0	Pre Divider Register (CCM_PRE3)	32	R/W	0000_0000h	5.2.8.22/ 733
3038_81B4	Pre Divider Register (CCM_PRE_ROOT3_SET)	32	R/W	0000_0000h	5.2.8.23/ 736
3038_81B8	Pre Divider Register (CCM_PRE_ROOT3_CLR)	32	R/W	0000_0000h	5.2.8.24/ 739
3038_81BC	Pre Divider Register (CCM_PRE_ROOT3_TOG)	32	R/W	0000_0000h	5.2.8.25/ 742
3038_81F0	Access Control Register (CCM_ACCESS_CTRL3)	32	R/W	0000_0000h	5.2.8.26/ 745
3038_81F4	Access Control Register (CCM_ACCESS_CTRL_ROOT3_SET)	32	R/W	0000_0000h	5.2.8.27/ 747
3038_81F8	Access Control Register (CCM_ACCESS_CTRL_ROOT3_CLR)	32	R/W	0000_0000h	5.2.8.28/ 750
3038_81FC	Access Control Register (CCM_ACCESS_CTRL_ROOT3_TOG)	32	R/W	0000_0000h	5.2.8.29/ 752
3038_8200	Target Register (CCM_TARGET_ROOT4)	32	R/W	0000_0000h	5.2.8.10/ 709
3038_8204	Target Register (CCM_TARGET_ROOT4_SET)	32	R/W	0000_0000h	5.2.8.11/ 711
3038_8208	Target Register (CCM_TARGET_ROOT4_CLR)	32	R/W	0000_0000h	5.2.8.12/ 713
3038_820C	Target Register (CCM_TARGET_ROOT4_TOG)	32	R/W	0000_0000h	5.2.8.13/ 715
3038_8210	Miscellaneous Register (CCM_MISC4)	32	R/W	0000_0000h	5.2.8.14/ 717
3038_8214	Miscellaneous Register (CCM_MISC_ROOT4_SET)	32	R/W	0000_0000h	5.2.8.15/ 718
3038_8218	Miscellaneous Register (CCM_MISC_ROOT4_CLR)	32	R/W	0000_0000h	5.2.8.16/ 719
3038_821C	Miscellaneous Register (CCM_MISC_ROOT4_TOG)	32	R/W	0000_0000h	5.2.8.17/ 720
3038_8220	Post Divider Register (CCM_POST4)	32	R/W	0000_0000h	5.2.8.18/ 721
3038_8224	Post Divider Register (CCM_POST_ROOT4_SET)	32	R/W	0000_0000h	5.2.8.19/ 724
3038_8228	Post Divider Register (CCM_POST_ROOT4_CLR)	32	R/W	0000_0000h	5.2.8.20/ 727
3038_822C	Post Divider Register (CCM_POST_ROOT4_TOG)	32	R/W	0000_0000h	5.2.8.21/ 730

