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	CANbus, EBI/EMI, Ethernet, I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, SSC, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	114
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 24x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamv71q20b-aab

Function	Application	Description	Event Source	Event Destination
	General-purpose	Temperature sensor Low-speed measurement (see **Notes 10, 11**)	RTC RTCOUT0	AFEC0 and AFEC1
Conversion trigger	General-purpose	Trigger source selection in DACC (Digital-to-Analog Converter Controller) (see **Note 13**)	TC0 TIOA0, TIOA1, TIOA2	DACC
			PIO DATRG	DACC
			PWM0 Event Line 0 and 1(14)	DACC
			PWM1 Event Line 0 and 1(14)	DACC
Image capture	Low-cost image sensor	Direct image transfer from sensor to system memory via DMA(15)	PIO PA3/4/5/9/10/11/12/13, PA22, PA14, PA21	DMA
Delay measurement	Motor control	Propagation delay of external components (IOs, power transistor bridge driver, etc.) See **Notes 16, 17**)	PWM0 Comparator Output OC0	TC0 TIOA0 and TIOB0
			PWM0 Comparator Output OC1	TC0 TIOA1 and TIOB1
			PWM0 Comparator Output OC2	TC0 TIOA2 and TIOB2
			PWM1 Comparator Output OC0	TC1 TIOA3 and TIOB3
			PWM1 Comparator Output OC1	TC1 TIOA4 and TIOB4
			PWM1 Comparator Output OC2	TC1 TIOA5 and TIOB5
			PWM0 Comparator Output OC0	TC2 TIOA6 and TIOB6
			PWM0 Comparator Output OC1	TC2 TIOA7 and TIOB7

Instance ID	Instance Name	NVIC Interrupt	PMC Clock Control	Description
45	UART3	X	X	Universal Asynchronous Receiver/ Transmitter
46	UART4	X	X	Universal Asynchronous Receiver/ Transmitter
47	TC2_CHANNEL0	X	X	16-bit Timer Counter 2, Channel 0
48	TC2_CHANNEL1	X	X	16-bit Timer Counter 2, Channel 1
49	TC2_CHANNEL2	X	X	16-bit Timer Counter 2, Channel 2
50	TC3_CHANNEL0	X	X	16-bit Timer Counter 3, Channel 0
51	TC3_CHANNEL1	X	X	16-bit Timer Counter 3, Channel 1
52	TC3_CHANNEL2	X	X	16-bit Timer Counter 3, Channel 2
53	–	–	–	Reserved
54	–	–	–	Reserved
55	–	–	–	Reserved
53	MLB	X	X	MediaLB IRQ 0
54	MLB	X	–	MediaLB IRQ 1
55	–	X	–	Reserved
56	AES	X	X	Advanced Encryption Standard
57	TRNG	X	X	True Random Number Generator
58	XDMAC	X	X	DMA Controller
59	ISI	X	X	Image Sensor Interface
60	PWM1	X	X	Pulse Width Modulation Controller
61	ARM	FPU	–	ARM Floating Point Unit interrupt associated with OFC, UFC, IOC, DZC and IDC bits
62	SDRAMC	X	–	SDRAM Controller
63	RSWDT	X	–	Reinforced Safety Watchdog Timer
64	ARM	CCW	–	ARM Cache ECC Warning
65	ARM	CCF	–	ARM Cache ECC Fault
66	GMAC	Q1	–	GMAC Queue 1 Interrupt signal toggled on a DMA write to the first word of each DMA data buffer associated with queue 1

3. When Flash programming is completed, the bit EEFC_FSR.FRDY rises. If an interrupt has been enabled by setting the bit EEFC_FMR.FRDY, the interrupt line of the EEFC is activated.

Three errors can be detected in EEFC_FSR after a programming sequence:

- Command Error: A bad keyword has been written in EEFC_FCR.
- Lock Error: The page to be programmed belongs to a locked region. A command must be run previously to unlock the corresponding region.
- Flash Error: When programming is completed, the WriteVerify test of the Flash memory has failed.

Only one page can be programmed at a time. It is possible to program all the bits of a page (full page programming) or only some of the bits of the page (partial page programming).

Depending on the number of bits to be programmed within the page, the EEFC adapts the write operations required to program the Flash.

When a ‘Write Page’ (WP) command is issued, the EEFC starts the programming sequence and all the bits written at 0 in the latch buffer are cleared in the Flash memory array.

