

Welcome to [E-XFL.COM](http://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.

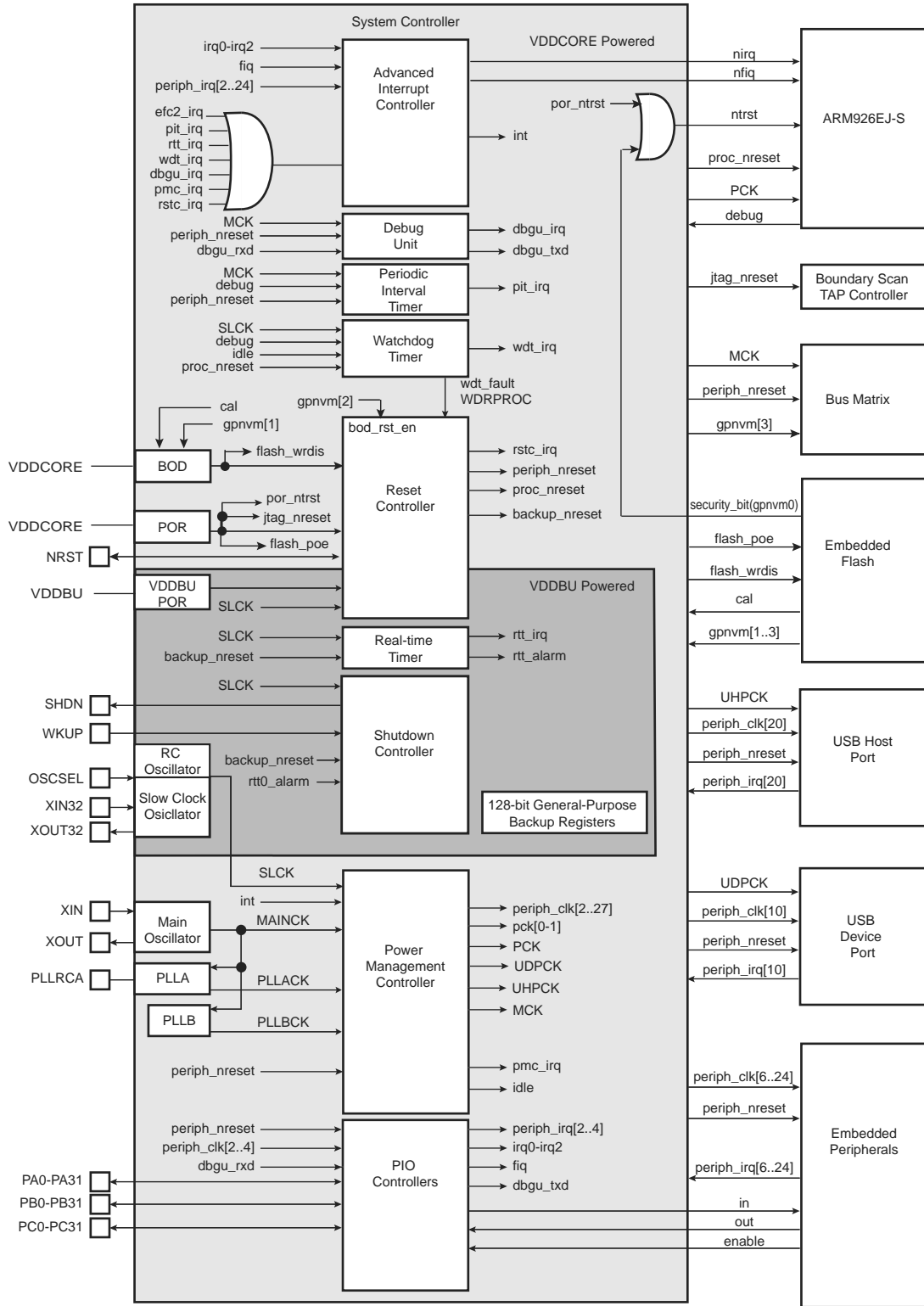
### Applications of [Embedded - Microprocessors](#)

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

Details	
Product Status	Active
Core Processor	ARM926EJ-S
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	180MHz
Co-Processors/DSP	-
RAM Controllers	SDRAM, SRAM
Graphics Acceleration	No
Display & Interface Controllers	LCD, Touchscreen
Ethernet	10/100Mbps
SATA	-
USB	USB 2.0 (3)
Voltage - I/O	1.8V, 2.5V, 3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	217-LFBGA
Supplier Device Package	217-LFBGA (15x15)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/at91sam9xe256b-cu">https://www.e-xfl.com/product-detail/microchip-technology/at91sam9xe256b-cu</a>

## 8.1 System Controller Block Diagram

Figure 8-1. System Controller Block Diagram



Each SAM9XE input/output pin corresponds to a 3-bit register in the BSR. The OUTPUT bit contains data that can be forced on the pad. The INPUT bit facilitates the observability of data applied to the pad. The CONTROL bit selects the direction of the pad.

