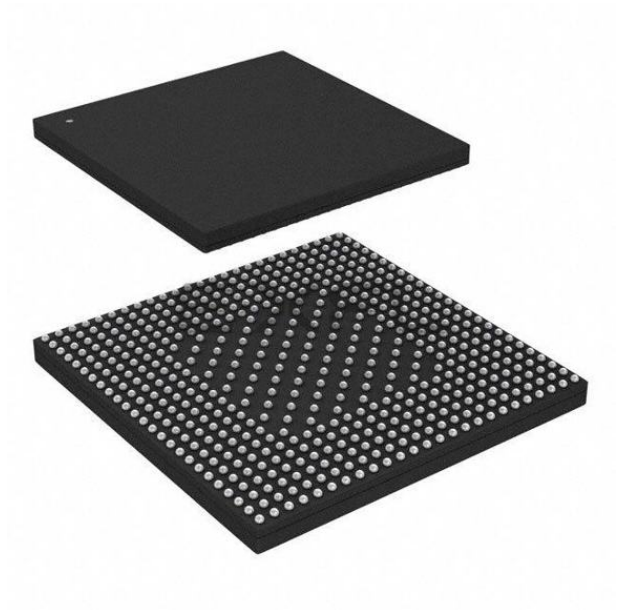


Welcome to E-XFL.COM

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	-20°C ~ 105°C (TJ)
Security Features	A-HAB, ARM TZ, CAAM, CSU, SJC, SNVS
Package / Case	541-LFBGA
Supplier Device Package	541-MAPBGA (19x19)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcimx7s5evm08sc

Chapter 10
Mass Storage

4.8.4 Resource ownership control

The resource ownership control regulates access to the shared peripherals and determines the steering of out-of-band signals.

4.8.4.1 Access control

CCM memory map (continued)