During programming, i.e., until EEFC_FSR.FDRY rises, access to the Flash is not allowed.

22.4.3.2.1 Full Page Programming

To program a full page, all the bits of the page must be erased before writing the latch buffer and issuing the WP command. The latch buffer must be written in ascending order, starting from the first address of the page. See Figure 22-8.

22.4.3.2.2 Partial Page Programming

To program only part of a page using the WP command, the following constraints must be respected:

- Data to be programmed must be contained in integer multiples of 128-bit address-aligned words.
- 128-bit words can be programmed only if all the corresponding bits in the Flash array are erased (at logical value ‘1’).

See 22.4.3.2.4 Programming Bytes.

22.4.3.2.3 Optimized Partial Page Programming

The EEFC automatically detects the number of 128-bit words to be programmed. If only one 128-bit aligned word is to be programmed in the Flash array, the process is optimized to reduce the time needed for programming.

If several 128-bit words are to be programmed, a standard page programming operation is performed.

See Figure 22-10.

22.4.3.2.4 Programming Bytes

Individual bytes can be programmed using the Partial Page Programming mode.

In this case, an area of 128 bits must be reserved for each byte.

Refer to Figure 22-11

Value	Name	Description
0	NO_TIMEVENT	No time event has occurred since the last clear.
1	TIMEVENT	At least one time event has occurred since the last clear.

Bit 2 – SEC Second Event

Value	Name	Description
0	NO_SECEVENT	No second event has occurred since the last clear.
1	SECEVENT	At least one second event has occurred since the last clear.

Bit 1 – ALARM Alarm Flag

Value	Name	Description
0	NO_ALARMEVENT	No alarm matching condition occurred.
1	ALARMEVENT	An alarm matching condition has occurred.

Bit 0 – ACKUPD Acknowledge for Update

Value	Name	Description
0	FREERUN	Time and calendar registers cannot be updated.
1	UPDATE	Time and calendar registers can be updated.

30. Clock Generator

30.1 Description

The Clock Generator user interface is embedded within the Power Management Controller and is described in Power Management Controller (PMC) User Interface. However, the Clock Generator registers are named CKGR_.

30.2 Embedded Characteristics

The Clock Generator is comprised of the following:

- A low-power 32.768 kHz crystal oscillator with Bypass mode
- A low-power Slow RC oscillator (32 kHz typical)
- A 3 to 20 MHz Main crystal oscillator with Bypass mode
- A Main RC oscillator. Three output frequencies can be selected: 4/8/12 MHz. By default 12 MHz is selected. 8 MHz and 12 MHz are factory-trimmed.
- A 480 MHz UTMI PLL, providing a clock for the USB high-speed controller
- A 160 to 500 MHz programmable PLL (input from 8 to 32 MHz)

It provides the following clocks:

- SLCK — Slow clock. The only permanent clock within the system
- MAINCK — output of the Main clock oscillator selection: either the Main crystal oscillator or Main RC oscillator
- PLLACK — output of the divider and 160 to 500 MHz programmable PLL (PLLA)
- UPLLCK — output of the 480 MHz UTMI PLL (UPLL)

Read CKGR_MCFR until the MAINFRDY field is set, after which the user can read CKGR_MCFR.MAINF by performing an additional read. This provides the number of Main clock cycles that have been counted during a period of 16 SLCK cycles.

If MAINF = 0, switch MAINCK to the Main RC Oscillator by clearing CKGR_MOR.MOSCSEL. If MAINF ≠ 0, proceed to Step 6.

6. Set PLLA and Divider (if not required, proceed to Step 7.):
   All parameters needed to configure PLLA and the divider are located in CKGR_PLLAR.

   CKGR_PLLAR.DIVA is used to control the divider. This parameter can be programmed between 0 and 127. Divider output is divider input divided by DIVA parameter. By default, DIVA field is cleared which means that the divider and PLLA are turned off.

   CKGR_PLLAR.MULA is the PLLA multiplier factor. This parameter can be programmed between 0 and 62. If MULA is cleared, PLLA will be turned off, otherwise the PLLA output frequency is PLLA input frequency multiplied by (MULA + 1).

   CKGR_PLLAR.PLLACOUNT specifies the number of SLCK cycles before PMC_SR.LOCKA is set after CKGR_PLLAR has been written.