**Table 11-2. SAM9XE JTAG Boundary Scan Register**

Bit Number	Pin Name	Pin Type	Associated BSR Cells
307	A0	IN/OUT	CONTROL
306			INPUT/OUTPUT
305	A1	IN/OUT	CONTROL
304			INPUT/OUTPUT
303	A10	IN/OUT	CONTROL
302			INPUT/OUTPUT
301	A11	IN/OUT	CONTROL
300			INPUT/OUTPUT
299	A12	IN/OUT	CONTROL
298			INPUT/OUTPUT
297	A13	IN/OUT	CONTROL
296			INPUT/OUTPUT
295	A14	IN/OUT	CONTROL
294			INPUT/OUTPUT
293	A15	IN/OUT	CONTROL
292			INPUT/OUTPUT
291	A16	IN/OUT	CONTROL
290			INPUT/OUTPUT
289	A17	IN/OUT	CONTROL
288			INPUT/OUTPUT
287	A18	IN/OUT	CONTROL
286			INPUT/OUTPUT
285	A19	IN/OUT	CONTROL
284			INPUT/OUTPUT
283	A2	IN/OUT	CONTROL
282			INPUT/OUTPUT
281	A20	IN/OUT	CONTROL
280			INPUT/OUTPUT
279	A21	IN/OUT	CONTROL
278			INPUT/OUTPUT
277	A22	IN/OUT	CONTROL
276			INPUT/OUTPUT
275	A3	IN/OUT	CONTROL
274			INPUT/OUTPUT
273	A4	IN/OUT	CONTROL
272			INPUT/OUTPUT

### 14.4.1 Reset Controller Control Register

**Name:** RSTC\_CR

**Address:** 0xFFFFFD00

**Access:** Write-only

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

- **PROCRST: Processor Reset**

0: No effect.

1: If KEY is correct, resets the processor.

- **PERRST: Peripheral Reset**

0: No effect.

1: If KEY is correct, resets the peripherals.

- **EXTRST: External Reset**

0: No effect.

1: If KEY is correct, asserts the NRST pin.

- **KEY: Password**

Should be written at value 0xA5. Writing any other value in this field aborts the write operation.

### 15.4.1 Real-time Timer Mode Register

**Name:** RTT\_MR  
**Address:** 0xFFFFFD20  
**Access:** Read/Write

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

- **RTPRES: Real-time Timer Prescaler Value**

Defines the number of SLCK periods required to increment the Real-time timer. RTPRES is defined as follows:

RTPRES = 0: The prescaler period is equal to  $2^{16}$ .

RTPRES ≠ 0: The prescaler period is equal to RTPRES.

- **ALMIEN: Alarm Interrupt Enable**

0: The bit ALMS in RTT\_SR has no effect on interrupt.

1: The bit ALMS in RTT\_SR asserts interrupt.

- **RTTINCIEN: Real-time Timer Increment Interrupt Enable**

0: The bit RTTINC in RTT\_SR has no effect on interrupt.

1: The bit RTTINC in RTT\_SR asserts interrupt.

- **RTTRST: Real-time Timer Restart**

1: Reloads and restarts the clock divider with the new programmed value. This also resets the 32-bit counter.

### 21.7.6.1 Software Configuration - Compact Flash

The following configuration has to be performed:

- Assign the EBI CS4 and/or EBI\_CS5 to the CompactFlash Slot 0 or/and Slot 1 by setting the bit EBI\_CS4A or/and EBI\_CS5A in the EBI Chip Select Assignment Register located in the bus matrix memory space.
- The address line A23 is to select I/O (A23 = 1) or Memory mode (A23 = 0) and the address line A22 for REG function.
- A23, CFRNW, CFS0, CFCS1, CFCE1 and CFCE2 signals are multiplexed with PIO lines and thus the dedicated PIOs must be programmed in peripheral mode in the PIO controller.
- Configure a PIO line as an output for CFRST and two others as an input for CFIRQ and CARD DETECT functions respectively.
- Configure SMC CS4 and/or SMC\_CS5 (for Slot 0 or 1) Setup, Pulse, Cycle and Mode according to Compact Flash timings and system bus frequency.

## 24.3 Functional Description

A page in NAND Flash and SmartMedia memories contains an area for main data and an additional area used for redundancy (ECC). The page is organized in 8-bit or 16-bit words. The page size corresponds to the number of words in the main area plus the number of words in the extra area used for redundancy.

Over time, some memory locations may fail to program or erase properly. In order to ensure that data is stored properly over the life of the NAND Flash device, NAND Flash providers recommend to utilize either 1 ECC per 256 bytes of data, 1 ECC per 512 bytes of data or 1 ECC for all of the page.

The only configurations required for ECC are the NAND Flash or the SmartMedia page size (528/2112/4224) and the type of correction wanted (1 ECC for all the page/1 ECC per 256 bytes of data /1 ECC per 512 bytes of data). Page size is configured setting the PAGESIZE field in the ECC Mode Register (ECC\_MR). Type of correction is configured setting the TYPCORRECT field in the ECC Mode Register (ECC\_MR).

ECC is automatically computed as soon as a read (00h)/write (80h) command to the NAND Flash or the SmartMedia is detected. Read and write access must start at a page boundary.

ECC results are available as soon as the counter reaches the end of the main area. Values in the ECC Parity Registers (ECC\_PR0 to ECC\_PR15) are then valid and locked until a new start condition occurs (read/write command followed by address cycles).

