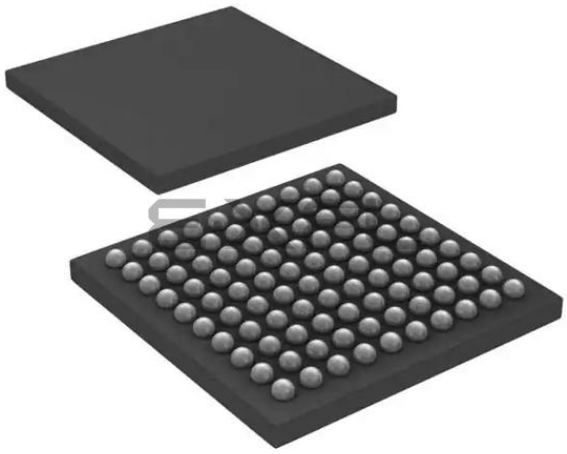


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M7
Core Size	32-Bit Single-Core
Speed	300MHz
Connectivity	I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	75
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	1.08V ~ 3.6V
Data Converters	A/D 10x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-VFBGA
Supplier Device Package	100-VFBGA (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsams70n20b-cfn

Note: Refer to the “Active Mode” section in the Power Considerations chapter for restrictions on the voltage range of analog cells.

LQFP Pin	QFN Pin (11)	Power Rail	I/O Type	Primary		Alternate		PIO Peripheral A		PIO Peripheral B		PIO Peripheral CDir		PIO Peripheral DDir		Reset State
				Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal	Dir	Signal, Dir, PU, PD, HiZ, ST
15	15	VDDIO	CLOCK	PA7	I/O	XIN32(3)	I	–	–	PWMC0_PWMH3	–	–	–	–	–	PIO, HiZ
16	16	VDDIO	CLOCK	PA8	I/O	XOUT32(3)	O	PWMC1_PWMH3	O	AFE0_ADTRG	I	–	–	–	–	PIO, HiZ
33	33	VDDIO	GPIO_AD	PA9	I/O	WKUP6/PIODC3(2)	I	URXD0	I	ISI_D3	I	PWMC0_PWM FI0	I	–	–	PIO, I, PU, ST
28	28	VDDIO	GPIO_AD	PA10	I/O	PIODC4(1)	I	UTXD0	O	PWMC0_PWMEXTRG0	I	RD	I	–	–	PIO, I, PU, ST
27	27	VDDIO	GPIO_AD	PA11	I/O	WKUP7/PIODC5(2)	I	QCS	O	PWMC0_PWMH0	O	PWMC1_PWM L0	O	–	–	PIO, I, PU, ST
29	29	VDDIO	GPIO_AD	PA12	I/O	PIODC6(1)	I	QIO1	I/O	PWMC0_PWMH1	O	PWMC1_PWM H0	O	–	–	PIO, I, PU, ST
18	18	VDDIO	GPIO_AD	PA13	I/O	PIODC7(1)	I	QIO0	I/O	PWMC0_PWMH2	O	PWMC1_PWM L1	O	–	–	PIO, I, PU, ST
19	19	VDDIO	GPIO_CLK	PA14	I/O	WKUP8/PIODCEN1(2)	I	QSCK	O	PWMC0_PWMH3	O	PWMC1_PWM H1	O	–	–	PIO, I, PU, ST
12	12	VDDIO	GPIO_AD	PA21	I/O	AFE0_AD1/PIODCEN2(7)	I	RXD1	I	PCK1	O	PWMC1_PWM FI0	I	–	–	PIO, I, PU, ST
17	17	VDDIO	GPIO_AD	PA22	I/O	PIODCCLK(1)	I	RK	I/O	PWMC0_PWMEXTRG1	I	NCS2	O	–	–	PIO, I, PU, ST
23	23	VDDIO	GPIO_AD	PA24	I/O	–	–	RTS1	O	PWMC0_PWMH1	O	A20	O	ISI_PCK	I	PIO, I, PU, ST
30	30	VDDIO	GPIO_AD	PA27	I/O	–	–	DTR1	O	TIOB2	I/O	MCDA3	I/O	ISI_D7	I	PIO, I, PU, ST
8	8	VDDIO	GPIO	PB0	I/O	AFE0_AD10/RTCOUT0(6)	I	PWMC0_PWMH0	O	–	–	RXD0	I	TF	I/O	PIO, I, PU, ST
7	7	VDDIO	GPIO	PB1	I/O	AFE1_AD0/RTCOUT1(6)	I	PWMC0_PWMH1	O	GTSUCOMP	O	TXD0	I/O	TK	I/O	PIO, I, PU, ST
9	9	VDDIO	GPIO	PB2	I/O	AFE0_AD5(4)	I	CANTX0	O	–	–	CTS0	I	SPI0_NPCS0	I/O	PIO, I, PU, ST
11	11	VDDIO	GPIO_AD	PB3	I/O	AFE0_AD2/WKUP12(6)	I	CANRX0	I	PCK2	O	RTS0	O	ISI_D2	I	PIO, I, PU, ST
46	46	VDDIO	GPIO_MLB	PB4	I/O	TDI(8)	I	TWD1	I/O	PWMC0_PWMH2	O	MLBCLK-	I-	TXD1	I/O	PIO, I, PD, ST
47	47	VDDIO	GPIO_MLB	PB5	I/O	TDO/TRACESWO/WKUP13(8)	O	TWCK1	O	PWMC0_PWML0	O	MLBDAT-	I/O-	TD	O	O, PU
35	35	VDDIO	GPIO	PB6	I/O	SWDIO/TMS(8)	I	–	–	–	–	–	–	–	–	PIO,I,ST
39	39	VDDIO	GPIO	PB7	I/O	SWCLK/TCK(8)	I	–	–	–	–	–	–	–	–	PIO,I,ST
62	63	VDDIO	CLOCK	PB8	I/O	XOUT(9)	O	–	–	–	–	–	–	–	–	PIO, HiZ
63	64	VDDIO	CLOCK	PB9	I/O	XIN(9)	I	–	–	–	–	–	–	–	–	PIO, HiZ
38	38	VDDIO	GPIO	PB12	I/O	ERASE(8)	I	PWMC0_PWML1	O	GTSUCOMP	O	–	–	PCK0	O	PIO, I, PD, ST
1	2	VDDIO	GPIO_AD	PD0	I/O	DAC1(11)	I	GTXCK	I	PWMC1_PWML0	O	SPI1_NPCS1	I/O	DCD0	I	PIO, I, PU, ST
57	57	VDDIO	GPIO	PD1	I/O	–	–	GTXEN	O	PWMC1_PWMH0	O	SPI1_NPCS2	I/O	DTR0	O	PIO, I, PU, ST
56	56	VDDIO	GPIO	PD2	I/O	–	–	GTX0	O	PWMC1_PWML1	O	SPI1_NPCS3	I/O	DSR0	I	PIO, I, PU, ST