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A608	Target Register (CCM_TARGET_ROOT76_CLR)	32	R/W	0000_0000h	5.2.8.12/ 713
3038_A60C	Target Register (CCM_TARGET_ROOT76_TOG)	32	R/W	0000_0000h	5.2.8.13/ 715
3038_A610	Miscellaneous Register (CCM_MISC76)	32	R/W	0000_0000h	5.2.8.14/ 717
3038_A614	Miscellaneous Register (CCM_MISC_ROOT76_SET)	32	R/W	0000_0000h	5.2.8.15/ 718
3038_A618	Miscellaneous Register (CCM_MISC_ROOT76_CLR)	32	R/W	0000_0000h	5.2.8.16/ 719
3038_A61C	Miscellaneous Register (CCM_MISC_ROOT76_TOG)	32	R/W	0000_0000h	5.2.8.17/ 720
3038_A620	Post Divider Register (CCM_POST76)	32	R/W	0000_0000h	5.2.8.18/ 721
3038_A624	Post Divider Register (CCM_POST_ROOT76_SET)	32	R/W	0000_0000h	5.2.8.19/ 724
3038_A628	Post Divider Register (CCM_POST_ROOT76_CLR)	32	R/W	0000_0000h	5.2.8.20/ 727
3038_A62C	Post Divider Register (CCM_POST_ROOT76_TOG)	32	R/W	0000_0000h	5.2.8.21/ 730
3038_A630	Pre Divider Register (CCM_PRE76)	32	R/W	0000_0000h	5.2.8.22/ 733
3038_A634	Pre Divider Register (CCM_PRE_ROOT76_SET)	32	R/W	0000_0000h	5.2.8.23/ 736
3038_A638	Pre Divider Register (CCM_PRE_ROOT76_CLR)	32	R/W	0000_0000h	5.2.8.24/ 739
3038_A63C	Pre Divider Register (CCM_PRE_ROOT76_TOG)	32	R/W	0000_0000h	5.2.8.25/ 742
3038_A670	Access Control Register (CCM_ACCESS_CTRL76)	32	R/W	0000_0000h	5.2.8.26/ 745
3038_A674	Access Control Register (CCM_ACCESS_CTRL_ROOT76_SET)	32	R/W	0000_0000h	5.2.8.27/ 747
3038_A678	Access Control Register (CCM_ACCESS_CTRL_ROOT76_CLR)	32	R/W	0000_0000h	5.2.8.28/ 750
3038_A67C	Access Control Register (CCM_ACCESS_CTRL_ROOT76_TOG)	32	R/W	0000_0000h	5.2.8.29/ 752
3038_A680	Target Register (CCM_TARGET_ROOT77)	32	R/W	0000_0000h	5.2.8.10/ 709
3038_A684	Target Register (CCM_TARGET_ROOT77_SET)	32	R/W	0000_0000h	5.2.8.11/ 711
3038_A688	Target Register (CCM_TARGET_ROOT77_CLR)	32	R/W	0000_0000h	5.2.8.12/ 713
3038_A68C	Target Register (CCM_TARGET_ROOT77_TOG)	32	R/W	0000_0000h	5.2.8.13/ 715

Table continues on the next page...

SRC_SISR field descriptions (continued)

Field	Description
3 OTGPHY2_ PASSED_ RESET	Interrupt generated to indicate that OTG PHY2 passed software reset and is ready to be used 0 Interrupt generated not due to OTG PHY2 passed reset 1 Interrupt generated due to OTG PHY2 passed reset
2 OTGPHY1_ PASSED_ RESET	Interrupt generated to indicate that OTG PHY1 passed software reset and is ready to be used 0 Interrupt generated not due to OTG PHY1 passed reset 1 Interrupt generated due to OTG PHY1 passed reset
1 HSICPHY_ PASSED_ RESET	Interrupt generated to indicate that HSIC PHY passed software reset and is ready to be used 0 Interrupt generated not due to HSIC PHY passed reset 1 Interrupt generated due to HSIC PHY passed reset
0 -	This field is reserved. Reserved

Table 7-43. Real-Time Debug Output Pins (continued)

Pin	Description
	1 Run mode
debug_event_channel_sel	Indicates if debug_event_channel displays current channel or last received event 0- debug_event_channel[5:0] gives the number of the current channel 1- debug_event_channel[5:0] gives the number of the last received event
debug_event_channel[5:0]	Gives the number of any DMA request as soon as it is received or the number of the current channel. The value of debug_event_channel_sel indicates if debug_event_channel displays the current channel or last received event. The signal debug_event_channel_sel must be observed to determine what information is provided on debug_event_chanel at any given time.
debug_pc[13:0]	Program Counter value; it has a meaning when the core is in run mode.
debug_mode	Set when the core is in debug. 0 - 1 Core is in debug
debug_bus_error	Set when an error was received during a load or a store (ld, st, ldf, or stf instruction) and registered in SF or DF flag. 0 No error during last load/store 1 Error during last load/store
debug_bus_device[4:0]	Indicates the device or functional unit that is accessed by the current instruction. The debug_bus_device output is always valid when in sleep mode, debug mode, or executing any instruction that does not access the functional units or the memory mapped devices, "no access" is output. 0 No access 1 MSA 2 MDA 3 MD 4 MS 5 PSA 6 PDA 7 PD 8 PS 9 RESERVED 10 RESERVED 11 RESERVED 12 RESERVED 13 CA 14 CS 15 Reserved 16 Memory (RAM or ROM) 17 Memory mapped register

Table continues on the next page...