### 24.3.1 Write Access

Once the Flash memory page is written, the computed ECC codes are available in the ECC Parity (ECC\_PR0 to ECC\_PR15) registers. The ECC code values must be written by the software application in the extra area used for redundancy. The number of write accesses in the extra area is a function of the value of the type of correction field. For example, for 1 ECC per 256 bytes of data for a page of 512 bytes, only the values of ECC\_PR0 and ECC\_PR1 must be written by the software application. Other registers are meaningless.

### 24.3.2 Read Access

After reading the whole data in the main area, the application must perform read accesses to the extra area where ECC code has been previously stored. Error detection is automatically performed by the ECC controller. Please note that it is mandatory to read consecutively the entire main area and the locations where Parity and NParity values have been previously stored to let the ECC controller perform error detection.

The application can check the ECC Status Registers (ECC\_SR1/ECC\_SR2) for any detected errors. It is up to the application to correct any detected error. ECC computation can detect four different circumstances:

- No error: XOR between the ECC computation and the ECC code stored at the end of the NAND Flash or SmartMedia page is equal to 0. No error flags in the ECC Status Registers (ECC\_SR1/ECC\_SR2).
- Recoverable error: Only the RECERR flags in the ECC Status registers (ECC\_SR1/ECC\_SR2) are set. The corrupted word offset in the read page is defined by the WORDADDR field in the ECC Parity Registers (ECC\_PR0 to ECC\_PR15). The corrupted bit position in the concerned word is defined in the BITADDR field in the ECC Parity Registers (ECC\_PR0 to ECC\_PR15).
- ECC error: The ECCERR flag in the ECC Status Registers (ECC\_SR1/ECC\_SR2) are set. An error has been detected in the ECC code stored in the Flash memory. The position of the corrupted bit can be found by the application performing an XOR between the Parity and the NParity contained in the ECC code stored in the Flash memory.
- Non correctable error: The MULERR flag in the ECC Status Registers (ECC\_SR1/ECC\_SR2) are set. Several unrecoverable errors have been detected in the Flash memory page.

ECC Status Registers, ECC Parity Registers are cleared when a read/write command is detected or a software reset is performed.

These status flags are described in the Peripheral Status Register.

#### 25.3.4 Data Transfers

The serial peripheral triggers its associated PDC channels' transfers using transmit enable (TXEN) and receive enable (RXEN) flags in the transfer control register integrated in the peripheral's user interface.

When the peripheral receives an external data, it sends a Receive Ready signal to its PDC receive channel which then requests access to the Matrix. When access is granted, the PDC receive channel starts reading the peripheral Receive Holding Register (RHR). The read data are stored in an internal buffer and then written to memory.

When the peripheral is about to send data, it sends a Transmit Ready to its PDC transmit channel which then requests access to the Matrix. When access is granted, the PDC transmit channel reads data from memory and puts them to Transmit Holding Register (THR) of its associated peripheral. The same peripheral sends data according to its mechanism.

#### 25.3.5 PDC Flags and Peripheral Status Register

Each peripheral connected to the PDC sends out receive ready and transmit ready flags and the PDC sends back flags to the peripheral. All these flags are only visible in the Peripheral Status Register.

Depending on the type of peripheral, half or full duplex, the flags belong to either one single channel or two different channels.

##### 25.3.5.1 Receive Transfer End

This flag is set when PERIPH\_RCR reaches zero and the last data has been transferred to memory.

It is reset by writing a non zero value in PERIPH\_RCR or PERIPH\_RNCR.

##### 25.3.5.2 Transmit Transfer End

This flag is set when PERIPH\_TCR reaches zero and the last data has been written into peripheral THR.

It is reset by writing a non zero value in PERIPH\_TCR or PERIPH\_TNCR.

##### 25.3.5.3 Receive Buffer Full

This flag is set when PERIPH\_RCR reaches zero with PERIPH\_RNCR also set to zero and the last data has been transferred to memory.

It is reset by writing a non zero value in PERIPH\_TCR or PERIPH\_TNCR.

##### 25.3.5.4 Transmit Buffer Empty

This flag is set when PERIPH\_TCR reaches zero with PERIPH\_TNCR also set to zero and the last data has been written into peripheral THR.

It is reset by writing a non zero value in PERIPH\_TCR or PERIPH\_TNCR.