   Once CKGR_PLLAR has been written, the user must wait for PMC_SR.LOCKA to be set. This can be done either by polling PMC_SR.LOCKA or by waiting for the interrupt line to be raised if the associated interrupt source (LOCKA) has been enabled in PMC_IER. All fields in CKGR_PLLAR can be programmed in a single write operation. If MULA or DIVA is modified, the LOCKA bit goes low to indicate that PLLA is not yet ready. When PLLA is locked, LOCKA is set again. The user must wait for the LOCKA bit to be set before using the PLLA output clock.
7. Select MCK and HCLK:
   MCK and HCLK are configurable via PMC_MCKR.

   CSS is used to select the clock source of MCK and HCLK. By default, the selected clock source is MAINCK.

   PRES is used to define the HCLK and MCK prescaler.s The user can choose between different values (1, 2, 3, 4, 8, 16, 32, 64). Prescaler output is the selected clock source frequency divided by the PRES value.

   MDIV is used to define the MCK divider. It is possible to choose between different values (0, 1, 2, 3). MCK output is the HCLK frequency divided by 1, 2, 3 or 4, depending on the value programmed in MDIV.

   By default, MDIV is cleared, which indicates that the HCLK is equal to MCK.

   Once the PMC_MCKR has been written, the user must wait for PMC_SR.MCKRDY to be set. This can be done either by polling PMC_SR.MCKRDY or by waiting for the interrupt line to be raised if the associated interrupt source (MCKRDY) has been enabled in PMC_IER. PMC_MCKR must not be programmed in a single write operation. The programming sequence for PMC_MCKR is as follows:

   If a new value for PMC_MCKR.CSS corresponds to any of the available PLL clocks:

   a. Program PMC_MCKR.PRES.

   b. Wait for PMC_SR.MCKRDY to be set.

   c. Program PMC_MCKR.MDIV.

   d. Wait for PMC_SR.MCKRDY to be set.

32.6.1.7 PIO Input Filter Enable Register

Name: PIO_IFER
Offset: 0x0020
Property: Write-only

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

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

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

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

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

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

Value	Description
0	No effect.
1	Enables the input glitch filter on the I/O line.

32.6.1.23 PIO Pull-Up Status Register

Name: PIO_PUSR
Offset: 0x0068
Property: Read-only

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

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

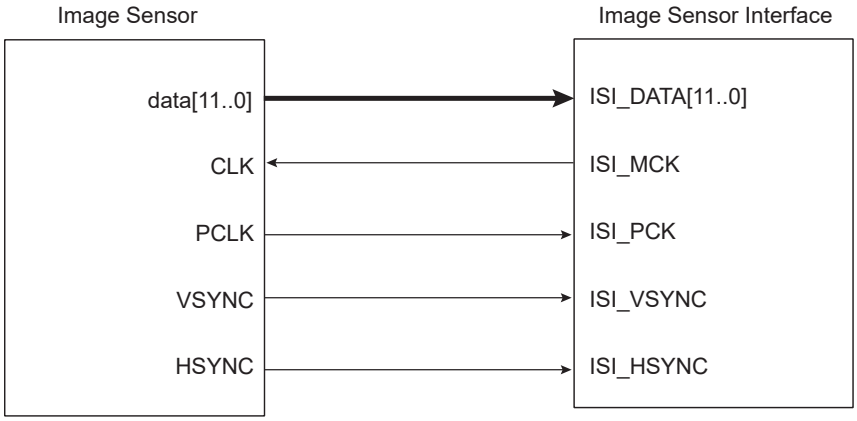
Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access								
Reset								

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

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – P PIO Pull-Up Status

Value	Description
0	Pullup resistor is enabled on the I/O line.
1	Pullup resistor is disabled on the I/O line.

Figure 37-1. ISI Connection Example



37.2 Embedded Characteristics

- ITU-R BT. 601/656 8-bit Mode External Interface Support
- Supports up to 12-bit Grayscale CMOS Sensors
- Support for ITU-R BT.656-4 SAV and EAV Synchronization
- Vertical and Horizontal Resolutions up to 2048 × 2048
- Preview Path up to 640 × 480 in RGB Mode
- Codec Path up to 2048 × 2048
- 16-byte FIFO on Codec Path
- 16-byte FIFO on Preview Path
- Support for Packed Data Formatting for YCbCr 4:2:2 Formats
- Preview Scaler to Generate Smaller Size image
- Programmable Frame Capture Rate
- VGA, QVGA, CIF, QCIF Formats Supported for LCD Preview
- Custom Formats with Horizontal and Vertical Preview Size as Multiples of 16 Also Supported for LCD Preview

The GMAC supports both hardware controlled pause of the transmitter, upon reception of a pause frame, and hardware generated pause frame transmission.