22.5.2 EEFC Flash Command Register

Name: EEFC_FCR
Offset: 0x04
Property: Write-only

GETD, GLB, GGPB, STUI, SPUI, GCALB, WUS, EUS, STUS, SPUS, EA	Commands requiring no argument, including Erase all command	FARG is meaningless, must be written with 0
ES	Erase sector command	FARG must be written with any page number within the sector to be erased
EPA	Erase pages command	FARG[1:0] defines the number of pages to be erased The start page must be written in FARG[15:2]. FARG[1:0] = 0: Four pages to be erased. FARG[15:2] = Page_Number / 4 FARG[1:0] = 1: Eight pages to be erased. FARG[15:3] = Page_Number / 8, FARG[2]=0 FARG[1:0] = 2: Sixteen pages to be erased. FARG[15:4] = Page_Number / 16, FARG[3:2]=0 FARG[1:0] = 3: Thirty-two pages to be erased. FARG[15:5] = Page_Number / 32, FARG[4:2]=0 Refer to “EEFC_FCR.FARG Field for EPA Command”.
WP, WPL, EWP, EWPL	Programming commands	FARG must be written with the page number to be programmed
SLB, CLB	Lock bit commands	FARG defines the page number to be locked or unlocked
SGPB, CGPB	GPNVM commands	FARG defines the GPNVM number to be programmed

Once all of the previous steps have been completed, the peripheral clocks can be enabled and/or disabled via registers PMC_PCERx and PMC_PCDRx.

31.18 Clock Switching Details

31.18.1 Master Clock Switching Timings

The following two tables, Clock Switching Timings (Worst Case) and Clock Switching Timings Between Two PLLs (Worst Case) give the worst case timings required for MCK to switch from one selected clock to another one. This is in the event that the prescaler is deactivated. When the prescaler is activated, an additional time of 64 clock cycles of the newly selected clock has to be added.

Table 31-2. Clock Switching Timings (Worst Case)

From	MAINCK	SLCK	PLL Clock
To			
MAINCK	–	4 x SLCK + 2.5 x MAINCK	3 x PLL Clock + 4 x SLCK + 1 x MAINCK
SLCK	0.5 x MAINCK + 4.5 x SLCK	–	3 x PLL Clock + 5 x SLCK
PLL Clock	0.5 x MAINCK + 4 x SLCK + PLLCOUNT x SLCK + 2.5 x PLL Clock	2.5 x PLL Clock + 5 x SLCK + PLLCOUNT x SLCK	See the following table.

Note:

1. PLL designates any available PLL of the Clock Generator.
2. PLLCOUNT designates either PLLACOUNT or UPLLCOUNT.

Table 31-3. Clock Switching Timings Between Two PLLs (Worst Case)

	From	PLLACK	UPLL Clock
To			
PLLACK		–	3 x PLLACK + 4 x SLCK + 1.5 x PLLACK
UPLLCKDIV		3 x UPLLCKDIV + 4 x SLCK + 1.5 x UPLLCKDIV	–

31.20.23 PMC Peripheral Clock Enable Register 1

Name: PMC_PCER1
Offset: 0x0100
Property: Write-only

This register can only be written if the WPEN bit is cleared in the PMC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
		PID62		PID60	PID59	PID58	PID57	PID56
Access								
Reset								
Bit	23	22	21	20	19	18	17	16
			PID53	PID52	PID51	PID50	PID49	PID48
Access								
Reset								
Bit	15	14	13	12	11	10	9	8
	PID47	PID46	PID45	PID44	PID43	PID42	PID41	PID40
Access								
Reset								
Bit	7	6	5	4	3	2	1	0
	PID39		PID37		PID35	PID34	PID33	PID32
Access								
Reset								

Bits 0:3,5,7:28,30 – PID Peripheral Clock x Disable

Value	Description
0	No effect.
1	Disables the corresponding peripheral clock.

Table 34-3. SDRAM Configuration Mapping: 4K Rows, 256/512/1024/2048 Columns

CPU Address Line																										
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
					Bk[1:0]		Row[11:0]												Column[7:0]								M0
				Bk[1:0]			Row[11:0]												Column[8:0]								M0
			Bk[1:0]				Row[11:0]												Column[9:0]								M0
		Bk[1:0]					Row[11:0]												Column[10:0]								M0

Note: M0 is the byte address inside a 16-bit half-word and Bk[1] = BA1, Bk[0] = BA0.

Table 34-4. SDRAM Configuration Mapping: 8K Rows, 256/512/1024/2048 Columns

CPU Address Line																										
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
				Bk[1:0]		Row[12:0]													Column[7:0]								M0
			Bk[1:0]			Row[12:0]													Column[8:0]								M0
		Bk[1:0]				Row[12:0]													Column[9:0]								M0
	Bk[1:0]					Row[12:0]													Column[10:0]								M0

Note: M0 is the byte address inside a 16-bit half-word and Bk[1] = BA1, Bk[0] = BA0.