Instance	Port	Pad	Mode
	SAI1_RX_SYNC	ENET1_TX_CTL	ALT2
		SAI1_RXFS	ALT0
	SAI1_TX_BCLK	ENET1_RX_CLK	ALT2
		SAI1_TXC	ALT0
	SAI1_TX_DATA	ENET1_COL	ALT2
		SAI1_TXD	ALT0
	SAI1_TX_SYNC	ENET1_CRS	ALT2
		SAI1_TXFS	ALT0
SAI2	SAI2_MCLK	GPIO1_IO02	ALT3
		SAI1_MCLK	ALT2
		SD2_RESET_B	ALT1
	SAI2_RX_BCLK	SAI1_RXC	ALT2
		SD2_CMD	ALT1
	SAI2_RX_DATA	SAI2_RXD	ALT0
		SD2_DATA0	ALT1
	SAI2_RX_SYNC	SAI1_RXFS	ALT2
		SD2_CLK	ALT1
	SAI2_TX_BCLK	SAI2_TXC	ALT0
		SD2_DATA1	ALT1
	SAI2_TX_DATA	SAI2_TXD	ALT0
		SD2_DATA3	ALT1
	SAI2_TX_SYNC	SAI2_TXFS	ALT0
		SD2_DATA2	ALT1
SAI3	SAI3_MCLK	GPIO1_IO03	ALT3
		SD1_RESET_B	ALT1
		SD3_RESET_B	ALT3
		UART1_TXD	ALT2
	SAI3_RX_BCLK	SD1_CMD	ALT1
		SD3_CMD	ALT3
		UART2_RXD	ALT2
	SAI3_RX_DATA	SD1_DATA0	ALT1
		SD3_DATA0	ALT3
		UART2_TXD	ALT2
	SAI3_RX_SYNC	SD1_CLK	ALT1
		SD3_CLK	ALT3
		UART3_RXD	ALT2
	SAI3_TX_BCLK	SD1_DATA1	ALT1
		SD3_DATA1	ALT3
		UART3_TXD	ALT2
	SAI3_TX_DATA	SD1_DATA3	ALT1
		SD3_DATA3	ALT3

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_CMD field descriptions

Field	Description
31–7 -	This field is reserved. Reserved
6–5 PS	Pull Select Field Select one out of next values for pad: SD2_CMD 00 **PS_0_100K_PD** — 100K PD 01 **PS_1_5K_PU** — 5K PU 10 **PS_2_47K_PU** — 47K PU 11 **PS_3_100K_PU** — 100K PU
4 PE	Pull Enable Field Select one out of next values for pad: SD2_CMD 0 **PE_0_Pull_Disabled** — Pull Disabled 1 **PE_1_Pull_Enabled** — Pull Enabled
3 HYS	Hyst. Enable Field Select one out of next values for pad: SD2_CMD 0 **HYS_0_Hysteresis_Disabled** — Hysteresis Disabled 1 **HYS_1_Hysteresis_Enabled** — Hysteresis Enabled
2 SRE	Slew Rate Field Select one out of next values for pad: SD2_CMD 0 **SRE_0_Fast_Slew_Rate** — Fast Slew Rate 1 **SRE_1_Slow_Slew_Rate** — Slow Slew Rate
DSE	Drive Strength Field Select one out of next values for pad: SD2_CMD 00 **DSE_0_X1** — X1 01 **DSE_1_X4** — X4 10 **DSE_2_X2** — X2 11 **DSE_3_X6** — X6

Table 9-19. Settings for 4K+218 FLASH (continued)

Setting	Value
DATAN_SIZE	512=0x200 (in register interface, assigned as 0x80)
ECCN	14=0x0E
GFN	GF(2^{13})
NBLOCKS	8

9.5.2.3.2 4K+128 flash, 10 bytes metadata, 1024 byte data blocks, separate metadata, assuming GF(2^{13}) for data and GF(2^{14}) for metadata

This flash will have 118 bytes available for ECC (after subtracting the metadata size), therefore, 994 bits.

Dividing by 4*14 (number of blocks * ECC level) we get 17.75, therefore we can support ECC16 on the data blocks. The number of free spare bits becomes 944 - 16 * 4 * 14 = 944 - 896 = 48, divided by 13 = 3.69, therefore the metadata can be also covered by ECC2.