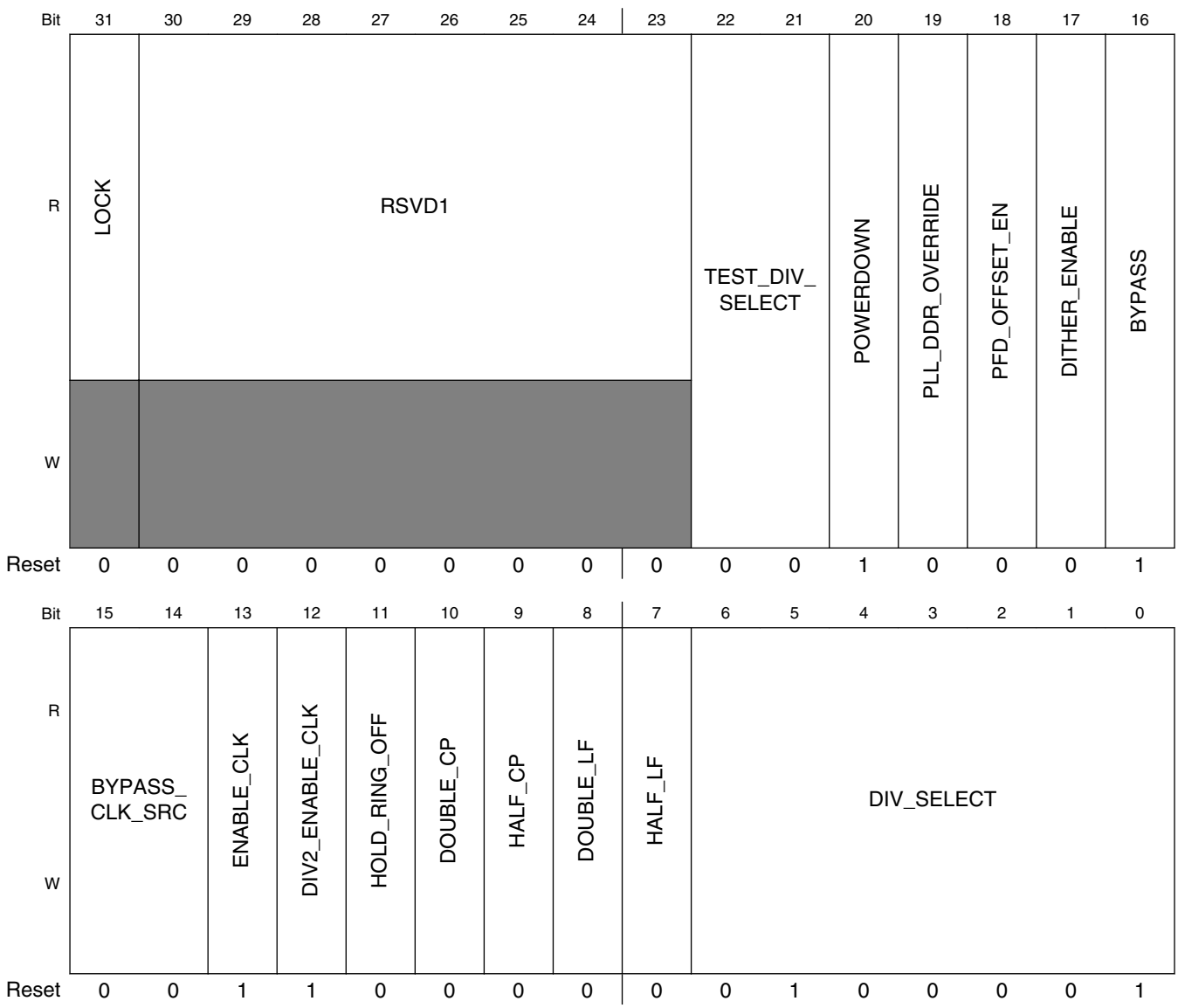
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9594	Miscellaneous Register (CCM_MISC_ROOT43_SET)	32	R/W	0000_0000h	5.2.8.15/ 708
3038_9598	Miscellaneous Register (CCM_MISC_ROOT43_CLR)	32	R/W	0000_0000h	5.2.8.16/ 709
3038_959C	Miscellaneous Register (CCM_MISC_ROOT43_TOG)	32	R/W	0000_0000h	5.2.8.17/ 710
3038_95A0	Post Divider Register (CCM_POST43)	32	R/W	0000_0000h	5.2.8.18/ 711
3038_95A4	Post Divider Register (CCM_POST_ROOT43_SET)	32	R/W	0000_0000h	5.2.8.19/ 714
3038_95A8	Post Divider Register (CCM_POST_ROOT43_CLR)	32	R/W	0000_0000h	5.2.8.20/ 717
3038_95AC	Post Divider Register (CCM_POST_ROOT43_TOG)	32	R/W	0000_0000h	5.2.8.21/ 720
3038_95B0	Pre Divider Register (CCM_PRE43)	32	R/W	0000_0000h	5.2.8.22/ 723
3038_95B4	Pre Divider Register (CCM_PRE_ROOT43_SET)	32	R/W	0000_0000h	5.2.8.23/ 726
3038_95B8	Pre Divider Register (CCM_PRE_ROOT43_CLR)	32	R/W	0000_0000h	5.2.8.24/ 729
3038_95BC	Pre Divider Register (CCM_PRE_ROOT43_TOG)	32	R/W	0000_0000h	5.2.8.25/ 732
3038_95F0	Access Control Register (CCM_ACCESS_CTRL43)	32	R/W	0000_0000h	5.2.8.26/ 735
3038_95F4	Access Control Register (CCM_ACCESS_CTRL_ROOT43_SET)	32	R/W	0000_0000h	5.2.8.27/ 737
3038_95F8	Access Control Register (CCM_ACCESS_CTRL_ROOT43_CLR)	32	R/W	0000_0000h	5.2.8.28/ 740
3038_95FC	Access Control Register (CCM_ACCESS_CTRL_ROOT43_TOG)	32	R/W	0000_0000h	5.2.8.29/ 742
3038_9600	Target Register (CCM_TARGET_ROOT44)	32	R/W	0000_0000h	5.2.8.10/ 699
3038_9604	Target Register (CCM_TARGET_ROOT44_SET)	32	R/W	0000_0000h	5.2.8.11/ 701
3038_9608	Target Register (CCM_TARGET_ROOT44_CLR)	32	R/W	0000_0000h	5.2.8.12/ 703
3038_960C	Target Register (CCM_TARGET_ROOT44_TOG)	32	R/W	0000_0000h	5.2.8.13/ 705
3038_9610	Miscellaneous Register (CCM_MISC44)	32	R/W	0000_0000h	5.2.8.14/ 707
3038_9614	Miscellaneous Register (CCM_MISC_ROOT44_SET)	32	R/W	0000_0000h	5.2.8.15/ 708
3038_9618	Miscellaneous Register (CCM_MISC_ROOT44_CLR)	32	R/W	0000_0000h	5.2.8.16/ 709

Table continues on the next page...