## 27.9.12 PMC Programmable Clock Register

**Name:** PMC\_PCKx  
**Address:** 0xFFFFFC40  
**Access:** Read/Write

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
–	–	–	PRES			CSS	

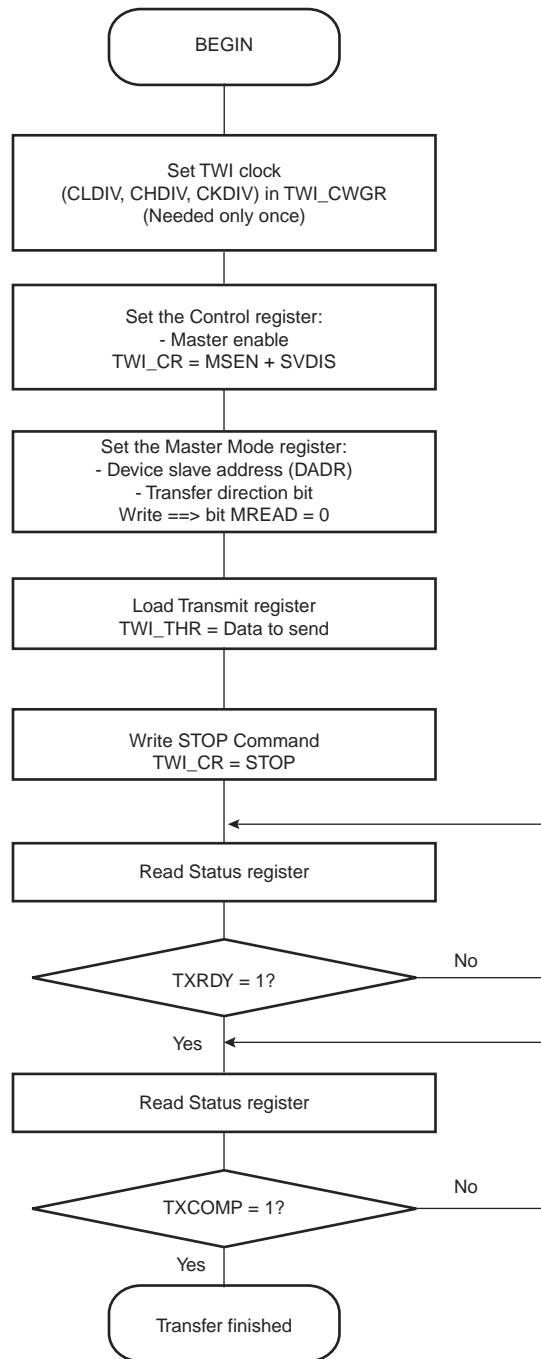
### • CSS: Master Clock Selection

CSS		Clock Source Selection
0	0	Slow Clock is selected
0	1	Main Clock is selected
1	0	PLL A Clock is selected
1	1	PLL B Clock is selected

### • PRES: Programmable Clock Prescaler

PRES			Programmable Clock
0	0	0	Selected clock
0	0	1	Selected clock divided by 2
0	1	0	Selected clock divided by 4
0	1	1	Selected clock divided by 8
1	0	0	Selected clock divided by 16
1	0	1	Selected clock divided by 32
1	1	0	Selected clock divided by 64
1	1	1	Reserved

Figure 32-15. TWI Write Operation with Single Data Byte without Internal Address



## Write Sequence

In the case of a Write sequence (SVREAD is low), the RXRDY (Receive Holding Register Ready) flag is set as soon as a character has been received in the TWI\_RHR (TWI Receive Holding Register). RXRDY is reset when reading the TWI\_RHR.

TWI continues receiving data until a STOP condition or a REPEATED\_START + an address different from SADR is detected. Note that at the end of the write sequence TXCOMP flag is set and SVACC reset.

See Figure 32-26 on page 486.

## Clock Synchronization Sequence

In the case where TWI\_THR or TWI\_RHR is not written/read in time, TWI performs a clock synchronization.

Clock stretching information is given by the SCLWS (Clock Wait state) bit.

See Figure 32-28 on page 487 and Figure 32-29 on page 488.

## General Call

In the case where a GENERAL CALL is performed, GACC (General Call ACCESS) flag is set.

After GACC is set, it is up to the programmer to interpret the meaning of the GENERAL CALL and to decode the new address programming sequence.

See Figure 32-27 on page 486.

## PDC

As it is impossible to know the exact number of data to receive/send, the use of PDC is NOT recommended in SLAVE mode.

### 32.6.5.5 Data Transfer

#### Read Operation

The read mode is defined as a data requirement from the master.

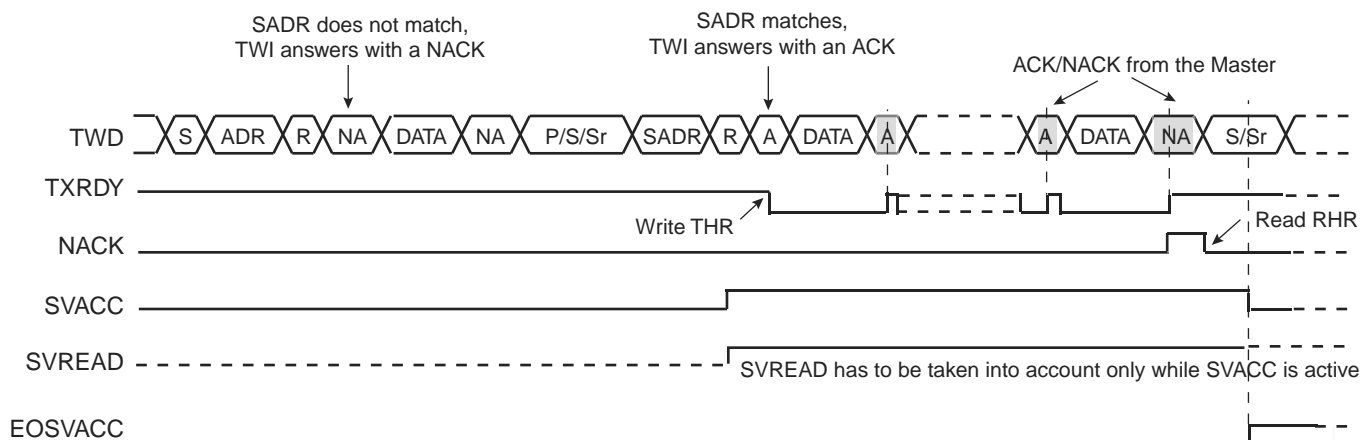
After a START or a REPEATED START condition is detected, the decoding of the address starts. If the slave address (SADR) is decoded, SVACC is set and SVREAD indicates the direction of the transfer.