Table 9-20. Settings for 4K+128 FLASH

Setting	Value
PAGE_SIZE	4096+128=4224=0x1080
META_SIZE	10=0x0A
DATA0_SIZE	0
ECC0	2
GF0	GF(2^{13})
DATAN_SIZE	1024=0x400 (in register interface, assigned as 0x100)
ECCN	16
GFN	GF(2^{14})
NBLOCKS	4

In this case, there will be additional unused spare bits, with the BCH will pad out with zeros.

9.5.2.4 Data Buffers in System Memory

While the data on the flash is interleaved with parity symbols, the BCH assumes that the data buffers in memory are contiguous.

BCH_CTRL*n* field descriptions

Field	Description
31 SFTRST	Set this bit to 0 to enable normal BCH operation. Set this bit to 1 (default) to disable clocking with the BCH and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the BCH block to its default state. This bit resets all state machines except for the AHB master state machine 0x0 **RUN** — Allow BCH to operate normally. 0x1 **RESET** — Hold BCH in reset.
30 CLKGATE	This bit must be set to 0 for normal operation. When set to 1 it gates off the clocks to the block. 0x0 **RUN** — Allow BCH to operate normally. 0x1 **NO_CLKS** — Do not clock BCH gates in order to minimize power consumption.
29–23 RSVD5	This field is reserved. This read-only field is reserved and always has the value 0.
22 DEBUGSYNDROME	(For debug purposes only). Enable write of computed syndromes to memory on BCH decode operations. Computed syndromes will be written to the auxiliary buffer after the status block. Syndromes will be written as padded 16-bit values.
21–20 RSVD4	This field is reserved. This read-only field is reserved and always has the value 0
19–18 M2M_LAYOUT	Selects the flash page format for memory-to-memory operations.
17 M2M_ENCODE	Selects encode (parity generation) or decode (correction) mode for memory-to-memory operations.
16 M2M_ENABLE	NOTE! WRITING THIS BIT INITIATES A MEMORY-TO-MEMORY OPERATION. The BCH module must be inactive (not processing data from the GPMI) when this bit is set. The M2M_ENCODE and M2M_LAYOUT bits as well as the ENCODEPTR, DATAPTR, and METAPTR registers are used for memory-to-memory operations and must be correctly programmed before writing this bit.
15–11 RSVD3	This field is reserved. This read-only field is reserved and always has the value 0
10 DEBUG_STALL_ IRQ_EN	1 = interrupt on debug stall mode is enabled. The IRQ is raised on every block
9 RSVD2	This field is reserved. This read-only field is reserved and always has the value 0.
8 COMPLETE_IRQ_ EN	1 = interrupt on completion of correction is enabled.
7–4 RSVD1	This field is reserved. This read-only field is reserved and always has the value 0.
3 BM_ERROR_IRQ	AHB Bus interface Error Interrupt Status. Write a 1 to the SCT clear address to clear the interrupt status bit.
2 DEBUG_STALL_IRQ	DEBUG STALL Interrupt Status. Write a 1 to the SCT clear address to clear the interrupt status bit.
1 RSVD0	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

9.6.6.5 GPMI Payload Address Register Description (GPMI_PAYLOAD)

The GPMI payload address register specifies the location of the data buffers in system memory. This value must be word aligned.

GPMI_PAYLOAD 0x040

Address: 3300_2000h base + 40h offset = 3300_2040h

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

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS														RSVD0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_PAYLOAD field descriptions

Field	Description
31–2 ADDRESS	Pointer to an array of one or more 512 byte payload buffers.
RSVD0	Always write zeroes to this bit field.

9.6.6.6 GPMI Auxiliary Address Register Description (GPMI_AUXILIARY)

The GPMI auxiliary address register specifies the location of the auxiliary buffers in system memory. This value must be word aligned.

GPMI_AUXILIARY 0x050

Address: 3300_2000h base + 50h offset = 3300_2050h

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

The IP accelerator function can be configured to remove the Ethernet padding bytes that might follow the IP datagram.

On transmit, the MAC automatically adds padding as necessary to fill any frame to a 64-byte length.

11.1.6.8.3 32-bit Ethernet payload alignment

The data FIFOs allow inserting two additional arbitrary bytes in front of a frame. This extends the 14-byte Ethernet header to a 16-byte header, which leads to alignment of the Ethernet payload, following the Ethernet header, on a 32-bit boundary.

This function can be enabled for transmit and receive independently with the corresponding SHIFT16 bits in the ENET*n*_TACC and ENET*n*_RACC registers.