5.2.9.2 Anadig DDR PLL Control Register (CCM_ANALOG_PLL_DDR*n*)

The control register provides control for the 480MHz PLL.

Address: 3036_0000h base + 70h offset + (4d × i), where i=0d to 3d



CCM_ANALOG_PLL_DDR*n* field descriptions

Field	Description
31 LOCK	1 - PLL is currently locked; 0 - PLL is not currently locked.

Table continues on the next page...

CCM_ANALOG_PLL_480*n* field descriptions

Field	Description
31 LOCK	1 - PLL is currently locked; 0 - PLL is not currently locked.
30–29 RSVD1	Always set to zero (0).
28 PFD2_DIV2_ CLKGATE	If set to 1, pll_sys_pfd2_135m_clk is off (power savings). 0: pll_sys_pfd2_135m_clk is enabled.
27 PFD1_DIV2_ CLKGATE	If set to 1, pll_sys_pfd1_166m_clk is off (power savings). 0: pll_sys_pfd1_166m_clk is enabled.
26 PFD0_DIV2_ CLKGATE	If set to 1, pll_sys_pfd0_196m_clk is off (power savings). 0: pll_sys_pfd0_196m_clk is enabled.
25 PFD7_ OVERRIDE	The OVERRIDE bit allows the clock control module to automatically override portions of the register.
24 PFD6_ OVERRIDE	The OVERRIDE bit allows the clock control module to automatically override portions of the register.
23 PFD5_ OVERRIDE	The OVERRIDE bit allows the clock control module to automatically override portions of the register.
22 PFD4_ OVERRIDE	The OVERRIDE bit allows the clock control module to automatically override portions of the register.
21 PFD3_ OVERRIDE	The OVERRIDE bit allows the clock control module to automatically override portions of the register.
20 PFD2_ OVERRIDE	The OVERRIDE bit allows the clock control module to automatically override portions of the register.
19 PFD1_ OVERRIDE	The OVERRIDE bit allows the clock control module to automatically override portions of the register.
18 PFD0_ OVERRIDE	The OVERRIDE bit allows the clock control module to automatically override portions of the register.
17 PLL_480_ OVERRIDE	The OVERRIDE bit allows the clock control module to automatically override portions of the register.
16 BYPASS	Bypass the pll.
15–14 BYPASS_CLK_ SRC	Determines the bypass source.
13 ENABLE_CLK	Enable the clock output.

Table continues on the next page...

CCM_ANALOG_PLL_480*n* field descriptions (continued)

Field	Description
12 POWERDOWN	Powers down the PLL.
11 HOLD_RING_ OFF	Analog debug bit.
10 DOUBLE_CP	Increases the charge pump gain 2x.
9 HALF_CP	Reduces the charge pump gain 2x.
8 DOUBLE_LF	Increases the frequency of the loop filter 2x.
7 HALF_LF	Reduces the frequency of the loop filter 2x.
6 MAIN_DIV4_ CLKGATE	If set to 1, pll_sys_main_120m_clk is off (power savings). 0: pll_sys_main_120m_clk is enabled.
5 MAIN_DIV2_ CLKGATE	If set to 1, pll_sys_main_240m_clk is off (power savings). 0: pll_sys_main_240m_clk is enabled.
4 MAIN_DIV1_ CLKGATE	If set to 1, pll_sys_main_480m_clk is off (power savings). 0: pll_sys_main_480m_clk is enabled.
3–1 RSVD0	Always set to zero (0).
0 DIV_SELECT	This field controls the pll loop divider. 0 - Fout=480MHz; 1 - Fout=528MHz.

7.2.4.12.1.11 Conditional Yielding-Burst DMA Unit

The standard SDMA transfer is based upon a hardware loop that has the following structure:

Hardware Loop

```
loop
load Rn,source            // can be ldf or ld
<computation>            // can be done through functional units
store Rn,dest              // can be st or stf
done 0               // yield
```

This structure needs to be kept independent of the functional units' particularities regarding the context switch. However, there can be variations in the context switch's efficiency, which can depend on the number of data received up to that point, and on the data itself.

The DMA, with its 8-word burst capability, has a preferable context switch period when its address register is 8-word aligned: It is the only moment that occurs once every eight loops when the succession of bursts is not broken by the context switch. When this is not the case, a context switch requires the storing (or loading) of less than eight words, which requires separate accesses and is far less efficient. The rest of the 8-word packet is stored (or loaded) after the context restore, and this is done as separate accesses.

The proposed solution is a conditional yielding, which occurs only when the DMA is in an optimum state. It does not require any modification to the scripts. The condition is decided at the DMA level.