Until a STOP or REPEATED START condition is detected, TWI continues sending data loaded in the TWI\_THR.

If a STOP condition or a REPEATED START + an address different from SADR is detected, SVACC is reset.

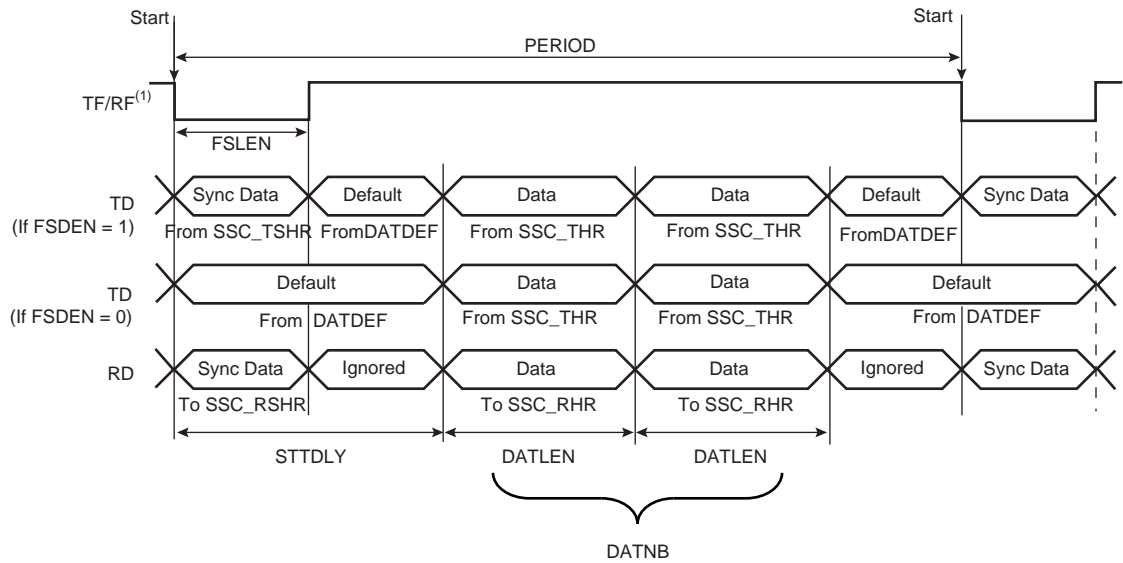
Figure 32-25 on page 485 describes the write operation.

**Figure 32-25. Read Access Ordered by a MASTER**



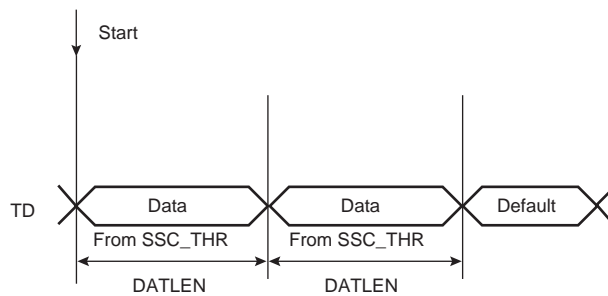
Notes: 1. When SVACC is low, the state of SVREAD becomes irrelevant.

**Figure 34-13. Transmit and Receive Frame Format in Edge/Pulse Start Modes**



Note: 1. Example of input on falling edge of TF/RF.

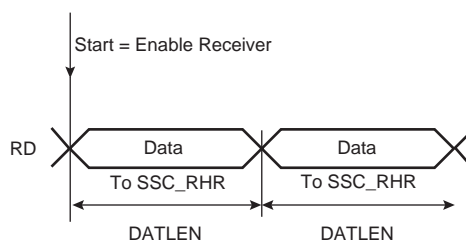
**Figure 34-14. Transmit Frame Format in Continuous Mode**



Start: 1. TXEMPTY set to 1  
2. Write into the SSC\_THR

Note: 1. STTDLY is set to 0. In this example, SSC\_THR is loaded twice. FSDEN value has no effect on the transmission. SyncData cannot be output in continuous mode.

**Figure 34-15. Receive Frame Format in Continuous Mode**



Note: 1. STTDLY is set to 0.

