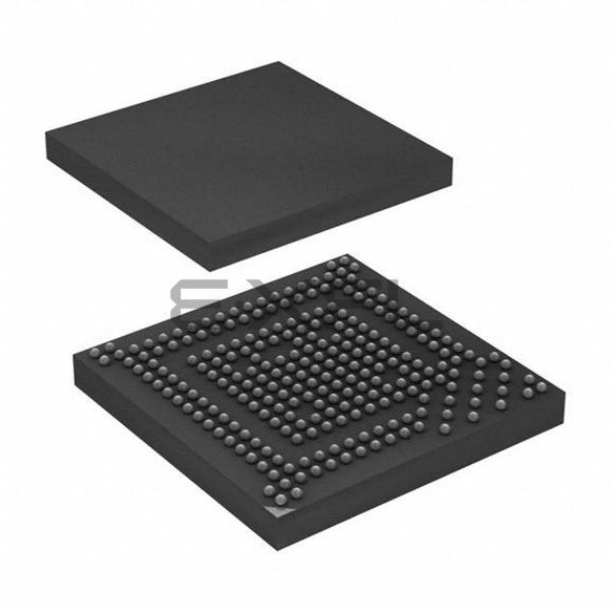
Microchip Technology - AT91SAM9G20B-CFU Datasheet



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	ARM926EJ-S
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	400MHz
Co-Processors/DSP	-
RAM Controllers	SDRAM, SRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10/100Mbps
SATA	-
USB	USB 2.0 (2)
Voltage - I/O	1.8V, 3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	247-TFBGA
Supplier Device Package	247-TFBGA (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at91sam9g20b-cfu

19.6.1 Bus Multiplexing

The EBI offers a complete set of control signals that share the 32-bit data lines, the address lines of up to 26 bits and the control signals through a multiplex logic operating in function of the memory area requests.

Multiplexing is specifically organized in order to guarantee the maintenance of the address and output control lines at a stable state while no external access is being performed. Multiplexing is also designed to respect the data float times defined in the Memory Controllers. Furthermore, refresh cycles of the SDRAM are executed independently by the SDRAM Controller without delaying the other external Memory Controller accesses.

19.6.2 Pull-up Control

The EBI Chip Select Assignment Register (EBI_CSA) in Section 18.6 “Chip Configuration User Interface” permits enabling of on-chip pull-up resistors on the data bus lines not multiplexed with the PIO Controller lines. The pull-up resistors are enabled after reset. Setting the EBI_CSA.EBI_DBPUC bit disables the pull-up resistors on the lines D0–D15. Enabling the pull-up resistor on the lines D16–D31 can be performed by programming the appropriate PIO controller.

19.6.3 Static Memory Controller

For information on the Static Memory Controller, refer to Section 20. “Static Memory Controller (SMC)”.

19.6.4 SDRAM Controller

For information on the SDRAM Controller, refer to Section 21. “SDRAM Controller (SDRAMC)”.

19.6.5 ECC Controller

For information on the ECC Controller, refer to Section 22. “Error Correction Code Controller (ECC)”.

19.6.6 CompactFlash Support

The External Bus Interface integrates circuitry that interfaces to CompactFlash devices.

The CompactFlash logic is driven by the Static Memory Controller (SMC) on the NCS4 and/or NCS5 address space. Programming the EBI_CS4A and/or EBI_CS5A bit in the EBI_CSA register to the appropriate value enables this logic. For details on this register, refer to Section 18. “SAM9G20 Bus Matrix”. Access to an external CompactFlash device is then made by accessing the address space reserved to NCS4 and/or NCS5 (i.e., between 0x5000 0000 and 0x5FFF FFFF for NCS4 and between 0x6000 0000 and 0x6FFF FFFF for NCS5).