The DMA can be programmed in two modes-conditional or always-true-for every channel, which provides complete flexibility. By default, the DMA is not in conditional mode.

The DMA condition is computed from the FIFO fill level and the various modes, as follows:

- When copy mode is selected, regardless of the transfer direction ('read' or 'write'), the condition is always true.
- In read mode, the condition is always true.
- In write mode, the condition is true when there are four bytes or less in the FIFO; it is false when there are more than four bytes. The 4-byte limit comes from the possibility of saving those bytes as MD with absolutely no impact on the bus accesses.

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

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.5.2.30 LDRPC (Load from RPC to Register)

Operation:

```
GReg[r] ← RPC
```

Assembler:

```
Syntax: ldrpc r

Example: ldrpc 3
```

copies RPC to GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Stores the contents of the RPC in a General Register. That instruction may be used to have more than one level of subroutines.

Instruction Format

IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SCLK field descriptions (continued)

Field	Description
	1 **ENABLED** — Force input path of pad ECSPI1_SCLK 0 **DISABLED** — Input Path is determined by functionality
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field. Select 1 of 7 iomux modes to be used for pad: ECSPI1_SCLK. 000 **ALT0_ECSPI1_SCLK** — Select mux mode: ALT0 mux port: SCLK of instance: ECSPI1 001 **ALT1_UART6_RX_DATA** — Select mux mode: ALT1 mux port: RX_DATA of instance: UART6 010 **ALT2_SD2_DATA4** — Select mux mode: ALT2 mux port: DATA4 of instance: SD2 011 **ALT3_CSI_DATA2** — Select mux mode: ALT3 mux port: DATA2 of instance: CSI 101 **ALT5_GPIO4_IO16** — Select mux mode: ALT5 mux port: IO16 of instance: GPIO4 110 **ALT6_EPDC_PWR_COM** — Select mux mode: ALT6 mux port: PWR_COM of instance: EPDC

8.2.7.87 SW_MUX_CTL_PAD_ECSPI1_MOSI SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MOSI)

SW_MUX_CTL Register

Address: 3033_0000h base + 16Ch offset = 3033_016Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
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	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

DDRC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
307A_0024	Temperature Derate Interval Register (DDRC_DERATEINT)	32	R/W	0000_0000h	9.2.5.2.7/ 2187
307A_0030	Low Power Control Register (DDRC_PWRCTL)	32	R/W	0000_0000h	9.2.5.2.8/ 2188
307A_0034	Low Power Timing Register (DDRC_PWRTMG)	32	R/W	0000_0000h	9.2.5.2.9/ 2189
307A_0038	Hardware Low Power Control Register (DDRC_HWLPCTL)	32	R/W	0000_0000h	9.2.5.2.10/ 2191
307A_0050	Refresh Control Register 0 (DDRC_RFSHCTL0)	32	R/W	0000_0000h	9.2.5.2.11/ 2193
307A_0054	Refresh Control Register 1 (DDRC_RFSHCTL1)	32	R/W	0000_0000h	9.2.5.2.12/ 2195
307A_0060	Refresh Control Register 0 (DDRC_RFSHCTL3)	32	R/W	0000_0000h	9.2.5.2.13/ 2196
307A_0064	Refresh Timing Register (DDRC_RFSHTMG)	32	R/W	0000_0000h	9.2.5.2.14/ 2197
307A_00D0	SDRAM Initialization Register 0 (DDRC_INIT0)	32	R/W	0000_0000h	9.2.5.2.15/ 2198
307A_00D4	SDRAM Initialization Register 1 (DDRC_INIT1)	32	R/W	0000_0000h	9.2.5.2.16/ 2199
307A_00D8	SDRAM Initialization Register 2 (DDRC_INIT2)	32	R/W	0000_0000h	9.2.5.2.17/ 2200
307A_00DC	SDRAM Initialization Register 3 (DDRC_INIT3)	32	R/W	0000_0000h	9.2.5.2.18/ 2201
307A_00E0	SDRAM Initialization Register 4 (DDRC_INIT4)	32	R/W	0000_0000h	9.2.5.2.19/ 2202
307A_00E4	SDRAM Initialization Register 5 (DDRC_INIT5)	32	R/W	0000_0000h	9.2.5.2.20/ 2202
307A_00F4	Rank Control Register (DDRC_RANKCTL)	32	R/W	0000_0000h	9.2.5.2.21/ 2203
307A_0100	SDRAM Timing Register 0 (DDRC_DRAMTMG0)	32	R/W	0000_0000h	9.2.5.2.22/ 2205
307A_0104	SDRAM Timing Register 1 (DDRC_DRAMTMG1)	32	R/W	0000_0000h	9.2.5.2.23/ 2207
307A_0108	SDRAM Timing Register 2 (DDRC_DRAMTMG2)	32	R/W	0000_0000h	9.2.5.2.24/ 2208
307A_010C	SDRAM Timing Register 3 (DDRC_DRAMTMG3)	32	R/W	0000_0000h	9.2.5.2.25/ 2210
307A_0110	SDRAM Timing Register 4 (DDRC_DRAMTMG4)	32	R/W	0000_0000h	9.2.5.2.26/ 2211
307A_0114	SDRAM Timing Register5 (DDRC_DRAMTMG5)	32	R/W	0000_0000h	9.2.5.2.27/ 2212
307A_0118	SDRAM Timing Register 6 (DDRC_DRAMTMG6)	32	R/W	0000_0000h	9.2.5.2.28/ 2214