When enabled, the valid frame data is arranged as shown in Table 11-29.

Table 11-29. 64-bit interface data structure with SHIFT16 enabled

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0	Any value	Any value
Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8	Byte 7	Byte 6
...							

11.1.6.8.3.1 Receive processing

When ENET*n*_RACC[SHIFT16] is set, each frame is received with two additional bytes in front of the frame.

The user application must ignore these first two bytes and find the first byte of the frame in bits 23–16 of the first word from the RX FIFO.

Note

SHIFT16 must be set during initialization and kept set during the complete operation, because it influences the FIFO write behavior.

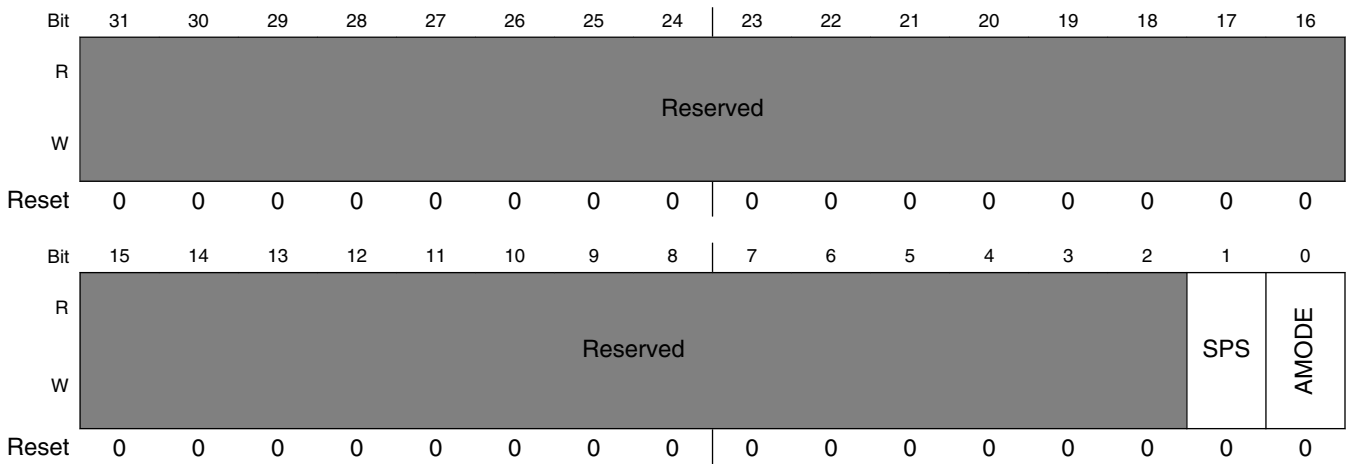
11.1.6.8.3.2 Transmit processing

When ENET*n*_TACC[SHIFT16] is set, the first two bytes of the first word written (bits 15–0) are discarded immediately by the FIFO write logic.

11.2.5.2 SIM Setup Register (SIMx_SETUP)

See the figure below for illustration of valid bits in the SIM Setup Register and the table below for description of the bit fields.

Address: Base address + 4h offset



SIMx_SETUP field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
1 SPS	**SIM card Port Select.** Controls which port the SIM interface uses. **NOTE:** The AMODE bit must be zero when the SPS bit is set to 1. 0 Port 0 Enabled (default) 1 Port 1 Enabled
0 AMODE	**Alternate SIM Card Mode enable.** Enables an alternate SIM module to control SIM card Port 1. **NOTE:** The SPS bit must be 0 to give the alternate SIM module control. 0 Alternate Port Disabled (default) 1 Alternate Port Enabled

11.2.5.3 SIM Port 1 Detect Register (SIMx_PORT1_DETECT)

See the figure below for illustration of valid bits in the SIM Port1 Detect Register and the table below for description of the bit fields.

Summary:

11.4.4.6.3.3 Data Toggle

Data toggle is a mechanism to maintain data coherency between host and device for any given data pipe.

For more information on data toggle, refer to the USB 2.0 specification.

11.4.4.6.3.3.1 Data Toggle Reset

The DCD may reset the data toggle state bit and cause the data toggle sequence to reset in the device controller by writing a '1' to the data toggle reset bit in the USB_UOG_ENDPTCTRLx register.

This should only be necessary when configuring/initializing an endpoint or returning from a STALL condition.