34.5 Product Dependencies

34.5.1 SDRAM Device Initialization

The initialization sequence is generated by software. The sequence to initialize SDRAM devices is the following:

1. Set the SDRAM features in the SDRAMC_CR: asynchronous timings (TRC, TRAS, etc.), number of columns, number of rows, CAS latency and data bus width. Set UNAL bit in SDRAMC_CFR1.
2. For mobile SDRAM, configure temperature-compensated self-refresh (TCSR), drive strength (DS) and partial array self-refresh (PASR) in the Low Power register (SDRAMC_LPR).
3. Select the SDRAM memory device type in the Memory Device register (SDRAMC_MDR).
4. A pause of at least 200 µs must be observed before a signal toggle.
5. A NOP command is issued to the SDRAM devices. The application must write a 1 to the MODE field in the Mode register (SDRAMC_MR) (see **Note**). Read the SDRAMC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any SDRAM address.
6. An All Banks Precharge command is issued to the SDRAM. The application must write a 2 to the MODE field in the SDRAMC_MR. Read the SDRAMC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any SDRAM address.
7. Eight autorefresh (CBR) cycles are provided. The application must set the MODE field to 4 in the SDRAMC_MR. Read the SDRAMC_MR and add a memory barrier assembler instruction just after the read. Perform a write access to any SDRAM location eight times.

34.7.3 SDRAMC Configuration Register

Name: SDRAMC_CR
Offset: 0x08
Reset: 0x852372C0
Property: Read/Write

WARNING Bit 7 (DBW) must always be set when programming the SDRAMC_CR.

Bit	31	30	29	28	27	26	25	24
	TXSR[3:0]				TRAS[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	1	0	1

Bit	23	22	21	20	19	18	17	16
	TRCD[3:0]				TRP[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	1	1

Bit	15	14	13	12	11	10	9	8
	TRC_TRFC[3:0]				TWR[3:0]			
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	1	1	1	0	0	1	0

Bit	7	6	5	4	3	2	1	0
	DBW	CAS[1:0]		NB	NR[1:0]		NC[1:0]	
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	0	0	0	0	0	0

Bits 31:28 – TXSR[3:0] Exit Self-Refresh to Active Delay
Reset value is eight cycles.

This field defines the delay between SCKE set high and an Activate Command in number of cycles. Number of cycles is between 0 and 15.

Bits 27:24 – TRAS[3:0] Active to Precharge Delay
Reset value is five cycles.

This field defines the delay between an Activate Command and a Precharge Command in number of cycles. Number of cycles is between 0 and 15.

Bits 23:20 – TRCD[3:0] Row to Column Delay
Reset value is two cycles.

This field defines the delay between an Activate Command and a Read/Write Command in number of cycles. Number of cycles is between 0 and 15.

Bits 19:16 – TRP[3:0] Row Precharge Delay
Reset value is three cycles.

Bits 23:0 – UBLEN[23:0] Microblock Length
This field indicates the number of data in the microblock. The microblock contains UBLEN data.

36.7 XDMAC Maintenance Software Operations

36.7.1 Disabling a Channel

A disable channel request occurs when a write operation is performed in the XDMAC_GD register. If the channel is source peripheral synchronized (bit XDMAC_CCx.TYPE is set and bit XDMAC_CCx.DSYNC is cleared), then pending bytes (bytes located in the FIFO) are written to memory and bit XDMAC_CISx.DIS is set. If the channel is not source peripheral synchronized, the current channel transaction (read or write) is terminated and XDMAC_CISx.DIS is set. XDMAC_GS.STx is cleared by hardware when the current transfer is completed. The channel is no longer active and can be reused.

36.7.2 Suspending a Channel

A disable channel request occurs when a write operation is performed in the XDMAC_GD register. If the channel is source peripheral synchronized (bit XDMAC_CCx.TYPE is set and bit XDMAC_CCx.DSYNC is cleared), then pending bytes (bytes located in the FIFO) are written to memory and bit XDMAC_CISx.DIS is set. If the channel is not source peripheral synchronized, the current channel transaction (read or write) is terminated and XDMAC_CISx.DIS is set. XDMAC_GS.STx is cleared by hardware when the current transfer is completed. The channel is no longer active and can be reused.

36.7.3 Flushing a Channel

A FIFO flush command is issued by writing to the XDMAC_SWF register. The content of the FIFO is written to memory. XDMAC_CISx.FIS (End of Flush Interrupt Status bit) is set when the last byte is successfully transferred to memory. The channel is not disabled. The flush operation is not blocking, meaning that read operation can be scheduled during the flush write operation. The flush operation is only relevant for peripheral to memory transfer where pending peripheral bytes are buffered into the channel FIFO.

36.7.4 Maintenance Operation Priority

36.7.4.1 Disable Operation Priority

- When a disable request occurs on a suspended channel, the XDMAC_GWS.WSx (Channel x Write Suspend bit) is cleared. If the transfer is source peripheral synchronized, the pending bytes are drained to memory. The bit XDMAC_CISx.DIS is set.
- When a disable request follows a flush request, if the flush last transaction is not yet scheduled, the flush request is discarded and the disable procedure is applied. Bit XDMAC_CISx.FIS is not set. Bit XDMAC_CISx.DIS is set when the disable request is completed. If the flush request transaction is already scheduled, the XDMAC_CISx.FIS is set. XDMAC_CISx.DIS is also set when the disable request is completed.

36.7.4.2 Flush Operation Priority

- When a flush request occurs on a suspended channel, if there are pending bytes in the FIFO, they are written out to memory, XDMAC_CISx.FIS is set. If the FIFO is empty, XDMAC_CISx.FIS is also set.
- If the flush operation is performed after a disable request, the flush command is ignored. XDMAC_CISx.FIS is not set.