Table continues on the next page...

2. Software must configure the MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 0 (normal fast boot), and bit 4 to select the ack mode or not. If the data will be sent through DMA mode, the software should configure bit 7 to enable the automatic stop at block gap feature, and configure bit 3-bit 0 to select the ack timeout value according to the SD CLK frequency.
3. Software then needs to configure the Block Attributes Register to set the block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes.
4. Software must configure the Protocol control register to set DTW (data transfer width). If in DDR fast boot mode, DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software needs to configure the Command Argument Register to set argument if needed (no need in normal fast boot).
6. Software must configure the Transfer Type Register to start the boot process. In normal boot mode, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN are kept at the default value. DPSEL bit is set to 1, DTDSEL is set to 1, MSBSEL is set to 1.
7. DMAEN should be configured as 0 in polling mode. And if BCEN is configured as 1, it is recommended to configure the number of blocks in the Block Attributes Register to the maximum value. If in DDR fast boot mode, DDR_EN needs to be set to 1.
8. When the step 6 is configured, the boot process will begin. Software needs to poll the data buffer ready status to read the data from the buffer in time. If a boot timeout happens (ack times out or the first data read times out), an interrupt will be triggered, and software must configure MMC Boot Register to bit 6 to 0 to disable boot. This makes CMD high, then after at least 56 clocks, it is ready to begin a normal initialization process.
9. If there is no timeout, software needs to determine when the data read is finished and then configure MMC Boot Register bit 6 to 0 to disable boot. This will make CMD line high and command completed asserted. After at least 56 clocks, it is ready to begin normal initialization process.
10. Reset the host and then can begin the normal process.

10.3.5.6.2 Alternative fast boot flow

1. Software needs to configure init_active bit (system control register bit 27) to make sure 74 card clocks are finished.
2. Software needs to configure MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 1 (alternative boot), and bit 4 to select the ack mode or not. If data needs to be sent through DMA mode, then configure bit 7 to enable the automatic stop at block gap feature. Software should also configure bit 3-bit 0 to select the ack timeout value according to the SD clock frequency.

11.1.6.10.5 Input Capture and Output Compare

The Input Capture Output Compare block can be used to provide precise hardware timing for input and output events.

11.1.6.10.5.1 Input capture

The TCCR*n* capture registers latch the time value when the corresponding external event occurs. An event can be a rising-, falling-, or either-edge of one of the 1588_TMR*n* signals. An event will cause the corresponding TCSR*n*[TF] timer flag to be set, indicating that an input capture has occurred. If the corresponding interrupt is enabled with the TCSR*n*[TIE] field, an interrupt can be generated.

11.1.6.10.5.2 Output compare

The TCCR*n* compare registers are loaded with the time at which the corresponding event should occur. When the ENET free-running counter value matches the output compare reference value in the TCCR*n* register, the corresponding flag, TCSR*n*[TF], is set, indicating that an output compare has occurred. The corresponding interrupt, if enabled by TCSR*n*[TIE], will be generated.The corresponding 1588_TMR*n* output signal will be asserted according to TCSR*n*[TMODE].

11.1.6.10.5.3 DMA requests

A DMA request can be enabled by setting TCSR*n*[TDRE]. The corresponding DMA request is generated when the TCSR*n*[TF] timer flag is set. When the DMA has completed, the corresponding TCSRn[TF] flag is cleared.