### 35.6.8 TC Register B

**Name:** TC\_RBx [x=0..2]

**Address:** 0xFFFA0018 (0)[0], 0xFFFA0058 (0)[1], 0xFFFA0098 (0)[2], 0xFFFD0018 (1)[0], 0xFFFD0058 (1)[1], 0xFFFD0098 (1)[2]

**Access:** Read-only if WAVE = 0, Read/Write if WAVE = 1

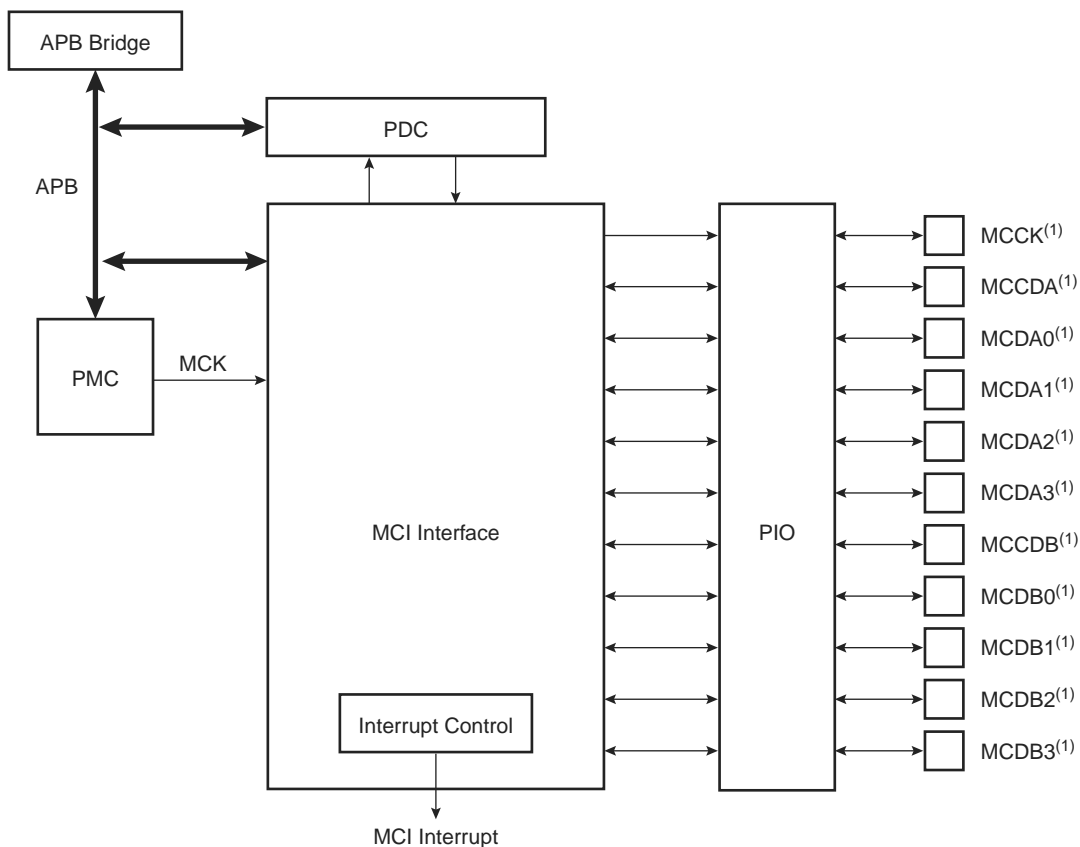
31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
RB							
7	6	5	4	3	2	1	0
RB							

- **RB: Register B**

RB contains the Register B value in real time.

## 36.2 Block Diagram

Figure 36-1. Block Diagram



Note: 1. When several MCI (x MCI) are embedded in a product, MCCK refers to MCIx\_CK, MCCDA to MCIx\_CDA, MCCDB to MCIx\_CDB, MCDAy to MCIx\_DAy, MCDBy to MCIx\_DBy.

**Table 37-1. Receive Buffer Descriptor Entry (Continued)**

Bit	Function
22	Type ID match
21	VLAN tag detected (i.e., type id of 0x8100)
20	Priority tag detected (i.e., type id of 0x8100 and null VLAN identifier)
19:17	VLAN priority (only valid if bit 21 is set)
16	Concatenation format indicator (CFI) bit (only valid if bit 21 is set)
15	End of frame - when set the buffer contains the end of a frame. If end of frame is not set, then the only other valid status are bits 12, 13 and 14.
14	Start of frame - when set the buffer contains the start of a frame. If both bits 15 and 14 are set, then the buffer contains a whole frame.
13:12	Receive buffer offset - indicates the number of bytes by which the data in the first buffer is offset from the word address. Updated with the current values of the network configuration register. If jumbo frame mode is enabled through bit 3 of the network configuration register, then bits 13:12 of the receive buffer descriptor entry are used to indicate bits 13:12 of the frame length.
11:0	Length of frame including FCS (if selected). Bits 13:12 are also used if jumbo frame mode is selected.

To receive frames, the buffer descriptors must be initialized by writing an appropriate address to bits 31 to 2 in the first word of each list entry. Bit zero must be written with zero. Bit one is the wrap bit and indicates the last entry in the list.

The start location of the receive buffer descriptor list must be written to the receive buffer queue pointer register before setting the receive enable bit in the network control register to enable receive. As soon as the receive block starts writing received frame data to the receive FIFO, the receive buffer manager reads the first receive buffer location pointed to by the receive buffer queue pointer register.