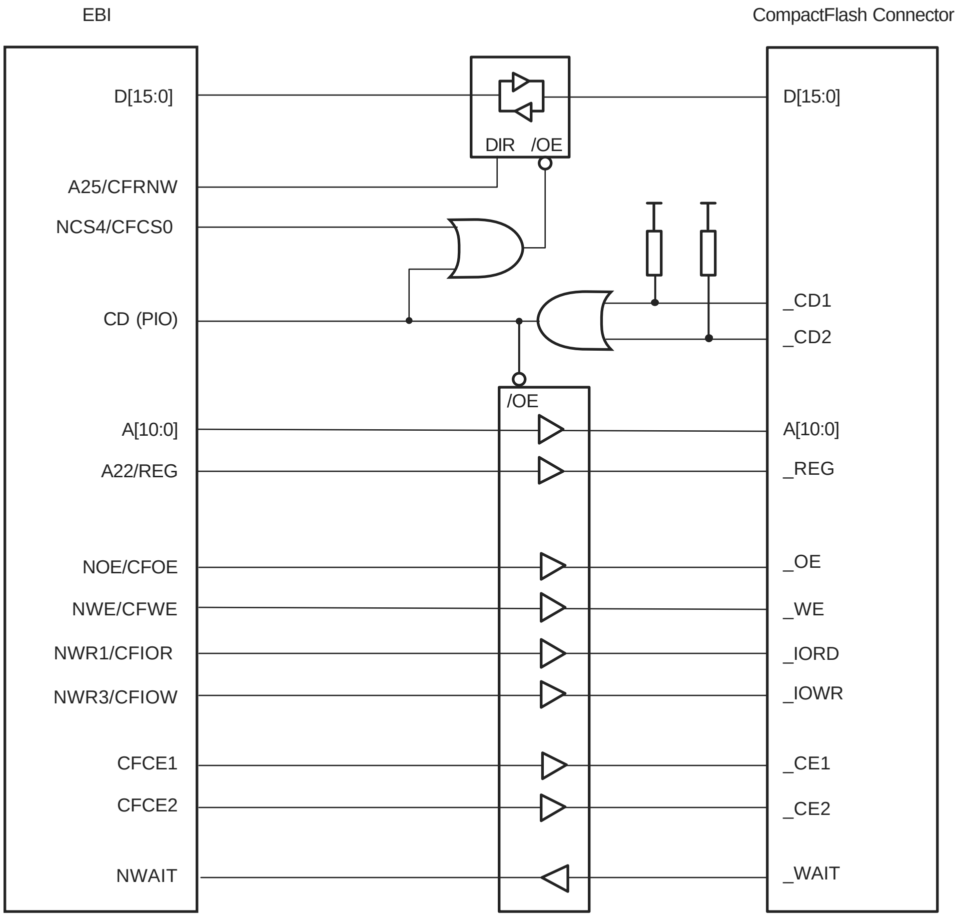
All CompactFlash modes (Attribute Memory, Common Memory, I/O and True IDE) are supported but the signals _IOIS16 (I/O and True IDE modes) and _ATA SEL (True IDE mode) are not handled.

19.6.6.1 I/O Mode, Common Memory Mode, Attribute Memory Mode and True IDE Mode

Within the NCS4 and/or NCS5 address space, the current transfer address is used to distinguish I/O mode, common memory mode, attribute memory mode and True IDE mode.

The different modes are accessed through a specific memory mapping as illustrated on Figure 19-3. A[23:21] bits of the transfer address are used to select the desired mode as described in Table 19-5.

Figure 19-5: CompactFlash Application Example



19.6.7 NAND Flash Support

The External Bus Interface integrates circuitry that interfaces to NAND Flash devices.

19.6.7.1 External Bus Interface

The NAND Flash logic is driven by the Static Memory Controller on the NCS3 address space. Programming the EBI_CS3A field in the EBI_CSA register to the appropriate value enables the NAND Flash logic. For details on this register, refer to Section 18. “SAM9G20 Bus Matrix”. Access to an external NAND Flash device is then made by accessing the address space reserved to NCS3 (i.e., between 0x4000 0000 and 0x4FFF FFFF).

The NAND Flash Logic drives the read and write command signals of the SMC on the NANDOE and NANDWE signals when the NCS3 signal is active. NANDOE and NANDWE are invalidated as soon as the transfer address fails to lie in the NCS3 address space. See Figure 19-6 for more information. For details on the waveforms, refer to Section 20. “Static Memory Controller (SMC)”.

20.8.1 Read Waveforms

The read cycle is shown on Figure 20-8.

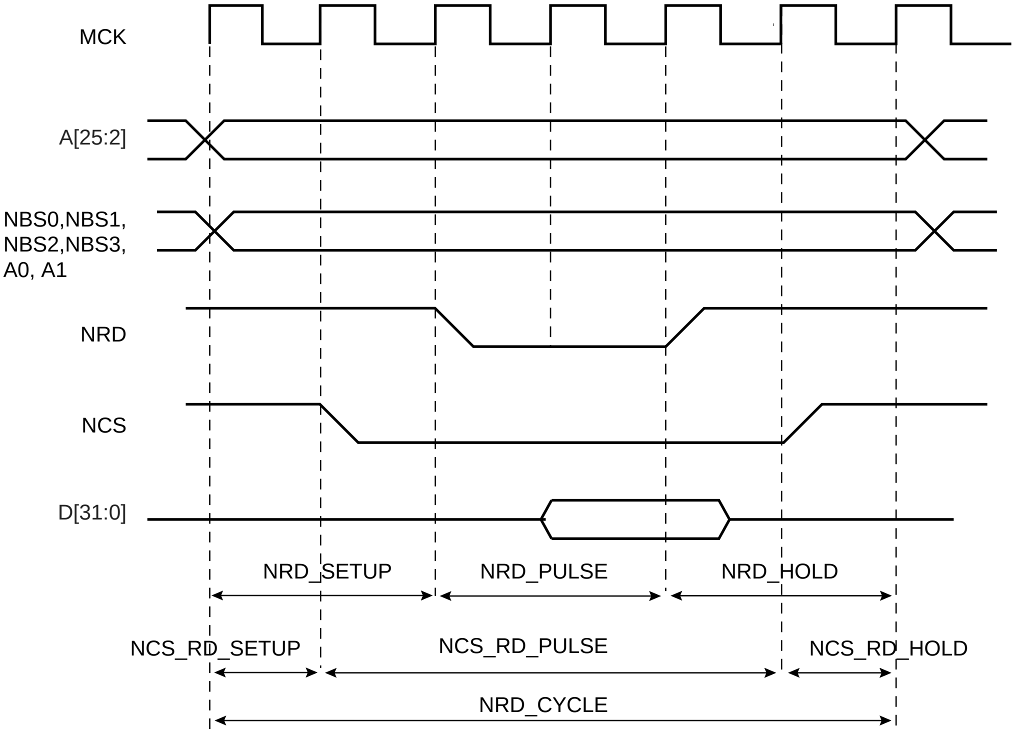
The read cycle starts with the address setting on the memory address bus, i.e.:

{A[25:2], A1, A0} for 8-bit devices

{A[25:2], A1} for 16-bit devices

A[25:2] for 32-bit devices.

Figure 20-8: Standard Read Cycle



20.8.1.1 NRD Waveform

The NRD signal is characterized by a setup timing, a pulse width and a hold timing.

1. NRD_SETUP: the NRD setup time is defined as the setup of address before the NRD falling edge;
2. NRD_PULSE: the NRD pulse length is the time between NRD falling edge and NRD rising edge;
3. NRD_HOLD: the NRD hold time is defined as the hold time of address after the NRD rising edge.

20.8.1.2 NCS Waveform

Similarly, the NCS signal can be divided into a setup time, pulse length and hold time:

1. NCS_RD_SETUP: the NCS setup time is defined as the setup time of address before the NCS falling edge.
2. NCS_RD_PULSE: the NCS pulse length is the time between NCS falling edge and NCS rising edge;
3. NCS_RD_HOLD: the NCS hold time is defined as the hold time of address after the NCS rising edge.

20.8.1.3 Read Cycle

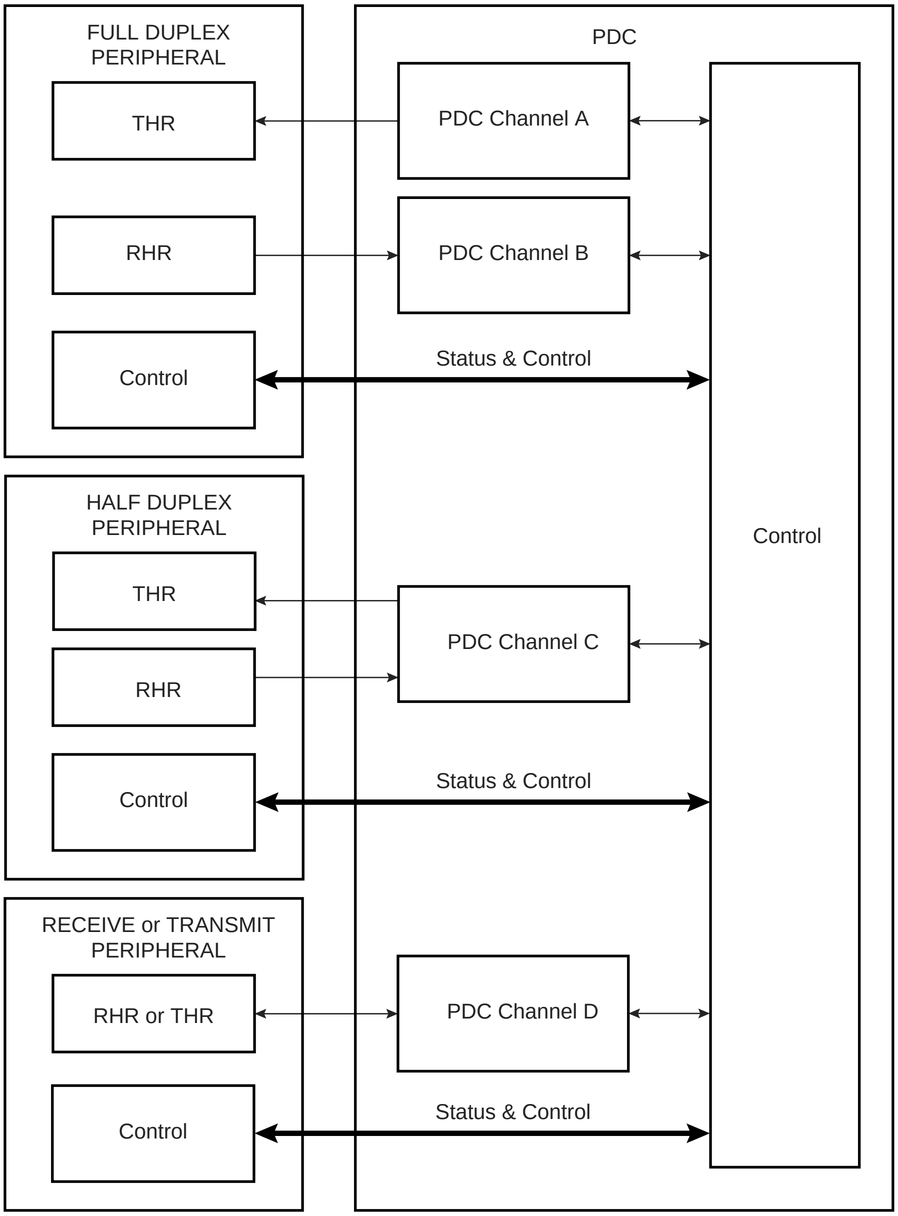
The NRD_CYCLE time is defined as the total duration of the read cycle, i.e., from the time where address is set on the address bus to the point where address may change. The total read cycle time is equal to:

NRD_CYCLE = NRD_SETUP + NRD_PULSE + NRD_HOLD

= NCS_RD_SETUP + NCS_RD_PULSE + NCS_RD_HOLD

23.2 Block Diagram

Figure 23-1: Block Diagram



23.3 Functional Description

23.3.1 Configuration

The PDC channel user interface enables the user to configure and control data transfers for each channel. The user interface of each PDC channel is integrated into the associated peripheral user interface.

The user interface of a serial peripheral, whether it is full or half duplex, contains four 32-bit pointers (RPR, RNPR, TPR, TNPR) and four 16-bit counter registers (RCR, RNCR, TCR, TNCR). However, the transmit and receive parts of each type are programmed differently: the transmit and receive parts of a full duplex peripheral can be programmed at the same time, whereas only one part (transmit or receive) of a half duplex peripheral can be programmed at a time.

32-bit pointers define the access location in memory for current and next transfer, whether it is for read (transmit) or write (receive). 16-bit counters define the size of current and next transfers. It is possible, at any moment, to read the number of transfers left for each channel.

23.4.2 Receive Counter Register

Name:PERIPH_RCR

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
RXCTR							
7	6	5	4	3	2	1	0
RXCTR							

RXCTR: Receive Counter Register

RXCTR must be set to receive buffer size.

When a half duplex peripheral is connected to the PDC, RXCTR = TXCTR.

0: Stops peripheral data transfer to the receiver

1–65535 = Starts peripheral data transfer if corresponding channel is active

26. Advanced Interrupt Controller (AIC)

26.1 Overview

The Advanced Interrupt Controller (AIC) is an 8-level priority, individually maskable, vectored interrupt controller, providing handling of up to thirty-two interrupt sources. It is designed to substantially reduce the software and real-time overhead in handling internal and external interrupts.

The AIC drives the nFIQ (fast interrupt request) and the nIRQ (standard interrupt request) inputs of an Arm processor. Inputs of the AIC are either internal peripheral interrupts or external interrupts coming from the product's pins.

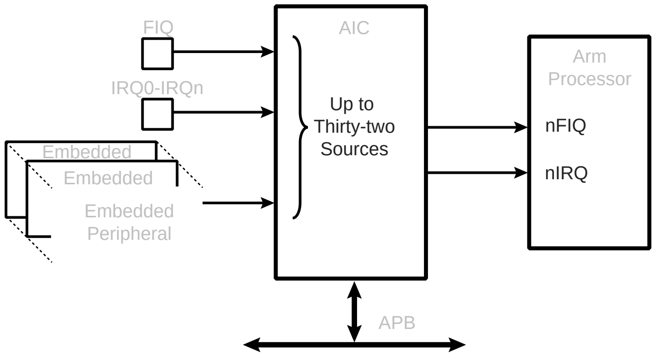
The 8-level Priority Controller allows the user to define the priority for each interrupt source, thus permitting higher priority interrupts to be serviced even if a lower priority interrupt is being treated.

Internal interrupt sources can be programmed to be level sensitive or edge triggered. External interrupt sources can be programmed to be positive-edge or negative-edge triggered or high-level or low-level sensitive.

The fast forcing feature redirects any internal or external interrupt source to provide a fast interrupt rather than a normal interrupt.