11.4.4.6.3.3.2 Data Toggle Inhibit

NOTE

This feature is for test purposes only and should never be used during normal device controller operation.

Setting the *data toggle Inhibit bit* active ('1') causes the device controller to ignore the data toggle pattern that is normally sent and accept all incoming data packets regardless of the data toggle state.

In normal operation, the device controller checks the DATA0/DATA1 bit against the data toggle to determine if the packet is valid. If Data PID does not match the data toggle state bit maintained by the device controller for that endpoint, the Data toggle is considered not valid. If the data toggle is not valid, the device controller assumes the packet was already received and discards the packet (not reporting it to the DCD). To prevent the host controller from re-sending the same packet, the device controller will respond to the error packet by acknowledging it with either an ACK or NYET response.

11.4.4.6.3.3.3 Priming Transmit Endpoints

Priming a transmit endpoint will cause the device controller to fetch the device transfer descriptor (dTD) for the transaction pointed to by the device queue head (dQH).

After the dTD is fetched, it will be stored in the dQH until the device controller completes the transfer described by the dTD. Storing the dTD in the dQH allows the device controller to fetch the operating context needed to handle a request from the host without the need to follow the linked list, starting at the dQH when the host request is received.

8. Also, set the DATA_SHIFT_DIR and SHIFT_NUM_BITS if it is required to shift the data left or right before it is output.
9. Indicate if the read data needs to color-space-converted and stored in a different RGB format by setting the READ_MODE_OUTPUT_IN_RGB_FORMAT field accordingly.
10. Set the WORD_LENGTH field appropriately: 0 = 16-bit input, 1 = 8-bit input, 2 = 18-bit input, 3 = 24-bit input if READ_MODE_OUTPUT_IN_RGB_FORMAT is required. Also, select the correct 16/18/24 bit data format with the corresponding fields in LCDIF_CTRL register.
11. Set the READ_MODE_NUM_PACKED_SUBWORDS field in LCDIF_CTRL2 according to the number of subwords per word required to be packed.
12. Set the READ_PACK_DIR to 1 if it is required to store the data in big-endian format.
13. Set the LCD_DATABUS_WIDTH appropriately: 0 = 16-bit output, 1 = 8-bit output, 2 = 18-bit output, 3 = 24-bit output.
14. Enable the necessary IRQs.

13.2.4.6 MPU Interface

The MPU interface is used to transfer data and commands between the SoC via the eLCDIF and the external display at modest data rates.

Bus master or PIO transactions using the LCDIF_DATA register can be used for MPU mode write operations. For MPU mode read operations, only PIO can be used. eLCDIF can support the 6800 as well as the 8080 MPU protocol. If DOTCLK_MODE, DVI_MODE and VSYNC_MODE bits in LCDIF_CTRL registers are 0, it implies that the block is in MPU interface mode of operation. The LCDIF MPU mode has four basic timing parameters: Setup and Hold for the Command/Data register selection (TCS, TCH) and Setup and Hold for the Data bus (TDS, TDH). These parameters are expressed in DISPLAY_CLK cycles. The LCD_WR signal is used as the write strobe while LCD_RS signal is typically used to switch between command and data modes.

CSI_CSISR field descriptions (continued)

Field	Description
	0 SOF is not detected. 1 SOF is detected.
15 F2_INT	CCIR Field 2 Interrupt Status. Indicates the presence of field 2 of video in CCIR mode. (Cleared automatically when current field does not match) **NOTE:** Only works in CCIR Interlace mode. 0 Field 2 of video is not detected 1 Field 2 of video is about to start
14 F1_INT	CCIR Field 1 Interrupt Status. Indicates the presence of field 1 of video in CCIR mode. (Cleared automatically when current field does not match) **NOTE:** Only works in CCIR Interlace mode. 0 Field 1 of video is not detected. 1 Field 1 of video is about to start.
13 COF_INT	Change Of Field Interrupt Status. Indicates that a change of the video field has been detected. Only works in CCIR Interlace mode. Software should read this bit first and then dispatch the new field from F1_INT and F2_INT. (Cleared by writing 1) 0 Video field has no change. 1 Change of video field is detected.
12–8 -	This field is reserved. Reserved. These bits are reserved and should read 0.
7 HRESP_ERR_INT	Hresponse Error Interrupt Status. Indicates that a hresponse error has been detected. (Cleared by writing 1) 0 No hresponse error. 1 Hresponse error is detected.
6–2 -	This field is reserved. Reserved. These bits are reserved and should read 0.
1 ECC_INT	CCIR Error Interrupt. This bit indicates an error has occurred. This only works in CCIR Interlace mode. (Cleared by writing 1) 0 No error detected 1 Error is detected in CCIR coding
0 DRDY	RXFIFO Data Ready. Indicates the presence of data that is ready for transfer in the RxFIFO. (Cleared automatically by reading FIFO) 0 No data (word) is ready 1 At least 1 datum (word) is ready in RXFIFO.