If the filter block then indicates that the frame should be copied to memory, the receive data DMA operation starts writing data into the receive buffer. If an error occurs, the buffer is recovered. If the current buffer pointer has its wrap bit set or is the 1024<sup>th</sup> descriptor, the next receive buffer location is read from the beginning of the receive descriptor list. Otherwise, the next receive buffer location is read from the next word in memory.

There is an 11-bit counter to count out the 2048 word locations of a maximum length, receive buffer descriptor list. This is added with the value originally written to the receive buffer queue pointer register to produce a pointer into the list. A read of the receive buffer queue pointer register returns the pointer value, which is the queue entry currently being accessed. The counter is reset after receive status is written to a descriptor that has its wrap bit set or rolls over to zero after 1024 descriptors have been accessed. The value written to the receive buffer pointer register may be any word-aligned address, provided that there are at least 2048 word locations available between the pointer and the top of the memory.

Section 3.6 of the AMBA 2.0 specification states that bursts should not cross 1K boundaries. As receive buffer manager writes are bursts of two words, to ensure that this does not occur, it is best to write the pointer register with the least three significant bits set to zero. As receive buffers are used, the receive buffer manager sets bit zero of the first word of the descriptor to indicate *used*. If a receive error is detected the receive buffer currently being written is recovered. Previous buffers are not recovered. Software should search through the *used* bits in the buffer descriptors to find out how many frames have been received. It should be checking the start-of-frame and end-of-frame bits, and not rely on the value returned by the receive buffer queue pointer register which changes continuously as more buffers are used.

For CRC errored frames, excessive length frames or length field mismatched frames, all of which are counted in the statistics registers, it is possible that a frame fragment might be stored in a sequence of receive buffers. Software can detect this by looking for start of frame bit set in a buffer following a buffer with no end of frame bit set.

### 37.5.9 Interrupt Enable Register

**Name:** EMAC\_IER

**Address:** 0xFFFC4028

**Access:** Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	PTZ	PFR	HRESP	ROVR	–	–
7	6	5	4	3	2	1	0
TCOMP	TXERR	RLE	TUND	TXUBR	RXUBR	RCOMP	MFD

- **MFD: Management Frame sent**

Enable management done interrupt.

- **RCOMP: Receive Complete**

Enable receive complete interrupt.

- **RXUBR: Receive Used Bit Read**

Enable receive used bit read interrupt.

- **TXUBR: Transmit Used Bit Read**

Enable transmit used bit read interrupt.

- **TUND: Ethernet Transmit Buffer Underrun**

Enable transmit underrun interrupt.

- **RLE: Retry Limit Exceeded**

Enable retry limit exceeded interrupt.

- **TXERR**

Enable transmit buffers exhausted in mid-frame interrupt.

- **TCOMP: Transmit Complete**

Enable transmit complete interrupt.

- **ROVR: Receive Overrun**

Enable receive overrun interrupt.

- **HRESP: Hresp not OK**

Enable Hresp not OK interrupt.

- **PFR: Pause Frame Received**

Enable pause frame received interrupt.



### 37.5.17 Specific Address 1 Top Register

**Name:** EMAC\_SA1T

**Address:** 0xFFFC409C

**Access:** Read/Write

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

- **ADDR**

The most significant bits of the destination address, that is bits 47 to 32.

### 38.5.2.5 Transmit Data Cancellation

Some endpoints have dual banks whereas some endpoints have only one bank. The procedure to cancel transmission data held in these banks is described below.

To see the organization of dual-bank availability refer to Table 38-1 "USB Endpoint Description".

#### Endpoints Without Dual Banks

There are two possibilities: In one case, TXPKTRDY field in UDP\_CSR has already been set. In the other instance, TXPKTRDY is not set.

- TXPKTRDY is not set:
  - Reset the endpoint to clear the FIFO (pointers). (See, Section 38.6.9 "UDP Reset Endpoint Register".)
- TXPKTRDY has already been set:
  - Clear TXPKTRDY so that no packet is ready to be sent
  - Reset the endpoint to clear the FIFO (pointers). (See, Section 38.6.9 "UDP Reset Endpoint Register".)

#### Endpoints With Dual Banks

There are two possibilities: In one case, TXPKTRDY field in UDP\_CSR has already been set. In the other instance, TXPKTRDY is not set.

- TXPKTRDY is not set:
  - Reset the endpoint to clear the FIFO (pointers). (See, Section 38.6.9 "UDP Reset Endpoint Register".)
- TXPKTRDY has already been set:
  - Clear TXPKTRDY and read it back until actually read at 0.
  - Set TXPKTRDY and read it back until actually read at 1.
  - Clear TXPKTRDY so that no packet is ready to be sent.
  - Reset the endpoint to clear the FIFO (pointers). (See, Section 38.6.9 "UDP Reset Endpoint Register".)

- **WAKEUP: Disable USB Bus Interrupt**

0: No effect.

1: Disables USB Bus Wakeup Interrupt.