36.7.4.3 Suspend Operation Priority

If the suspend operation is performed after a disable request, the write suspend operation is ignored.

OUTPUT	INPUT	352 × 288	640 × 480	800 × 600	1280 × 1024	1600 × 1200	2048 × 1536
320 × 240							
CIF 352 × 288	F	16	26	33	56	66	85
QCIF 176 × 144	F	32	53	66	113	133	170

Example:

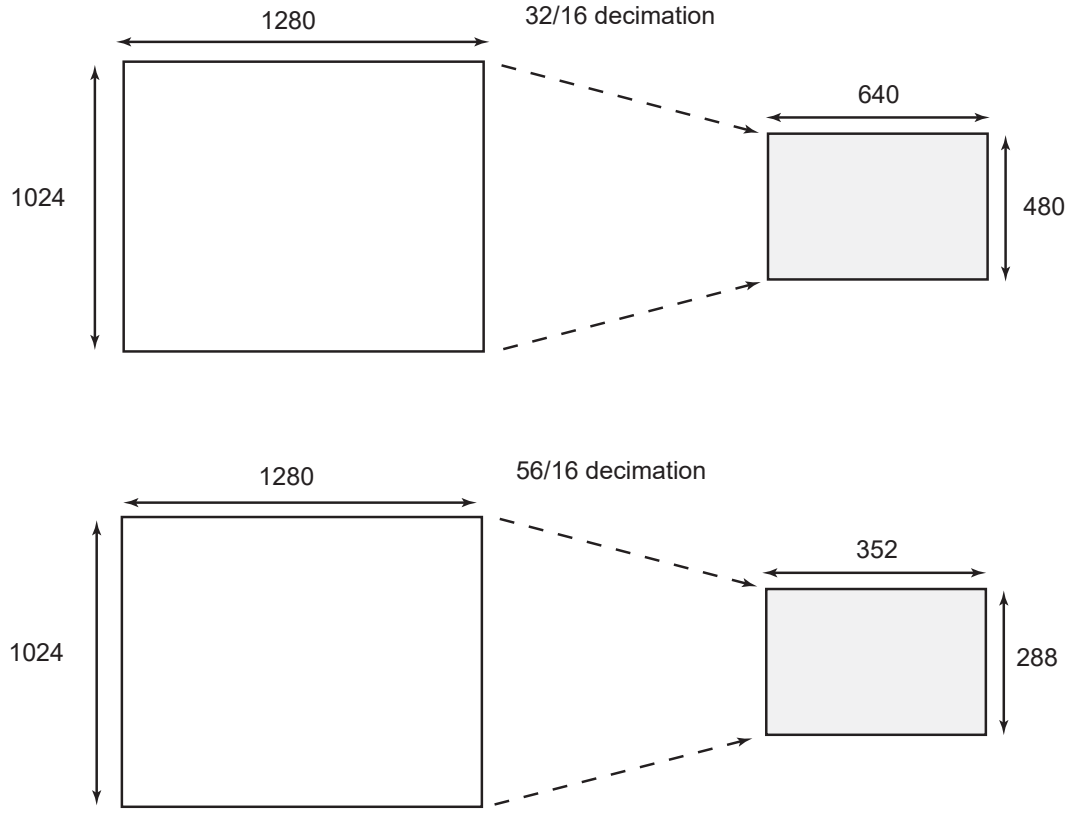
Input 1280 × 1024 Output = 640 × 480

Hratio = 1280/640 = 2

Vratio = 1024/480 = 2.1333

The decimation factor is 2 so 32/16.

Figure 37-5. Resize Examples



37.5.4.2 Color Space Conversion

This module converts YCrCb or YUV pixels to RGB color space. Clipping is performed to ensure that the samples value do not exceed the allowable range. The conversion matrix is defined below and is fully programmable:

To accommodate the status and statistics associated with each frame, three words per packet (or two if configured in 64-bit datapath mode) are reserved at the end of the packet data. If the packet is bad and requires to be dropped, the status and statistics are the only information held on that packet.

The receiver packet buffer will also detect a full condition so that an overflow condition can be detected. If this occurs, subsequent packets are dropped and an RX overflow interrupt is raised.

For full store and forward, the DMA only begins packet fetches once the status and statistics for a frame are available. If the frame has a bad status due to a frame error, the status and statistics are passed on to the GMAC registers. If the frame has a good status, the information is used to read the frame from the packet buffer memory and burst onto the AHB using the DMA buffer management protocol. Once the last frame data has been transferred to the packet buffer, the status and statistics are updated to the GMAC registers.

If Partial Store and Forward mode is active, the DMA will begin fetching the packet data before the status is available. As soon as the status becomes available, the DMA will fetch this information as soon as possible before continuing to fetch the remainder of the frame. Once the last frame data has been transferred to the packet buffer, the status and statistics are updated to the GMAC registers.

38.6.3.9 Priority Queuing in the DMA

The DMA by default uses a single transmit and receive queue. This means the list of transmit/receive buffer descriptors point to data buffers associated with a single transmit/receive data stream. The GMAC can select up to 6 priority queues. Each queue has an independent list of buffer descriptors pointing to separate data streams.

The table below gives the DPRAM size associated with each queue.

Table 38-4. Queue Size

Queue Number	Queue Size
5 (highest priority)	1 KB
4	2 KB
3	2 KB
2	512 bytes
1	512 bytes
0 (lowest priority)	2 KB

In the transmit direction, higher priority queues are always serviced before lower priority queues, with Q0 as lowest priority and Q5 as highest priority. This strict priority scheme requires the user to ensure that high priority traffic is constrained so that lower priority traffic will have required bandwidth. The GMAC DMA will determine the next queue to service by initiating a sequence of buffer descriptor reads interrogating the ownership bits of each. The buffer descriptor corresponding to the highest priority queue is read first.