38.6.16.1 802.3 Pause Frame Reception

Bit 13 of the Network Configuration register is the pause enable control for reception. If this bit is set, transmission will pause if a non zero pause quantum frame is received.

If a valid pause frame is received then the Pause Time register is updated with the new frame's pause time, regardless of whether a previous pause frame is active or not. An interrupt (either bit 12 or bit 13 of the Interrupt Status register) is triggered when a pause frame is received, but only if the interrupt has been enabled (bit 12 and bit 13 of the Interrupt Mask register). Pause frames received with non zero quantum are indicated through the interrupt bit 12 of the Interrupt Status register. Pause frames received with zero quantum are indicated on bit 13 of the Interrupt Status register.

Once the Pause Time register is loaded and the frame currently being transmitted has been sent, no new frames are transmitted until the pause time reaches zero. The loading of a new pause time, and hence the pausing of transmission, only occurs when the GMAC is configured for full duplex operation. If the GMAC is configured for half duplex there will be no transmission pause, but the pause frame received interrupt will still be triggered. A valid pause frame is defined as having a destination address that matches either the address stored in Specific Address register 1 or if it matches the reserved address of 0x0180C2000001. It must also have the MAC control frame type ID of 0x8808 and have the pause opcode of 0x0001.

Pause frames that have frame check sequence (FCS) or other errors will be treated as invalid and will be discarded. 802.3 Pause frames that are received after Priority-based Flow Control (PFC) has been negotiated will also be discarded. Valid pause frames received will increment the pause frames received statistic register.

The pause time register decrements every 512 bit times once transmission has stopped. For test purposes, the retry test bit can be set (bit 12 in the Network Configuration register) which causes the Pause Time register to decrement every GTXCK cycle once transmission has stopped.

The interrupt (bit 13 in the Interrupt Status register) is asserted whenever the Pause Time register decrements to zero (assuming it has been enabled by bit 13 in the Interrupt Mask register). This interrupt is also set when a zero quantum pause frame is received.

38.6.16.2 802.3 Pause Frame Transmission

Automatic transmission of pause frames is supported through the transmit pause frame bits of the Network Control register. If either bit 11 or bit 12 of the Network Control register is written with logic 1, an 802.3 pause frame will be transmitted, providing full duplex is selected in the Network Configuration register and the transmit block is enabled in the Network Control register.

Pause frame transmission will happen immediately if transmit is inactive or if transmit is active between the current frame and the next frame due to be transmitted.

Transmitted pause frames comprise the following:

- A destination address of 01-80-C2-00-00-01
- A source address taken from Specific Address register 1
- A type ID of 88-08 (MAC control frame)
- A pause opcode of 00-01
- A pause quantum register
- Fill of 00 to take the frame to minimum frame length
- Valid FCS

39. USB High-Speed Interface (USBHS)

39.1 Description

The USB High-Speed Interface (USBHS) complies with the Universal Serial Bus (USB) 2.0 specification in all speeds.

Each pipe/endpoint can be configured in one of several USB transfer types. It can be associated with one, two or three banks of a DPRAM used to store the current data payload. If two or three banks are used, then one DPRAM bank is read or written by the CPU or the DMA, while the other is read or written by the USBHS core. This feature is mandatory for isochronous pipes/endpoints.

The following table describes the hardware configuration of the USB MCU device.

Table 39-1. Description of USB Pipes/Endpoints

Pipe/ Endpoint	Mnemonic	Max. Number Banks	DMA	High Band Width	Max. Pipe/ Endpoint Size	Type
0	PEP_0	1	N	N	64	Control
1	PEP_1	3	Y	Y	1024	Isochronous/Bulk/ Interrupt/Control
2	PEP_2	3	Y	Y	1024	Isochronous/Bulk/ Interrupt/Control
3	PEP_3	2	Y	Y	1024	Isochronous/Bulk/ Interrupt/Control
4	PEP_4	2	Y	Y	1024	Isochronous/Bulk/ Interrupt/Control
5	PEP_5	2	Y	Y	1024	Isochronous/Bulk/ Interrupt/Control
6	PEP_6	2	Y	Y	1024	Isochronous/Bulk/ Interrupt/Control
7	PEP_7	2	Y	Y	1024	Isochronous/Bulk/ Interrupt/Control
8	PEP_8	2	N	Y	1024	Isochronous/Bulk/ Interrupt/Control
9	PEP_9	2	N	Y	1024	Isochronous/Bulk/ Interrupt/Control