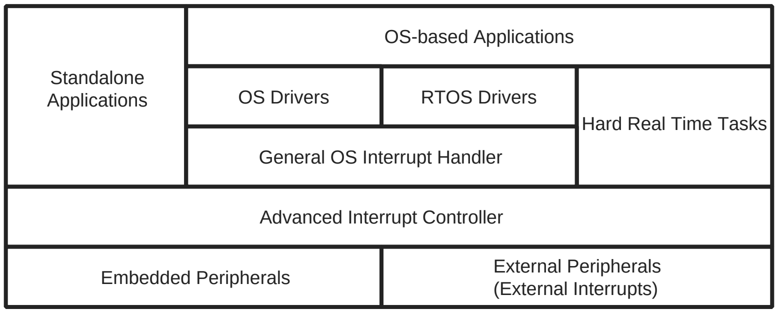
26.2 Block Diagram

Figure 26-1: Block Diagram



26.3 Application Block Diagram

Figure 26-2: Description of the Application Block



27.5.6 Debug Unit Status Register

Name:DBGU_SR

Access: Read-only

31	30	29	28	27	26	25	24
COMMRX	COMMTX	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	RXBUFF	TXBUFE	–	TXEMPTY	–
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PARE	FRAME	OVRE	ENDTX	ENDRX	–	TXRDY	RXRDY

RXRDY: Receiver Ready

0: No character has been received since the last read of the DBGU_RHR or the receiver is disabled.

1: At least one complete character has been received, transferred to DBGU_RHR and not yet read.

TXRDY: Transmitter Ready

0: A character has been written to DBGU_THR and not yet transferred to the Shift Register, or the transmitter is disabled.

1: There is no character written to DBGU_THR not yet transferred to the Shift Register.

ENDRX: End of Receiver Transfer

0: The End of Transfer signal from the receiver Peripheral Data Controller channel is inactive.

1: The End of Transfer signal from the receiver Peripheral Data Controller channel is active.

ENDTX: End of Transmitter Transfer

0: The End of Transfer signal from the transmitter Peripheral Data Controller channel is inactive.

1: The End of Transfer signal from the transmitter Peripheral Data Controller channel is active.

OVRE: Overrun Error

0: No overrun error has occurred since the last RSTSTA.

1: At least one overrun error has occurred since the last RSTSTA.

FRAME: Framing Error

0: No framing error has occurred since the last RSTSTA.

1: At least one framing error has occurred since the last RSTSTA.

PARE: Parity Error

0: No parity error has occurred since the last RSTSTA.

1: At least one parity error has occurred since the last RSTSTA.

TXEMPTY: Transmitter Empty

0: There are characters in DBGU_THR, or characters being processed by the transmitter, or the transmitter is disabled.

1: There are no characters in DBGU_THR and there are no characters being processed by the transmitter.

TXBUFE: Transmission Buffer Empty

0: The buffer empty signal from the transmitter PDC channel is inactive.

1: The buffer empty signal from the transmitter PDC channel is active.

28.6.1 PIO Controller PIO Enable Register

Name:PIO_PER

Access: Write-only

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24

23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16

15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8

7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

P0–P31: PIO Enable

0: No effect.

1: Enables the PIO to control the corresponding pin (disables peripheral control of the pin).