11.1.6.11 FIFO thresholds

The core FIFO thresholds are fully programmable to dynamically change the FIFO operation.

For example, store and forward transfer can be enabled by a simple change in the FIFO threshold registers.

The thresholds are defined in 64-bit words.

The receive and transmit FIFOs both have a depth of 1024 words.

Table 11-50. SIM Card Detect Interrupts

Flag	Flag register	Mask	Mask Register	Description
SDI1	PORT1_DETECT	SDIM1	PORT1_DETECT	SIM Detect Interrupt for port 1
SDI0	PORT0_DETECT	SDIM0	PORT0_DETECT	SIM Detect Interrupt for port 0

11.2.1.8 SIM General Purpose Counter Overview

The SIM module provides a 16-bit counter that can be configured to count using clock sources from the card clock, receive clock, or transmitter clock (ETU Clock). A programmable 16-bit register is provided to compare with the counter for interrupt generation:

- Selectable clock source
- 16-bit counter and comparator
- One Interrupt source (see the table below)

Table 11-51. SIM General Purpose Counter Interrupts

Flag	Flag Register	Mask	Mask Register	Description
GPCNT	XMT_STATUS	GPCNTM	INT_MASK	GPCNT Comparator Interrupt

11.2.1.9 SIM LRC Block Overview

The SIM module contains a block designed to generate Linear Redundancy Checking (LRC) information for both received and transmitted characters. The LRC block has the following features:

- 8-bit LRC value
- Valid LRC Detector
- There are no interrupt sources generated by the SIM LRC block

11.2.1.10 SIM CRC Block Overview

The SIM module contains a block designed to generate Cyclic Redundancy Checking (CRC) information for both received and transmitted characters. The CRC block has the following features:

- 16-bit CRC value
- 16-bit CRC Polynomial Generator

11.3.4.3.8.3 Empty Asynchronous Schedule Detection

The Enhanced Host Controller Interface uses two bits to detect when the asynchronous schedule is empty.

The queue head data structure (see Table 11-78) defines an *H-bit* in the queue head, which allows software to mark a queue head as being the *head* of the reclaim list. The Enhanced Host Controller Interface also keeps a 1-bit flag in the USB_USBSTS register (*Reclamation*) that is set to zero when the Enhanced Interface Host Controller observes a queue head with the H-bit set to one. The reclamation flag in the status register is set to one when any USB transaction from the asynchronous schedule is executed (or whenever the asynchronous schedule starts, see Asynchronous schedule traversal: *Start Event*).

If the Enhanced Host Controller Interface ever encounters an *H-bit* of one and a *Reclamation* bit of zero, the EHCI controller simply stops traversal of the asynchronous schedule.

An example illustrating the H-bit in a schedule is shown in the following figure.

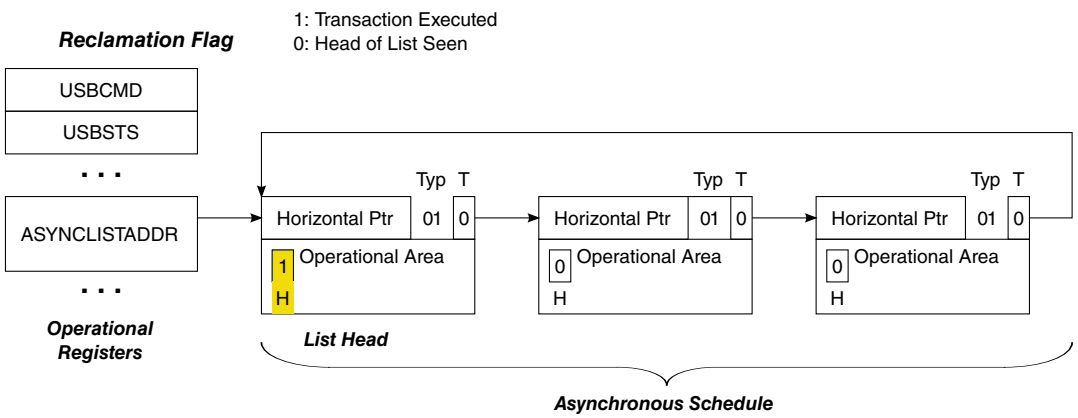


Figure 11-67. Asynchronous Schedule List w/Annotation to Mark Head of List

Software must ensure there is at most one queue head with the *H-bit* set to one, and that it is always coherent with respect to the schedule.

11.3.4.3.8.4 Restarting Asynchronous Schedule Before EOF

There are many situations where the host controller will detect an empty list *long* before the end of the micro-frame.