EXAMPLE

Address: 3070_0000h base + 4F0h offset = 3070_04F0h

Bit	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
R W	BACKGROUND_COLOR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_HW_PXP_INPUT_FETCH_BACKGROUND_COLOR_CH0 field descriptions

Field	Description
BACKGROUND_COLOR	background color(in 32bpp format) for any pixels not within the bufffer range specified by the ULC/LRC

13.7.12.71 PXP_HW_PXP_INPUT_FETCH_BACKGROUND_COLOR_CH1

This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

EXAMPLE

Address: 3070_0000h base + 500h offset = 3070_0500h

Bit	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
R W	BACKGROUND_COLOR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PXP_HW_PXP_INPUT_FETCH_BACKGROUND_COLOR_CH1 field descriptions

Field	Description
BACKGROUND_COLOR	background color(in 32bpp format) for any pixels not within the bufffer range specified by the ULC/LRC

13.7.12.72 PXP_HW_PXP_INPUT_FETCH_PITCH

This register defines the control bits for the pxp prefetch_engine sub-block.

The Control register contains the control bits for the pxp prefetch_engine sub-block.

EXAMPLE

HW_PXP_DITHER_FETCH_CTRL_CH1_SET: 0x864

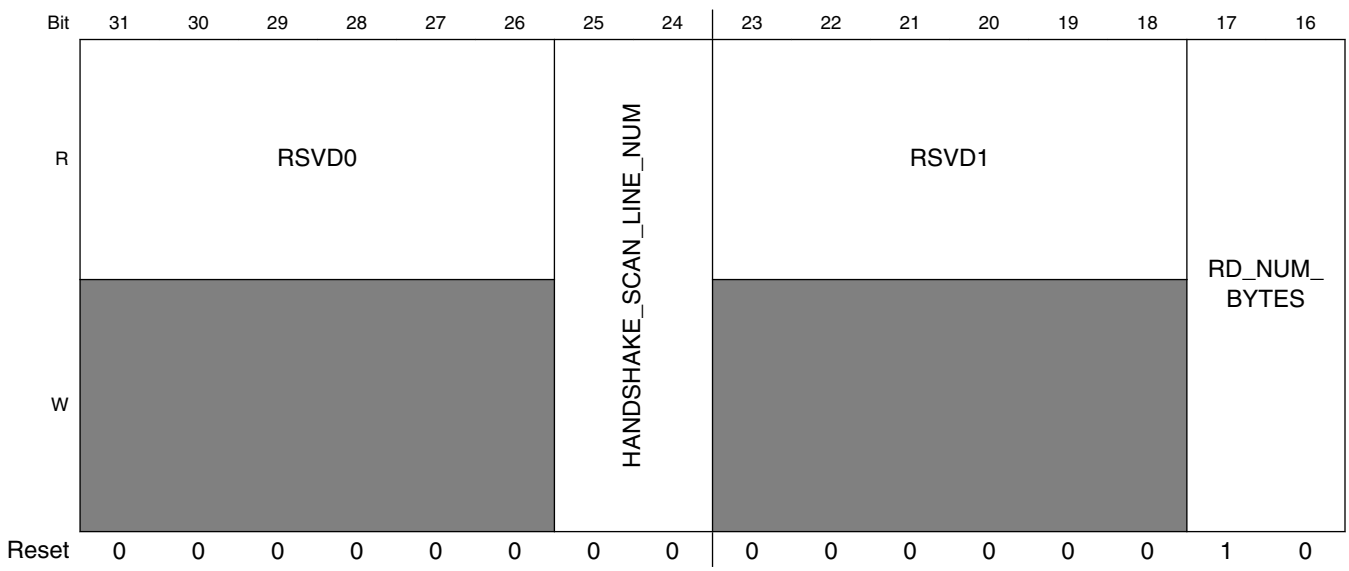
HW_PXP_DITHER_FETCH_CTRL_CH1_CLR: 0x868

HW_PXP_DITHER_FETCH_CTRL_CH1_TOG: 0x86C

The Control register contains the control bits for the pxp prefetch_engine sub-block.