28.6.29 PIO Output Write Status Register

Name:PIO_OWSR

Access: Read-only

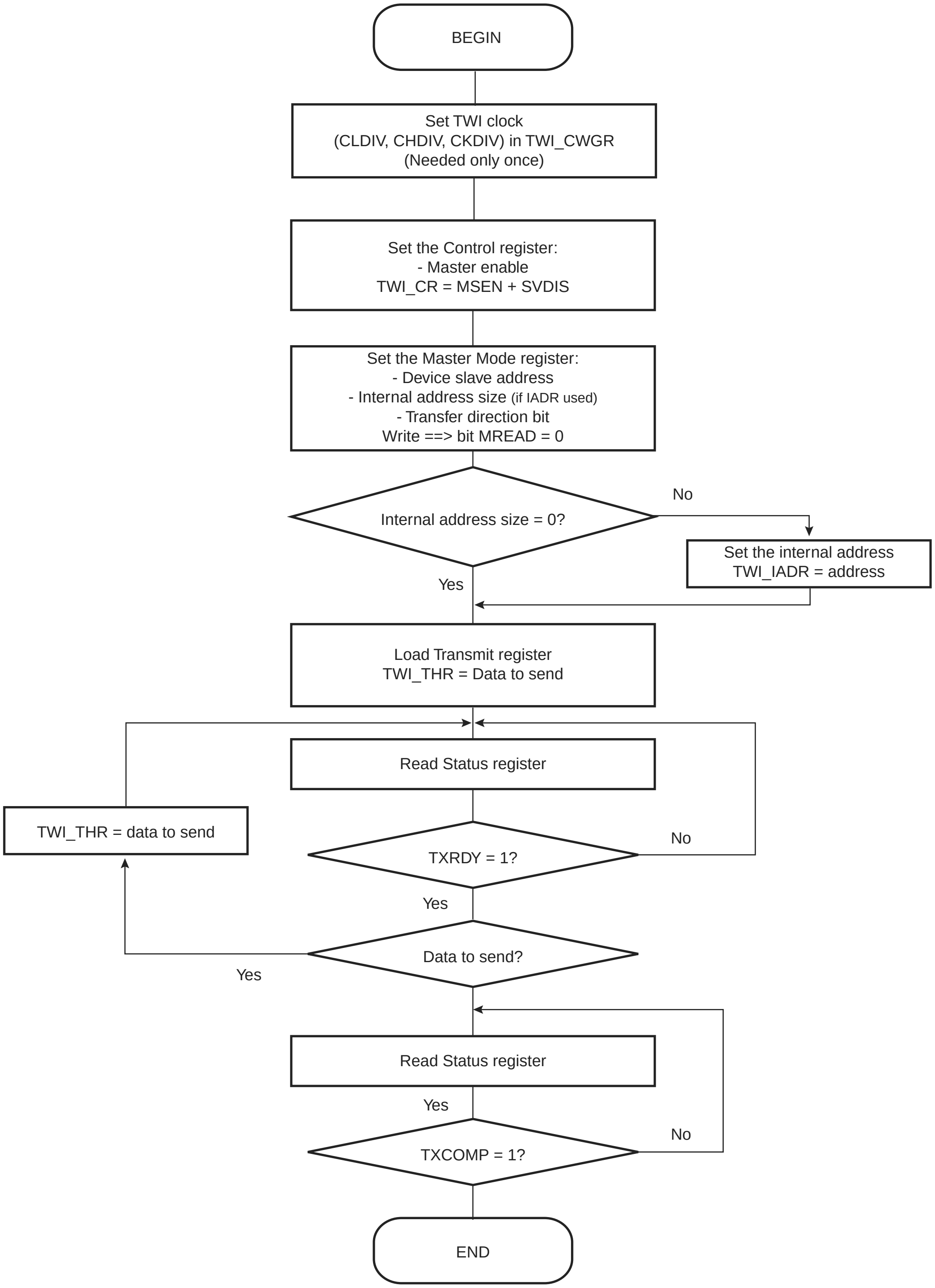
31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

P0–P31: Output Write Status

0: Writing PIO_ODSR does not affect the I/O line.

1: Writing PIO_ODSR affects the I/O line.

Figure 30-16: TWI Write Operation with Multiple Data Bytes with or without Internal Address



30.6.4 Multi-master Mode

30.6.4.1 Definition

More than one master may handle the bus at the same time without data corruption by using arbitration.

Arbitration starts as soon as two or more masters place information on the bus at the same time, and stops (arbitration is lost) for the master that intends to send a logical one while the other master sends a logical zero.

As soon as arbitration is lost by a master, it stops sending data and listens to the bus in order to detect a stop. When the stop is detected, the master who has lost arbitration may put its data on the bus by respecting arbitration.

Arbitration is illustrated in Figure 30-21.

30.6.4.2 Different Multi-master Modes

Two multi-master modes may be distinguished:

1. TWI is considered as a Master only and will never be addressed.
2. TWI may be either a Master or a Slave and may be addressed.

Note: In both Multi-master modes arbitration is supported.

- TWI as Master Only

In this mode, TWI is considered as a Master only (MSEN is always at one) and must be driven like a Master with the ARBLST (ARBitration Lost) flag in addition.

If arbitration is lost (ARBLST = 1), the programmer must reinitiate the data transfer.

If the user starts a transfer (ex.: DADR + START + W + Write in THR) and if the bus is busy, the TWI automatically waits for a STOP condition on the bus to initiate the transfer (see Figure 30-20).

Note: The state of the bus (busy or free) is not indicated in the user interface.

- TWI as Master or Slave

The automatic reversal from Master to Slave is not supported in case of a lost arbitration.

Then, in the case where TWI may be either a Master or a Slave, the programmer must manage the pseudo Multi-master mode described in the steps below.

1. Program TWI in Slave mode (SADR + MSDIS + SVEN) and perform Slave Access (if TWI is addressed).
2. If TWI has to be set in Master mode, wait until TXCOMP flag is at 1.
3. Program Master mode (DADR + SVDIS + MSEN) and start the transfer (ex: START + Write in THR).
4. As soon as the Master mode is enabled, TWI scans the bus in order to detect if it is busy or free. When the bus is considered as free, TWI initiates the transfer.
5. As soon as the transfer is initiated and until a STOP condition is sent, the arbitration becomes relevant and the user must monitor the ARBLST flag.
6. If the arbitration is lost (ARBLST is set to 1), the user must program the TWI in Slave mode in the case where the Master that won the arbitration wanted to access the TWI.
7. If TWI has to be set in Slave mode, wait until TXCOMP flag is at 1 and then program the Slave mode.

Note: In the case where the arbitration is lost and TWI is addressed, TWI will not acknowledge even if it is programmed in Slave mode as soon as ARBLST is set to 1. Then, the Master must repeat SADR.