It is important to remember that under many circumstances the schedule traversal has stopped due to Nak/Nyet responses from all endpoints.

USBx_USBSTS field descriptions (continued)

Field	Description
5 AAI	Interrupt on Async Advance - R/WC. System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a one to the Interrupt on Async Advance Doorbell bit in the n_USBCMD register. This status bit indicates the assertion of that interrupt source. Only used in host operation mode.
4 SEI	System Error- R/WC. This bit is will be set to '1b' when an Error response is seen to a read on the system interface.
3 FRI	Frame List Rollover - R/WC. The Host Controller sets this bit to a one when the Frame List Index rolls over from its maximum value to zero. The exact value at which the rollover occurs depends on the frame list size. For example. If the frame list size (as programmed in the Frame List Size field of the USB_n_USBCMD register) is 1024, the Frame Index Register rolls over every time FRINDEX [13] toggles. Similarly, if the size is 512, the Host Controller sets this bit to a one every time FHINDEX [12] toggles. Only used in host operation mode.
2 PCI	Port Change Detect - R/WC. The Host Controller sets this bit to a one when on any port a Connect Status occurs, a Port Enable/ Disable Change occurs, or the Force Port Resume bit is set as the result of a J-K transition on the suspended port. The Device Controller sets this bit to a one when the port controller enters the full or high-speed operational state. When the port controller exits the full or high-speed operation states due to Reset or Suspend events, the notification mechanisms are the USB Reset Received bit and the DCSuspend bits respectively.
1 UEI	USB Error Interrupt (USBERRINT) - R/WC. When completion of a USB transaction results in an error condition, this bit is set by the Host/Device Controller. This bit is set along with the USBINT bit, if the TD on which the error interrupt occurred also had its interrupt on complete (IOC) bit set. The device controller detects resume signaling only.
0 UI	USB Interrupt (USBINT) - R/WC. This bit is set by the Host/Device Controller when the cause of an interrupt is a completion of a USB transaction where the Transfer Descriptor (TD) has an interrupt on complete (IOC) bit set. This bit is also set by the Host/Device Controller when a short packet is detected. A short packet is when the actual number of bytes received was less than the expected number of bytes.

- BYTE clock (RX_BYTE_CLK_HS0)
- External clock (I_WRAP_CLK)
- APB clock (I_PCLK)

Pixel clock uses one of those clocks (Wrapper, APB), since CSIS V3.3 does not use BYTE clock for Pixel clock in the User defined data type.

The relationship between input and output bandwidth is that the output bandwidth should be faster than input bandwidth. There is an equation of previous relationship.

(freq. of RX_BYTE_CLK_HS) * (number of data lane) * 8 bits ≤ (freq. of Pixel clock) * (bitwidth of image format)

13.5.3.11 Interface Signals

13.5.3.11.1 Reset and clock signal

Table 13-30. Variation of Clock and Reset

Clock and Reset	Frequency	Clock Generator	Usage
I_PRESETn	—	—	Global Asynchronous reset
I_PCLK	—	System controller	Main control clock (system dependent)
RX_BYTE_CLK_HS0	up to 187.5 MHz (1.5 G) 125 MHz (1 G)	Data lane 0 of D-phy	MIPI CSI processes data with ByteCLK
I_WRAP_CLK	up to 200 MHz	External clock source	Output pixel clock
I_DTLPCLK	up to 200 MHz	External test device	For D-phy scan test mode

13.5.3.11.2 PPI (PHY-Protocol Interface) signals

BandCtrl and HSzeroCtl signals of following table are additional signals for D-phy timing configuration.The other PPI signals in the following table are compatible with MIPI D-phy spec.

Table 13-31. Protocol-to-Phy Interface Signal List

Signal	I/O	Description
Clock lane		
STOP_STATE_CLK	I	Stop state indicator (active high)
ULPS_CLK	I	ULPS state indicator (active low)
ENABLE_CLK	O	Clock lane enabler
Data lane (lane0 ~ 1 common, % = 0 and 1)		

Table continues on the next page...