## 39.4 Functional Description

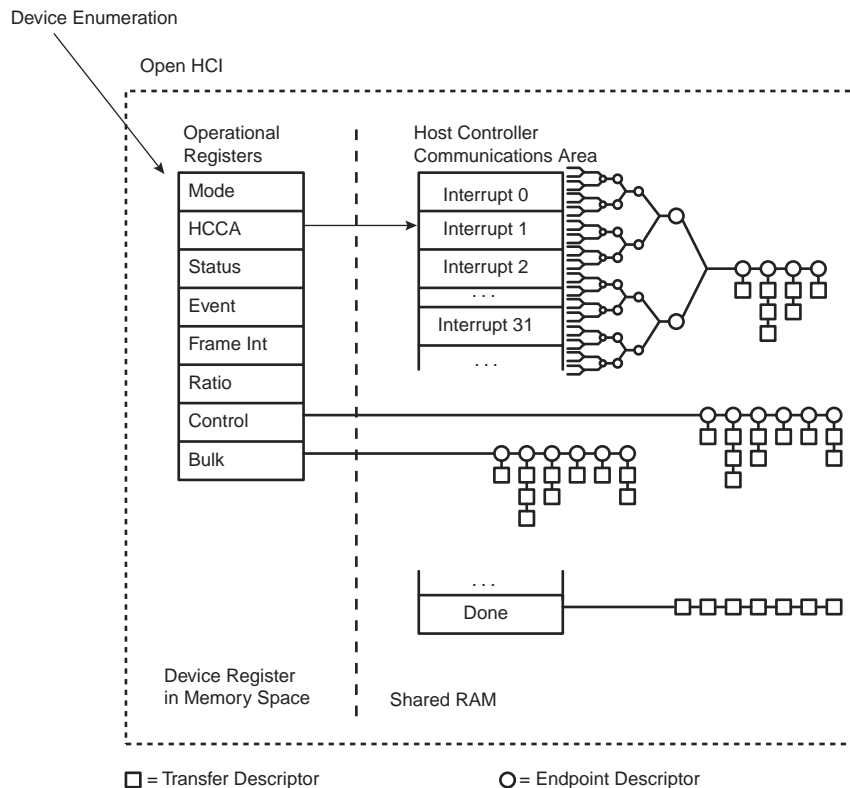
Please refer to the Open Host Controller Interface Specification for USB Release 1.0.a.

### 39.4.1 Host Controller Interface

There are two communication channels between the Host Controller and the Host Controller Driver. The first channel uses a set of operational registers located on the USB Host Controller. The Host Controller is the target for all communications on this channel. The operational registers contain control, status and list pointer registers. They are mapped in the memory mapped area. Within the operational register set there is a pointer to a location in the processor address space named the Host Controller Communication Area (HCCA). The HCCA is the second communication channel. The host controller is the master for all communication on this channel. The HCCA contains the head pointers to the interrupt Endpoint Descriptor lists, the head pointer to the done queue and status information associated with start-of-frame processing.

The basic building blocks for communication across the interface are Endpoint Descriptors (ED, 4 double words) and Transfer Descriptors (TD, 4 or 8 double words). The host controller assigns an Endpoint Descriptor to each endpoint in the system. A queue of Transfer Descriptors is linked to the Endpoint Descriptor for the specific endpoint.

Figure 39-2. USB Host Communication Channels



## 40.4 Image Sensor Interface (ISI) User Interface

**Table 40-9. Register Mapping**

Offset	Register	Name	Access	Reset
0x00	ISI Control 1 Register	ISI_CR1	Read/Write	0x00000002
0x04	ISI Control 2 Register	ISI_CR2	Read/Write	0x00000000
0x08	ISI Status Register	ISI_SR	Read-only	0x00000000
0x0C	ISI Interrupt Enable Register	ISI_IER	Write-only	–
0x10	ISI Interrupt Disable Register	ISI_IDR	Write-only	–
0x14	ISI Interrupt Mask Register	ISI_IMR	Read-only	0x00000000
0x18	Reserved	–	–	–
0x1C	Reserved	–	–	–
0x20	ISI Preview Size Register	ISI_PSIZE	Read/Write	0x00000000
0x24	ISI Preview Decimation Factor Register	ISI_PDECF	Read/Write	0x00000010
0x28	ISI Preview Primary FBD Register	ISI_PPFBD	Read/Write	0x00000000
0x2C	ISI Codec DMA Base Address Register	ISI_CDBA	Read/Write	0x00000000
0x30	ISI CSC YCrCb To RGB Set 0 Register	ISI_Y2R_SET0	Read/Write	0x6832cc95
0x34	ISI CSC YCrCb To RGB Set 1 Register	ISI_Y2R_SET1	Read/Write	0x00007102
0x38	ISI CSC RGB To YCrCb Set 0 Register	ISI_R2Y_SET0	Read/Write	0x01324145
0x3C	ISI CSC RGB To YCrCb Set 1 Register	ISI_R2Y_SET1	Read/Write	0x01245e38
0x40	ISI CSC RGB To YCrCb Set 2 Register	ISI_R2Y_SET2	Read/Write	0x01384a4b
0x44–0xF8	Reserved	–	–	–
0xFC	Reserved	–	–	–

Note: Several parts of the ISI controller use the pixel clock provided by the image sensor (ISI\_PCK). Thus the user must first program the image sensor to provide this clock (ISI\_PCK) before programming the Image Sensor Controller.