31.7.9 USART Baud Rate Generator Register

Name:US_BRGR

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	FP		

15	14	13	12	11	10	9	8
CD							

7	6	5	4	3	2	1	0
CD							

CD: Clock Divider

CD	USART_MODE ≠ ISO7816			USART_MODE = ISO7816
	SYNC = 0		SYNC = 1	
	OVER = 0	OVER = 1		
0	Baud Rate Clock Disabled			
1–65535	Baud Rate = Selected Clock/16/CD	Baud Rate = Selected Clock/8/CD	Baud Rate = Selected Clock /CD	Baud Rate = Selected Clock/CD/FI_DI_RATIO

FP: Fractional Part

0: Fractional divider is disabled.

1–7: Baudrate resolution, defined by FP x 1/8.

32.8.12 SSC Receive Compare 1 Register

Name:SSC_RC1R

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
CP1							
7	6	5	4	3	2	1	0
CP1							

CP1: Receive Compare Data 1

34.9.8 MCI Response Register

Name: MCI_RSPR

Access: Read-only

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

RSP: Response

Note: The response register can be read by N accesses at the same MCI_RSPR or at consecutive addresses (0x20 to 0x2C). N depends on the size of the response.

35.5.9 Interrupt Enable Register

Name:EMAC_IER

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.

PTZ: Pause Time Zero

Enable pause time zero interrupt.

35.5.26.8 Deferred Transmission Frames Register

Name:EMAC_DTF

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
DTF							

7	6	5	4	3	2	1	0
DTF							

DTF: Deferred Transmission Frames

A 16-bit register counting the number of frames experiencing deferral due to carrier sense being active on their first attempt at transmission. Frames involved in any collision are not counted nor are frames that experienced a transmit underrun.

Example

The first FBD, stored at address 0x30000, defines the location of the first frame buffer.

Destination Address: frame buffer ID0 0x02A000

Next FBD address: 0x30010

Second FBD, stored at address 0x30010, defines the location of the second frame buffer.

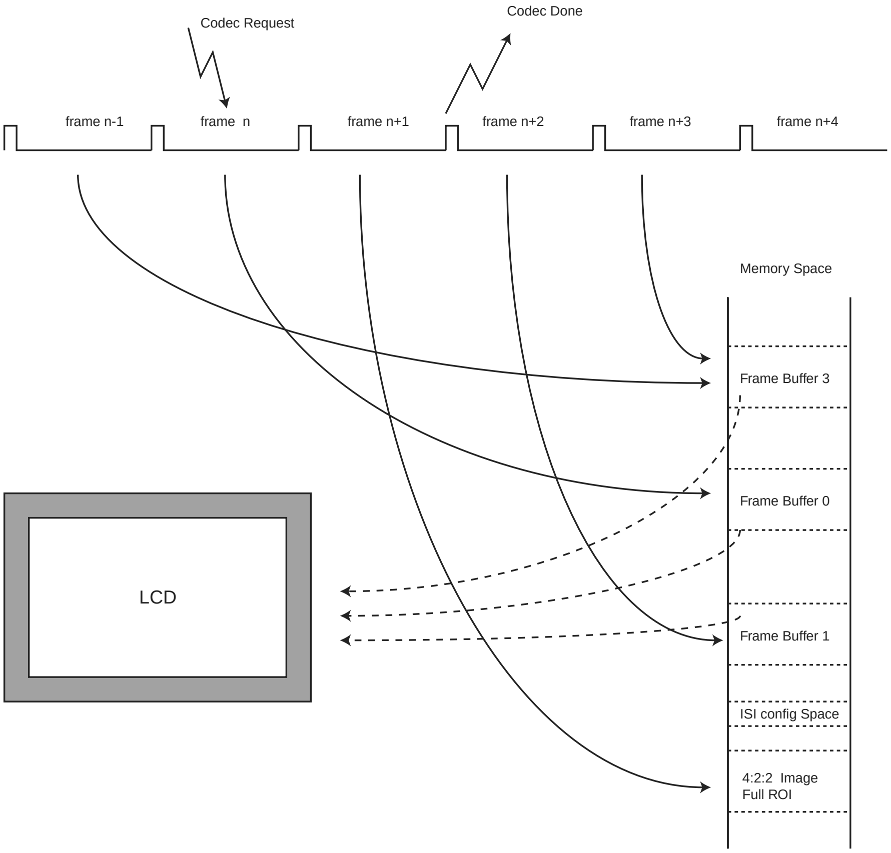
Destination Address: frame buffer ID1 0x3A000

Transfer width: 32 bit

Next FBD address: 0x30000, wrapping to first FBD.