39.2 Embedded Characteristics

- Compatible with the USB 2.0 Specification
- Supports High-Speed (480 Mbps), Full-Speed (12 Mbps) and Low-Speed (1.5 Mbps) Communication

(PINGEN) bit and the bInterval Parameter for the Bulk-Out/Ping Transaction (BINTERVAL) field in USBHS_HSTPIPCFGx. See the Host Pipe x Configuration Register for additional information.

Figure 39-21. Example of an OUT Pipe with one Data Bank

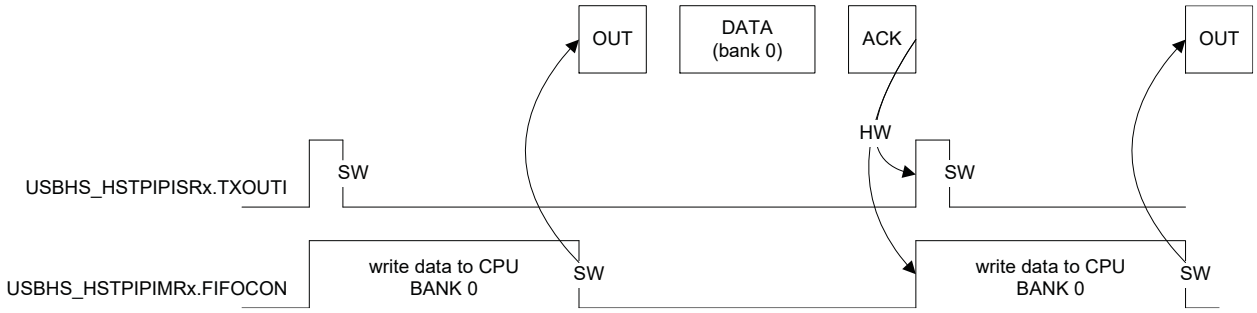


Figure 39-22. Example of an OUT Pipe with two Data Banks and no Bank Switching Delay

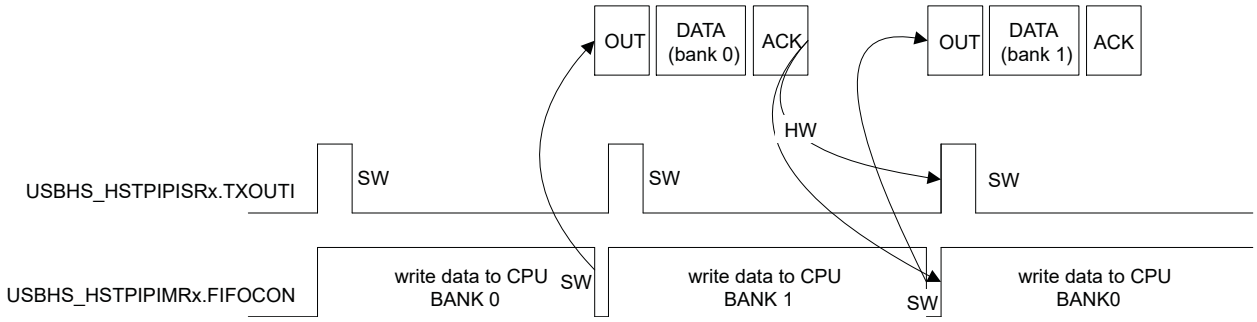
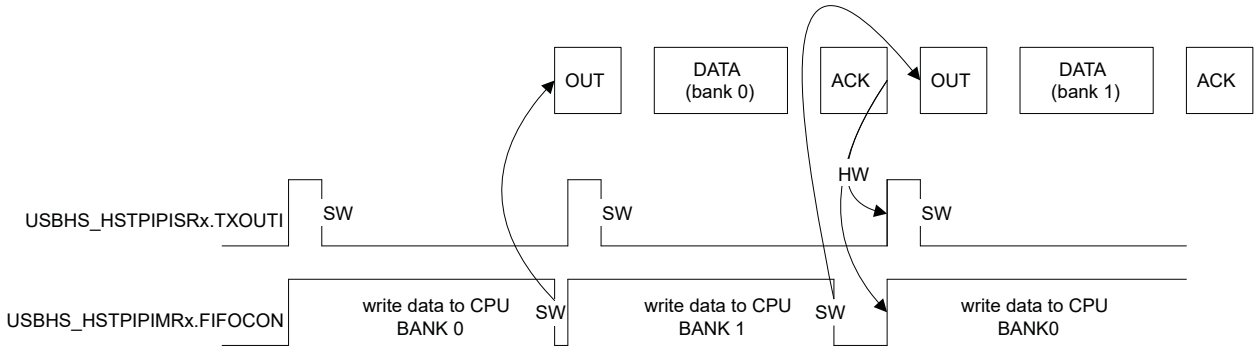