EXAMPLE

Address: 3070_0000h base + 860h offset = 3070_0860h



Fixed recessive bit, used only in extended format. It must be set to '1' by the user for transmission (Tx Buffers) and will be stored with the value received on the CAN bus for Rx receiving buffers. It can be received as either recessive or dominant. If FLEXCAN receives this bit as dominant, then it is interpreted as arbitration loss.

1= Recessive value is compulsory for transmission in Extended Format frames

0= Dominant is not a valid value for transmission in Extended Format frames

IDE - ID Extended Bit

This bit identifies whether the frame format is standard or extended. It is also used as part of the reception filter.

1= Frame format is extended

0= Frame format is standard

RTR - Remote Transmission Request

This bit affects the behavior of Remote Frames and is part of the reception filter. Refer to the tables above and RRS bit in Control 2 Register (FLEXCAN_CTRL2) for additional details.

If FLEXCAN transmits this bit as '1' (recessive) and receives it as '0' (dominant), it is interpreted as arbitration loss. If this bit is transmitted as '0' (dominant), then if it is received as '1' (recessive), the FLEXCAN module treats it as bit error. If the value received matches the value transmitted, it is considered as a successful bit transmission.

1= Indicates the current MB has a Remote Frame to be transmitted if MB is Tx. If the MB is Rx then incoming Remote Request Frames may be stored.

0= Indicates the current MB has a Data Frame to be transmitted. In Rx MB it may be considered in matching processes.

DLC - Length of Data in Bytes

This 4-bit field is the length (in bytes) of the Rx or Tx data, which is located in offset 0x08 through 0x0F of the MB space (see the first table above). In reception, this field is written by the FLEXCAN module, copied from the DLC (Data Length Code) field of the received frame. In transmission, this field is written by the ARM and corresponds to the DLC field value of the frame to be transmitted. When RTR=1, the Frame to be transmitted is a Remote Frame and does not include the data field, regardless of the Length field. The DLC field indicates which DATA BYTEs are valid as shown in the table below.

TIME STAMP - Free-Running Counter Time Stamp

In order to abort a transmission, ARM must write a specific abort code (0b1001) to the CODE field of the Control and Status word. The active MBs configured as transmission must be aborted first and then they may be updated. If the abort code is written to a Mailbox that is currently being transmitted, or to a Mailbox that was already loaded into the SMB for transmission, the write operation is blocked and the MB is kept active, but the abort request is captured and kept pending until one of the following conditions are satisfied:

- The module loses the bus arbitration
- There is an error during the transmission
- The module is put into Freeze Mode
- The module enters in BusOff state
- There is an overload frame

If none of conditions above are reached, the MB is transmitted correctly, the interrupt flag is set in the IFLAG register and an interrupt to the ARM is generated (if enabled). The abort request is automatically cleared when the interrupt flag is set. In the other hand, if one of the above conditions is reached, the frame is not transmitted, therefore the abort code is written into the CODE field, the interrupt flag is set in the IFLAG and an interrupt is (optionally) generated to ARM.

If ARM writes the ABORT code before the transmission begins internally, then the write operation is not blocked, therefore the MB is updated and the interrupt flag is set. In this way ARM just needs to read the abort code to make sure the active MB was *safely inactivated* . Although the AEN bit is asserted and ARM wrote the abort code, in this case the MB is inactivated and not aborted, because the transmission did not start yet. One Mailbox is only aborted when the abort request is captured and kept pending until one of the previous conditions are satisfied.

The abort procedure can be summarized as follows:

1. ARM checks the corresponding IFLAG and clears it, if asserted.
2. ARM writes 0b1001 into the CODE field of the C/S word.
3. ARM waits for the corresponding IFLAG indicating that the frame was either transmitted or aborted.
4. ARM reads the CODE field to check if the frame was either transmitted (CODE=0b1000) or aborted (CODE=0b1001).
5. It is necessary to clear the corresponding IFLAG in order to allow the MB to be reconfigured.