As an example, if the ownership bit of this descriptor is set, the DMA will progress by reading the 2nd highest priority queue’s descriptor. If that ownership bit read of this lower priority queue is set as well, the DMA will read the 3rd highest priority queue’s descriptor. If all the descriptors return an ownership bit set, a resource error has occurred, so an interrupt is generated and transmission is automatically halted. Transmission can only be restarted by writing a '1' to the Transmission Start bit in the Network Control register (GMAC_NCR.TSTART). The GMAC DMA will need to identify the highest available queue to

Writing a '0' to this bit disables the Management Port, and forces MDIO to high impedance state and MDC to low impedance.

Value	Description
0	Management Port is disabled.
1	Management Port is enabled.

Bit 3 – TXEN Transmit Enable
Writing a '1' to this bit enables the GMAC transmitter to send data.

Writing a '0' to this bit stops transmission immediately, the transmit pipeline and control registers is cleared, and the Transmit Queue Pointer Register will be set to point to the start of the transmit descriptor list.

Value	Description
0	Transmit is disabled.
1	Transmit is enabled.

Bit 2 – RXEN Receive Enable
Writing a '1' to this bit enables the GMAC to receive data.

Writing a '0' to this bit stops frame reception immediately, and the receive pipeline is cleared. The Receive Queue Pointer Register is not affected.

Value	Description
0	Receive is disabled.
1	Receive is enabled.

Bit 1 – LBL Loop Back Local
Writing '1' to this bit connects GTX to GRX, GTXEN to GRXDV, and forces full duplex mode.

GRXCK and GTXCK may malfunction as the GMAC is switched into and out of internal loop back. It is important that receive and transmit circuits have already been disabled when making the switch into and out of internal loop back.

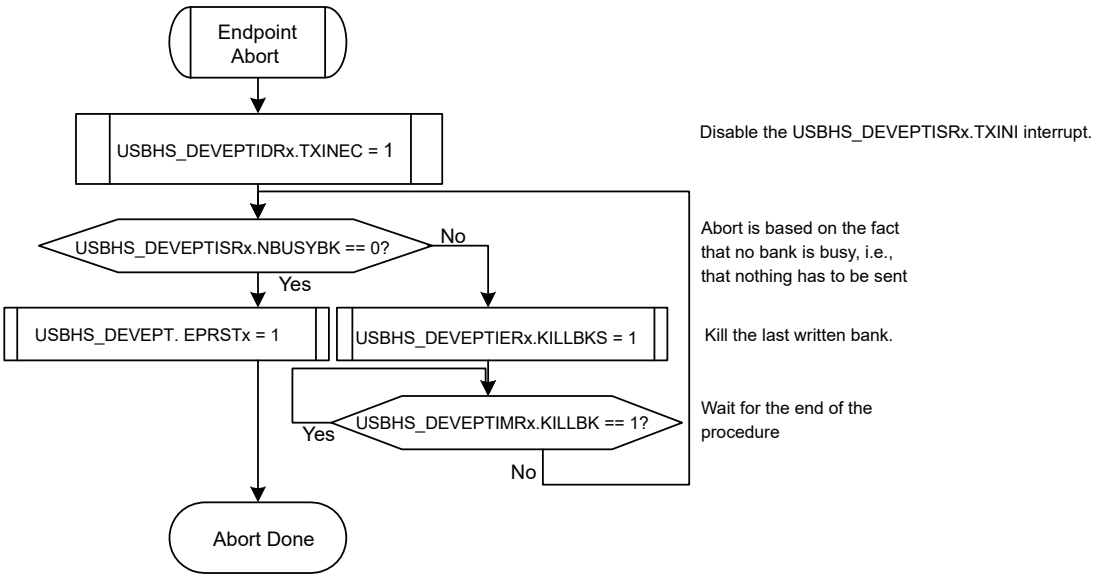
Value	Description
0	Loop back local is disabled.
1	Loop back local is enabled.

- The user writes the data into the current bank by using the USB Pipe/Endpoint nFIFO Data (USBFIFOnDATA) register, until all the data frame is written or the bank is full (in which case USBHS_DEVEPTISRx.RWALL is cleared and the Byte Count (USBHS_DEVEPTISRx.BYCT) field reaches the endpoint size).
- The user allows the controller to send the bank and switches to the next bank (if any) by clearing USBHS_DEVEPTIMRx.FIFOCON.

If the endpoint uses several banks, the current one can be written while the previous one is being read by the host. Then, when the user clears USBHS_DEVEPTIMRx.FIFOCON, the following bank may already be free and USBHS_DEVEPTISRx.TXINI is set immediately.

An “Abort” stage can be produced when a zero-length OUT packet is received during an IN stage of a control or isochronous IN transaction. The Kill IN Bank (USBHS_DEVEPTIMRx.KILLBK) bit is used to kill the last written bank. The best way to manage this abort is to apply the algorithm represented in the following figure.

Figure 39-13. Abort Algorithm



39.5.2.13 Management of OUT Endpoints

Overview

OUT packets are sent by the host. All data which acknowledges or not the bank can be read when it is empty.

The endpoint must be configured first.

The USBHS_DEVEPTISRx.RXOUTI bit is set at the same time as USBHS_DEVEPTIMRx.FIFOCON when the current bank is full. This triggers a PEP_x interrupt if the Received OUT Data Interrupt Enable (USBHS_DEVEPTIMRx.RXOUTE) bit is one.

USBHS_DEVEPTISRx.RXOUTI is cleared by software (by writing a one to the Received OUT Data Interrupt Clear (USBHS_DEVEPTICRx.RXOUTIC) bit to acknowledge the interrupt, which has no effect on the endpoint FIFO.