Figure 39-23. Example of an OUT Pipe with two Data Banks and a Bank Switching Delay



39.5.3.12 CRC Error

This error exists only for isochronous IN pipes. It sets the CRC Error Interrupt (USBHS_HSTPIPISRx.CRCERRI) bit, which triggers a PEP_x interrupt if then the CRC Error Interrupt Enable (USBHS_HSTPIPIMRx.CRCERRE) bit is one.

A CRC error can occur during IN stage if the USBHS detects a corrupted received packet. The IN packet is stored in the bank as if no CRC error had occurred (USBHS_HSTPIPISRx.RXINI is set).

39.5.3.13 Interrupts

See the structure of the USB host interrupt system on Figure 39-3.

There are two kinds of host interrupts: processing, i.e., their generation is part of the normal processing, and exception, i.e., errors (not related to CPU exceptions).

Global Interrupts

The processing host global interrupts are:

Bits 21:19 – TRTYP[2:0] Transfer Type

Value	Name	Description
0	SINGLE	MMC/SD Card Single Block
1	MULTIPLE	MMC/SD Card Multiple Block
2	STREAM	MMC Stream
4	BYTE	SDIO Byte
5	BLOCK	SDIO Block

Bit 18 – TRDIR Transfer Direction
0 (WRITE): Write.

1 (READ): Read.

Bits 17:16 – TRCMD[1:0] Transfer Command

Value	Name	Description
0	NO_DATA	No data transfer
1	START_DATA	Start data transfer
2	STOP_DATA	Stop data transfer
3	Reserved	Reserved

Bit 12 – MAXLAT Max Latency for Command to Response
0 (5): 5-cycle max latency.

1 (64): 64-cycle max latency.

Bit 11 – OPDCMD Open Drain Command
0 (PUSHPULL): Push pull command.

1 (OPENDRAIN): Open drain command.

Bits 10:8 – SPCMD[2:0] Special Command

Value	Name	Description
0	STD	Not a special CMD.
1	INIT	Initialization CMD: 74 clock cycles for initialization sequence.
2	SYNC	Synchronized CMD: Wait for the end of the current data block transfer before sending the pending command.
3	CE_ATA	CE-ATA Completion Signal disable Command. The host cancels the ability for the device to return a command completion signal on the command line.
4	IT_CMD	Interrupt command: Corresponds to the Interrupt Mode (CMD40).
5	IT_RESP	Interrupt response: Corresponds to the Interrupt Mode (CMD40).

Value	Name	Description
6	65536	CSTOCYC x 65536
7	1048576	CSTOCYC x 1048576

Bits 3:0 – CSTOCYC[3:0] Completion Signal Timeout Cycle Number
This field determines the maximum number of Master Clock cycles that the HSMCI waits between two data block transfers. Its value is calculated by (CSTOCYC x Multiplier).

Bit 5 – OVRE Overrun Error Interrupt Disable

Bit 1 – TXRDY TXRDY Interrupt Disable

Bit 0 – RXRDY RXRDY Interrupt Disable

46.7.14 USART Interrupt Mask Register (SPI_MODE)

Name: US_IMR (SPI_MODE)
Offset: 0x0010
Reset: 0x0
Property: Read-only

This configuration is relevant only if USART_MODE = 0xE or 0xF in the USART Mode Register.