Using this technique, several frame buffers can be configured through the linked list. Figure 38-6 illustrates a typical three frame buffer application. Frame n is mapped to frame buffer 0, frame n+1 is mapped to frame buffer 1, frame n+2 is mapped to Frame buffer 2, further frames wrap. A codec request occurs, and the full-size 4:2:2 encoded frame is stored in a dedicated memory space.

Figure 38-6: Three Frame Buffers Application and Memory Mapping



38.3.5 Codec Path

38.3.5.1 Color Space Conversion

Depending on user selection, this module can be bypassed so that input YCrCb stream is directly connected to the format converter module. If the RGB input stream is selected, this module converts RGB to YCrCb color space with the formulas given below:

$$\begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix} = \begin{bmatrix} C_0 & C_1 & C_2 \\ C_3 & -C_4 & -C_5 \\ -C_6 & -C_7 & C_8 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} Y_{off} \\ Cr_{off} \\ Cb_{off} \end{bmatrix}$$

An example of coefficients is given below:

$$\begin{cases} Y = 0.257 \cdot R + 0.504 \cdot G + 0.098 \cdot B + 16 \\ C_r = 0.439 \cdot R - 0.368 \cdot G - 0.071 \cdot B + 128 \\ C_b = -0.148 \cdot R - 0.291 \cdot G + 0.439 \cdot B + 128 \end{cases}$$

38.3.5.2 Memory Interface

Dedicated FIFO are used to support packed memory mapping. YCrCb pixel components are sent in a single 32-bit word in a contiguous space (packed). Data is stored in the order of natural scan lines. Planar mode is not supported.

38.3.5.3 DMA Features

Unlike preview datapath, codec datapath DMA mode does not support linked list operation. Only the CODEC_DMA_ADDR register is used to configure the frame buffer base address.

THMASK: Threshold Mask

0: 4, 8 and 16 AHB bursts are allowed

1: 8 and 16 AHB bursts are allowed

2: Only 16 AHB bursts are allowed

CODEC_ON: Enable the Codec Path Enable Bit

Write-only.

0: The codec path is disabled

1: The codec path is enabled and the next frame is captured. Refer to bit CDC_PND in “ISI Status Register” .

SLD: Start of Line Delay

SLD pixel clock periods to wait before the beginning of a line.

SFD: Start of Frame Delay

SFD lines are skipped at the beginning of the frame.

5. The Host controller raises the request for the first write transaction. By the time the transaction is completed, a frame boundary is crossed.
6. After completing the first write transaction, the Host controller skips the second write transaction.

Consequence: When this error occurs, the Host controller tries the same IN token again.

Problem Fix/Workaround

This problem can be avoided if the system guarantees that the status update can be completed within the same frame.

44.1.12.2 UHP: ISO OUT transfers

Conditions:

Consider the following sequence:

1. The Host controller sends an ISO OUT token after fetching 16 bytes of data from the system memory.
2. When the Host controller is sending the ISO OUT data, because of system latencies, remaining bytes of the packet are not available. This results in a buffer underrun condition.
3. While there is an underrun condition, if the Host controller is in the process of bit-stuffing, it causes the Host controller to hang.

Consequence: After the failure condition, the Host controller stops sending the SOF. This causes the connected device to go into suspend state.

Problem Fix/Workaround

This problem can be avoided if the system can guarantee that no buffer underrun occurs during the transfer.

44.1.12.3 UHP: Remote Wakeup Event

Conditions:

When a Remote Wakeup event occurs on a downstream port, the OHCI Host controller begins sending resume signaling to the device. The Host controller is supposed to send this resume signaling for 20 ms. However, if the driver sets the HcControl.HCFS into USBOPERATIONAL state during the resume event, then the Host controller terminates sending the resume signal with an EOP to the device.

Consequence: If the Device does not recognize the resume (< 20 ms) event then the Device remains in suspend state.

Problem Fix/Workaround

Host stack can do a port resume after it sets the HcControl.HCFS to USBOPERATIONAL.

44.1.13 USART

44.1.13.1 USART: TXD Signal is floating in Modem and Hardware Handshaking mode.

TXD signal should be pulled up in Modem and Hardware Handshaking mode.

Problem Fix/Workaround

TXD is multiplexed with PIO which integrates a pull up resistor. This internal pull-up must be enabled.

44.1.13.2 USART: DCD is Active High instead of Low

The DCD signal is active at High level in the USART Modem Mode .

DCD should be active at Low level.

Problem Fix/Workaround

Add an inverter.

44.1.14 Power Management Controller (PMC)

44.1.14.1 PMC: PMC bad frequency after MDIV switching

If MDIV and another field (CSS or PRES) are changed at the same, the clock frequency may not be correct.

Problem Fix/Workaround

For each clock switching, the user must take care to:

- change fields CSS, MDIV, PRES one by one
- wait MCKRDY bit setting in PMC_SR before changing PMC_MCKR
- ensure each transitory frequency value is in operational range for PCK and MCK