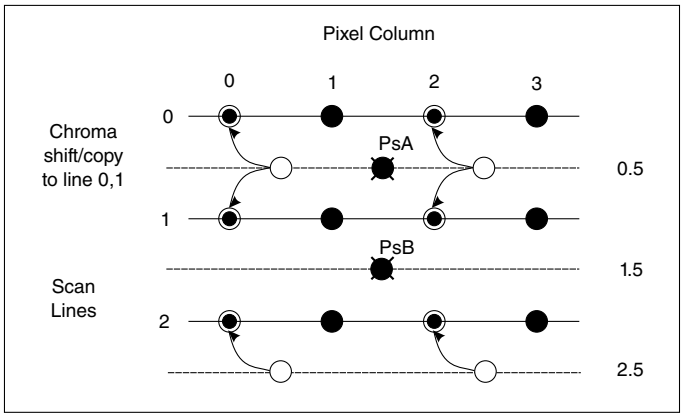


Figure 13-78. Scaled Chroma Computation Examples

The preceding diagram has two examples for the computation of the scaled chroma output pixel. For chroma at output position PsA (vertical position 0.5), interpolation occurs in the X axis using chroma values at column 0 and column 2. However, since line 0 and line 1 have equal chroma values due to chroma line replication, scaling in the Y axis results in replication of chroma values.

For chroma at output position PsB (vertical position 1.5), interpolation occurs in both the X and Y axis. The Y axis is an interpolation since the chroma values copied to scan line 1 and 2 and not the same.

In summary, any output image pixels that map to an odd scan line above and an even scan line below are interpolated vertically. Output image pixels that map to an even scan line above and an odd scan line below are replicated vertically.

13.6.3.7 RGB/YUV444 Image Scaling

For all RGB formats, the RGB pixels are converted up to RGB888 with 8 bits per each color component.

Then each color component is passed to the scaling engine and each component is treated in the same manor. The RGB scaling operation is the same as for the Y scaling operation described in the preceding sections. Also, YUV444 contains a byte for each color plane at each pixel location, so all three color components are scaled in the same manor.

13.6.3.8 Color Space Conversion (CSC)

There are two modules in the PXP to convert pixels between color spaces. They are referred to as CSC1 and CSC2 (for lack of a better naming convention).

PXP_HW_PXP_DITHER_FETCH_CTRL_CH1 field descriptions (continued)

Field	Description
11 RSVD3	Reserved, always set to zero.
10 VFLIP	Enables VFLIP 0x0 **0** — VFLIP disable 0x1 **1** — VFLIP enable
9 HFLIP	Enables HFLIP. 0x0 **0** — HFLIP disable 0x1 **1** — VFLIP enable
8–5 RSVD4	Reserved, always set to zero.
4 BYPASS_ PIXEL_EN	Selects Channel 1 pixel source 0x0 **0** — Channel 1 is from memory 0x1 **1** — Channel 1 is from previous process engine
3 HANDSHAKE_ EN	Enable bit for handshake with the store engine. 0x0 **0** — Handshake with the store engine is disabled 0x1 **1** — Handshake with the store engine is enabled
2 BLOCK_16	Determines the block sixe. 0x0 **8x8** — Block size is 8x8 0x1 **16x16** — Block size is 16x16
1 BLOCK_EN	Choses the prefetch mode. 0x0 **0** — Prefetch in scan mode 0x1 **1** — Prefetch in block mode
0 CH_EN	Channel enable. 0x0 **0** — Prefetch function is disable 0x1 **1** — Prefetch function is enable

13.6.12.121 Pre-fetch engine status Channel 0 Register (PXP_HW_PXP_DITHER_FETCH_STATUS_CH0)

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

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

EXAMPLE

13.6.12.205 16-level Histogram Parameter 2 Register. (PXP_HW_PXP_HIST16_PARAM2)

This register specifies four of the valid values for a 16-level histogram. If all pixels in a bitmap match the 16-level histogram values, STATUS[3] will be set at the end of frame processing. All comparator values should be programmed such that they are consistent with the PANEL_MODE control field.

Address: 3070_0000h base + 2B60h offset = 3070_2B60h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVD11		VALUE11						RSVD10		VALUE10					
W																
Reset	0	0	0	0	1	0	1	1	0	0	0	0	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVD9		VALUE9						RSVD8		VALUE8					
W																
Reset	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0

PXP_HW_PXP_HIST16_PARAM2 field descriptions

Field	Description
31–30 RSVD11	Reserved, always set to zero.
29–24 VALUE11	GRAY11 value for 16-level histogram
23–22 RSVD10	Reserved, always set to zero.
21–16 VALUE10	GRAY10 value for 16-level histogram
15–14 RSVD9	Reserved, always set to zero.
13–8 VALUE9	GRAY9 value for 16-level histogram
7–6 RSVD8	Reserved, always set to zero.
VALUE8	GRAY8 value for 16-level histogram