The following configuration values are valid for all listed bit names of this register:

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset					0			

Bit	15	14	13	12	11	10	9	8
Access								
Reset						0	0	

Bit	7	6	5	4	3	2	1	0
Access								
Reset			0				0	0

Bit 19 – NSSE NSS Line (Driving CTS Pin) Rising or Falling Edge Event Interrupt Mask

Bit 10 – UNRE SPI Underrun Error Interrupt Mask

Bit 9 – TXEMPTY TXEMPTY Interrupt Mask

Bit 5 – OVRE Overrun Error Interrupt Mask

Bit 1 – TXRDY TXRDY Interrupt Mask

Bit 0 – RXRDY RXRDY Interrupt Mask

Value	Description
0	No LIN checksum error has been detected since the last RSTSTA.
1	A LIN checksum error has been detected since the last RSTSTA.

Bit 27 – LINIPE LIN Identifier Parity Error (cleared by writing a one to bit US_CR.RSTSTA)

Value	Description
0	No LIN identifier parity error has been detected since the last RSTSTA.
1	A LIN identifier parity error has been detected since the last RSTSTA.

Bit 26 – LINISFE LIN Inconsistent Synch Field Error (cleared by writing a one to bit US_CR.RSTSTA)

Value	Description
0	No LIN inconsistent synch field error has been detected since the last RSTSTA
1	The USART is configured as a slave node and a LIN Inconsistent synch field error has been detected since the last RSTSTA.

Bit 25 – LINBE LIN Bit Error (cleared by writing a one to bit US_CR.RSTSTA)

Value	Description
0	No bit error has been detected since the last RSTSTA.
1	A bit error has been detected since the last RSTSTA.

Bit 23 – LINBLS LIN Bus Line Status

Value	Description
0	LIN bus line is set to 0.
1	LIN bus line is set to 1.

Bit 15 – LINTC LIN Transfer Completed (cleared by writing a one to bit US_CR.RSTSTA)

Value	Description
0	The USART is idle or a LIN transfer is ongoing.
1	A LIN transfer has been completed since the last RSTSTA.

Bit 14 – LINID LIN Identifier Sent or LIN Identifier Received (cleared by writing a one to bit US_CR.RSTSTA)
If USART operates in LIN Master mode (USART_MODE = 0xA):

If USART operates in LIN Slave mode (USART_MODE = 0xB):

Value	Description
0	No LIN identifier has been sent since the last RSTSTA.
1	At least one LIN identifier has been sent since the last RSTSTA.
0	No LIN identifier has been received since the last RSTSTA.
1	At least one LIN identifier has been received since the last RSTSTA

Bit 13 – LINBK LIN Break Sent or LIN Break Received (cleared by writing a one to bit US_CR.RSTSTA)
Applicable if USART operates in LIN master mode (USART_MODE = 0xA):

If USART operates in LIN Slave mode (USART_MODE = 0xB):

- MLB_MADR.WNR = 1
- MLB_MADR.TB = 1
- MLB_MADR.ADDR[13:0] = 14-bit Target Address

The MIF block sets MLB_MCTL.XCMP = 1 to inform the HC when the write is complete.

Direct DBR Reads

For a direct read of the DBR, the HC initiates a read cycle by writing the address and control information to MLB_MADR as follows:

- MLB_MADR.WNR = 0
- MLB_MADR.TB = 1
- MLB_MADR.ADDR[13:0] = 14-bit target address

The MIF block sets MLB_MCTL.XCMP = 1 to inform the HC when the read is complete. The HC can then read the 8-bit data entry from the MLB_MDAT0 register at bits[7:0].

48.6.3.5 Interrupt Interface Block

The Interrupt Interface (INTIF) block performs a low-priority polling algorithm of each of the HBI channel descriptors.

The INTIF alerts the HBI block when specific changes to HBI Channel Descriptors occur.

- For asynchronous and control read/write channels:
  - a packet is available to read in the channel buffer, or
  - sufficient empty space is available in the channel buffer to accept a requested packet write.
- For isochronous read/write channels:
  - the number of valid bytes in the channel buffer exceeds the block size, or
  - the number of empty bytes in the channel buffer exceeds the block size.

48.6.3.6 AHB Block

The AHB block manages data exchange between local channel data buffers within the MLB and the system memory buffer.

To support system memory buffering, a ping-pong memory structure is implemented on a per-channel basis using 128-bit descriptors for AHB Descriptor Table (ADT) entries.

Note: The 64 ADT entries are directly mapped to the 64 HBI physical channels.

Each logical channel is assigned a separate 128-bit descriptor, defining the data buffers in the system memory used by the DMA interface for that channel. The descriptors are stored at fixed addresses in the external CTR.