The user then reads from the FIFO and clears the USBHS_DEVEPTIMRx.FIFOCON bit to free the bank. If the OUT endpoint is composed of multiple banks, this also switches to the next bank. The

Value	Description
0	Cleared when USBHS_HSTPIPICR.RXINIC = 1.
1	Set when a new USB message is stored in the current bank of the pipe. This triggers an interrupt if USBHS_HSTPIPIMR.RXINE = 1.

Value	Description
0	Cleared when USBHS_HSTPIPIDR.OVERFIEC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.OVERFIE).
1	Set when USBHS_HSTPIPIER.OVERFIES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.OVERFIE).

Bit 4 – NAKEDE NAKed Interrupt Enable

Value	Description
0	Cleared when USBHS_HSTPIPIDR.NAKEDEC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.NAKEDE).
1	Set when USBHS_HSTPIPIER.NAKEDES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.NAKEDE).

Bit 3 – PERRE Pipe Error Interrupt Enable

Value	Description
0	Cleared when USBHS_HSTPIPIDR.PERREC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.PERRE).
1	Set when USBHS_HSTPIPIER.PERRES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.PERRE).

Bit 2 – UNDERFIE Underflow Interrupt Enable

Value	Description
0	Cleared when USBHS_HSTPIPIDR.UNDERFIEC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.UNDERFIE).
1	Set when USBHS_HSTPIPIER.UNDERFIES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.UNDERFIE).

Bit 1 – TXOUTE Transmitted OUT Data Interrupt Enable

Value	Description
0	Cleared when USBHS_HSTPIPIDR.TXOUTEC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.TXOUTE).
1	Set when USBHS_HSTPIPIER.TXOUTES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.TXOUTE).

Bit 0 – RXINE Received IN Data Interrupt Enable

Value	Description
0	Cleared when USBHS_HSTPIPIDR.RXINEC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.RXINE).
1	Set when USBHS_HSTPIPIER.RXINES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.RXINE).

43.7.15 TWIHS Write Protection Mode Register

Name: TWIHS_WPMR
Offset: 0xE4
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	WPKEY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	23	22	21	20	19	18	17	16
	WPKEY[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8
	WPKEY[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0
Bit	7	6	5	4	3	2	1	0
								WPEN
Access								R/W
Reset								0

Bits 31:8 – WPKEY[23:0] Write Protection Key

Value	Name	Description
0x545749	PASSWD	Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0.

Bit 0 – WPEN Write Protection Enable
See Register Write Protection for the list of registers that can be write-protected.

Value	Description
0	Disables the write protection if WPKEY corresponds to 0x545749 (“TWI” in ASCII).
1	Enables the write protection if WPKEY corresponds to 0x545749 (“TWI” in ASCII).

Figure 48-22. Single-packet Asynchronous or Control System Memory Structure

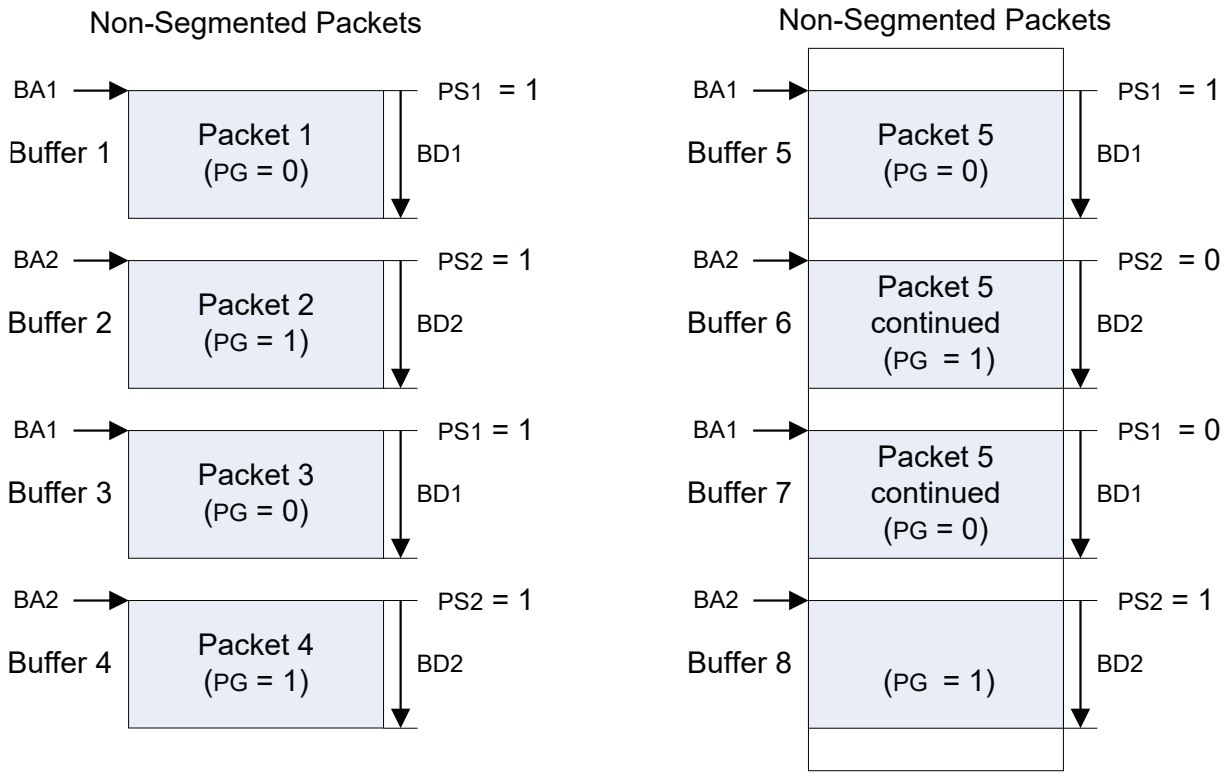


Table 48-23. Single-packet Asynchronous and Control Entry Format

Bit Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	CE	LE	PG	Reserved												
16	Reserved															
32	RDY1	DNE1	ERR1	PS1	MEP1	BD1[10:0]										
48	RDY2	DNE2	ERR2	PS2	MEP2	BD2[10:0]										
64	BA1[15:0]															
80	BA1[31:16]															
96	BA2[15:0]															
112	BA2[31:16]															

Multiple-packet Mode

The multiple-packet mode asynchronous and control buffering scheme supports more than one packet per system memory buffer, as shown in the following figure. Multiple- packet mode reduces the interrupt rate for packet channels at the cost of increasing buffering and latency.

For Tx packet channels in multiple-packet mode, software sets the packet start bit (PSn) for every buffer. Setting PSn informs hardware that the first two bytes of the buffer contains the port message length (PML) of the first packet. After the first packet, hardware keeps track of where packets start and end within the current buffer. Software should not write to PSn while the buffer is active (RDYn = 1 and DNEn = 0). For Tx packet channels, the buffer is done (DNEn= 1) when the last byte of the last packet in the

not fully deactivated while no conversion is requested, thereby providing lower power savings but faster wakeup.

The conversion sequencer allows automatic processing with minimum processor intervention and optimized power consumption. Conversion sequences are performed periodically using a Timer/Counter output or the PWM event line.

The DMA can automatically process the periodic acquisition of several samples without processor intervention.

The sequence can be customized by programming the Channel Sequence registers AFEC_SEQ1R and AFEC_SEQ2R and setting AFEC_MR.USEQ. The user selects a specific order of channels and can program up to 12 conversions by sequence. The user may create a personal sequence by writing channel numbers in AFEC_SEQ1R and AFEC_SEQ2R. Channel numbers can be written in any order and repeated several times. Only enabled USCHx fields are converted. Thus, to program a 15-conversion sequence, the user disables AFEC_CHSR.CH15, thus disabling AFEC_SEQ2R.USCH15.

Note: The reference voltage pins always remain connected in Normal mode as in Sleep mode.

Related Links

58. Electrical Characteristics for SAM V70/V71

52.6.8 Comparison Window

The AFEC features automatic comparison functions. It compares converted values to a low threshold, a high threshold or both, depending on the value of AFEC_EMR.CMPMODE. The comparison can be done on all channels or only on the channel specified in AFEC_EMR.CMPSEL. To compare all channels, AFEC_EMR.CMPALL must be set.

Moreover, a filtering option can be set by writing the number of consecutive comparison errors needed to raise the flag. This number can be written and read in AFEC_EMR.CMPFILTER.

The flag can be read on AFEC_ISR.COMPE and can trigger an interrupt.

The high threshold and the low threshold can be read/written in the Compare Window Register (AFEC_CWR).

Depending on the sign of the conversion, chosen by setting the SIGNMODE bit in the AFEC Extended Mode Register, the high threshold and low threshold values must be signed or unsigned to maintain consistency during the comparison. If the conversion is signed, both thresholds must also be signed; if the conversion is unsigned, both thresholds must be unsigned. If comparison occurs on all channels, the SIGNMODE bit must be set to ALL_UNSIGNED or ALL_SIGNED and thresholds must be set accordingly.

52.6.9 Differential Inputs

The AFE can be used either as a single-ended AFE (AFEC_DIFFR.DIFF = 0) or as a fully differential AFE (AFEC_DIFFR.DIFF = 1). By default, after a reset, the AFE is in Single-ended mode.

The AFEC can apply a different mode on each channel.

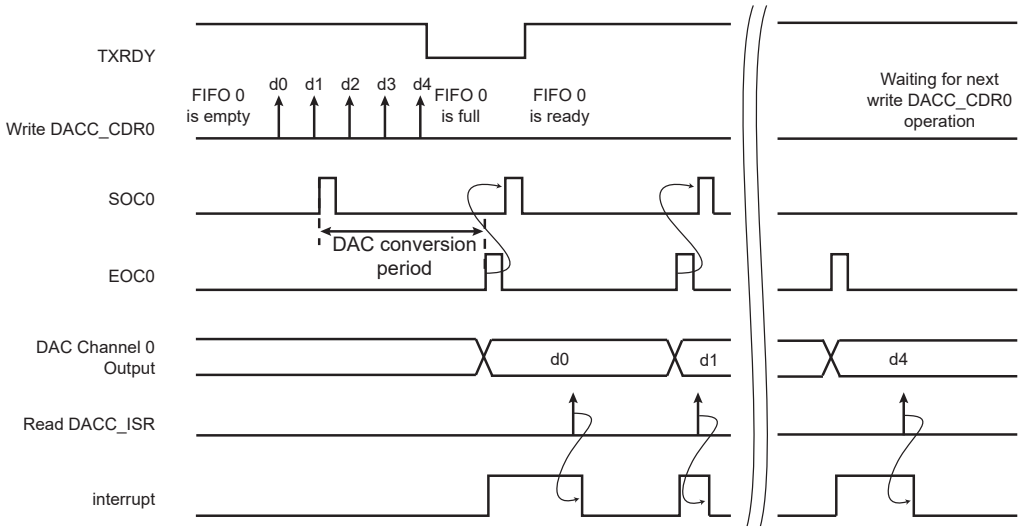
The same inputs are used in Single-ended or Differential mode.

Depending on the AFE mode, the analog multiplexer selects one or two inputs to map to a channel. The table below provides input mapping for both modes.

the corresponding analog output. The next data is converted only when the EOC of the previous data is set.

If the FIFO is emptied, no conversion occurs and the data is maintained at the output of the DAC.