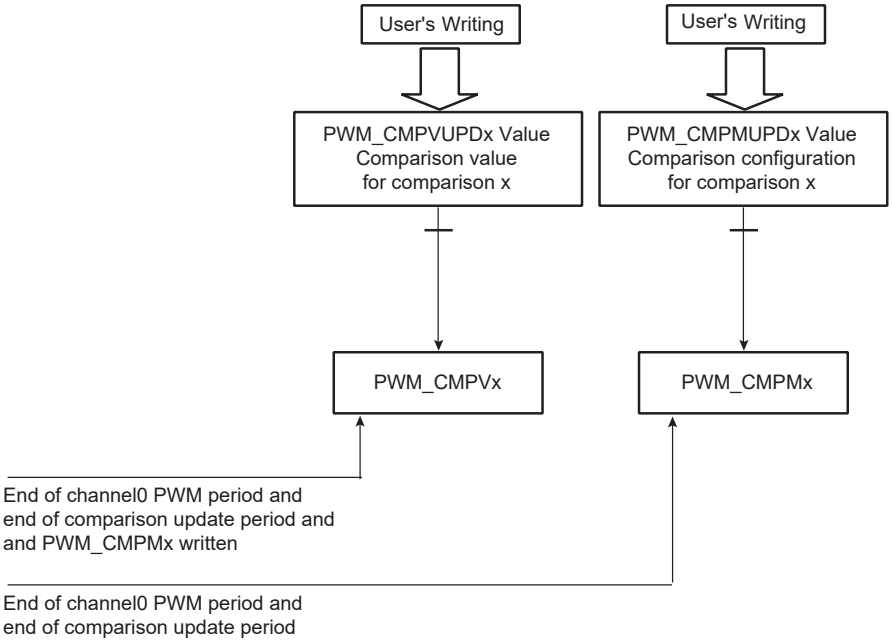
AHB Descriptor Table

The following table provides an overview of field definitions for ADT entries.

Table 48-20. ADT Field Definitions

Field	No. of Bits	Description	Accessibility
CE	1	Channel enable: 0 = Disabled	r,w,u (1)

Figure 51-36. Synchronized Update of Comparison Values and Configurations



51.6.6.6 Interrupt Sources

Depending on the interrupt mask in PWM_IMR1 and PWM_IMR2, an interrupt can be generated at the end of the corresponding channel period (CHIDx in the PWM Interrupt Status Register 1 (PWM_ISR1)), after a fault event (FCHIDx in PWM_ISR1), after a comparison match (CMPMx in PWM_ISR2), after a comparison update (CMPUx in PWM_ISR2) or according to the Transfer mode of the synchronous channels (WRDY and UNRE in PWM_ISR2).

If the interrupt is generated by the flags CHIDx or FCHIDx, the interrupt remains active until a read operation in PWM_ISR1 occurs.

If the interrupt is generated by the flags WRDY or UNRE or CMPMx or CMPUx, the interrupt remains active until a read operation in PWM_ISR2 occurs.

A channel interrupt is enabled by setting the corresponding bit in PWM_IER1 and PWM_IER2. A channel interrupt is disabled by setting the corresponding bit in PWM_IDR1 and PWM_IDR2.

51.6.7 Register Write Protection

To prevent any single software error that may corrupt PWM behavior, the registers listed below can be write-protected by writing the field WPCMD in the PWM Write Protection Control Register (PWM_WPCR). They are divided into six groups:

- Register group 0:
  - PWM Clock Register
- Register group 1:
  - PWM Disable Register
  - PWM Interrupt Enable Register 1
  - PWM Interrupt Disable Register 1
  - PWM Interrupt Enable Register 2
  - PWM Interrupt Disable Register 2
- Register group 2:

52.7.23 AFEC Analog Control Register

Name: AFEC_ACR
Offset: 0x94
Reset: 0x00000100
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

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

Bit	7	6	5	4	3	2	1	0
					PGA1EN	PGA0EN		
Access					R/W	R/W		
Reset					0	0		

Bits 9:8 – IBCTL[1:0] AFE Bias Current Control
Adapts performance versus power consumption. (Refer to AFE Characteristics in section “Electrical Characteristics”.)

Bit 3 – PGA1EN PGA1 Enable

Value	Description
0	Programmable Gain Amplifier is disabled.
1	Programmable Gain Amplifier is enabled.

Bit 2 – PGA0EN PGA0 Enable

Value	Description
0	Programmable Gain Amplifier is disabled.
1	Programmable Gain Amplifier is enabled.