Figure 53-3. Conversion Sequence in Free-running Mode



53.6.4.3 Max Speed Mode

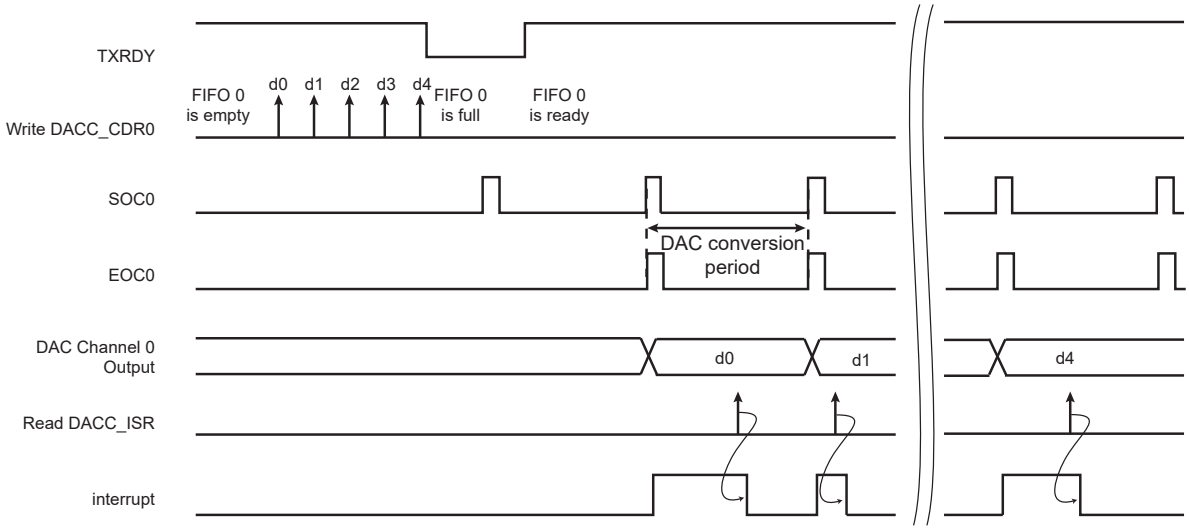
Max speed mode is enabled by setting DACC_TRIGR.TRGENx and DACC_MR.MAXSx.

The conversion rate is forced by the controller, which starts one conversion every 12 DAC clock periods. The controller does not wait for the EOC of the previous data to send a new data to the DAC and the DAC is always clocked.

If the FIFO is emptied, the controller send the last converted data to the DAC at a rate of 12 DAC clock periods.

The DACC_ACR.IBCTLCHx field must be configured for 1 MSps (see the section “Electrical Characteristics”).

Figure 53-4. Conversion Sequence in Max Speed Mode



55.6 Register Summary

Offset	Name	Bit Pos.								
0x00	ICM_CFG	7:0	BBC[3:0]					SLBDIS	EOMDIS	WBDIS
		15:8	UALGO[2:0]			UIHASH			DUALBUFF	ASCD
		23:16								
		31:24								
0x04	ICM_CTRL	7:0	REHASH[3:0]					SWRST	DISABLE	ENABLE
		15:8	RMEN[3:0]				RMDIS[3:0]			
		23:16								
		31:24								
0x08	ICM_SR	7:0								ENABLE
		15:8	RMDIS[3:0]				RAWRMDIS[3:0]			
		23:16								
		31:24								
0x0C ... 0x0F	Reserved									
0x10	ICM_IER	7:0	RDM[3:0]				RHC[3:0]			
		15:8	RWC[3:0]				RBE[3:0]			
		23:16	RSU[3:0]				REC[3:0]			
		31:24								URAD
0x14	ICM_IDR	7:0	RDM[3:0]				RHC[3:0]			
		15:8	RWC[3:0]				RBE[3:0]			
		23:16	RSU[3:0]				REC[3:0]			
		31:24								URAD
0x18	ICM_IMR	7:0	RDM[3:0]				RHC[3:0]			
		15:8	RWC[3:0]				RBE[3:0]			
		23:16	RSU[3:0]				REC[3:0]			
		31:24								URAD
0x1C	ICM_ISR	7:0	RDM[3:0]				RHC[3:0]			
		15:8	RWC[3:0]				RBE[3:0]			
		23:16	RSU[3:0]				REC[3:0]			
		31:24								URAD
0x20	ICM_UASR	7:0						URAT[2:0]		
		15:8								
		23:16								
		31:24								
0x24 ... 0x2F	Reserved									
0x30	ICM_DSCR	7:0	DASA[1:0]							
		15:8	DASA[9:2]							
		23:16	DASA[17:10]							
		31:24	DASA[25:18]							
0x34	ICM_HASH	7:0	HASA[0:0]							

55.6.4 ICM Interrupt Enable Register

Name: ICM_IER
Offset: 0x10
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
								URAD
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
	RSU[3:0]				REC[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	–	0	0	0	–

Bit	15	14	13	12	11	10	9	8
	RWC[3:0]				RBE[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	–	0	0	0	–

Bit	7	6	5	4	3	2	1	0
	RDM[3:0]				RHC[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	–	0	0	0	–

Bit 24 – URAD Undefined Register Access Detection Interrupt Enable

Value	Description
0	No effect.
1	The Undefined Register Access interrupt is enabled.

Bits 23:20 – RSU[3:0] Region Status Updated Interrupt Disable

Value	Description
0	No effect.
1	When RSU[i] is set to one, the region i Status Updated interrupt is enabled.

Bits 19:16 – REC[3:0] Region End bit Condition Detected Interrupt Enable

Value	Description
0	No effect.
1	When REC[i] is set to one, the region i End bit Condition interrupt is enabled.

Bits 15:12 – RWC[3:0] Region Wrap Condition detected Interrupt Enable

Value	Description
0	No effect.
1	When RWC[i] is set to one, the Region i Wrap Condition interrupt is enabled.