Welcome to E-XFL.COM

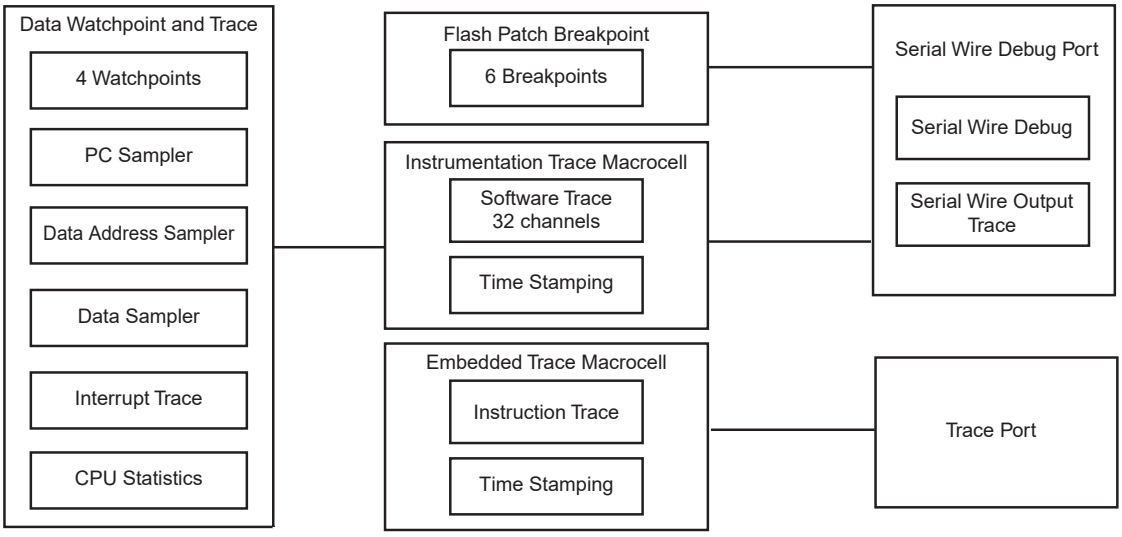
What is "Embedded - Microcontrollers"?

"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, Ethernet, I²C, IrDA, LINbus, SPI, SSC, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	44
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 5x12b; D/A 1x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsame70j20a-an

Figure 16-4. Debug Architecture



16.7.3 Serial Wire Debug Port (SW-DP) Pins

The SW-DP pins SWCLK and SWDIO are commonly provided on a standard 20-pin JTAG connector defined by ARM. For more details on voltage reference and reset state, refer to the "Signal Description" chapter.

At startup, SW-DP pins are configured in SW-DP mode to allow connection with debugging probe.

SW-DP pins can be used as standard I/Os to provide users more general input/output pins when the debug port is not needed in the end application. Mode selection between SW-DP mode (System I/O mode) and general I/O mode is performed through the AHB Matrix Chip Configuration registers (CCFG_SYSIO). Configuration of the pad for pullup, triggers, debouncing and glitch filters is possible regardless of the mode.

The JTAGSEL pin is used to select the JTAG boundary scan when asserted at a high level. It integrates a permanent pulldown resistor of about 15 kΩ to GND, so that it can be left unconnected for normal operations.

The JTAG debug ports TDI, TDO, TMS and TCK are inactive. They are provided for Boundary Scan Manufacturing Test purposes only. By default the SW-DP is active; TDO/TRACESWO can be used for trace.

Table 16-2. SW-DP Pin List

Pin Name	JTAG Boundary Scan	Serial Wire Debug Port
TMS/SWDIO	TMS	SWDIO
TCK/SWCLK	TCK	SWCLK
TDI	TDI	–
TDO/TRACESWO	TDO	TRACESWO (optional: trace)

SW-DP is selected when JTAGSEL is low. It is not possible to switch directly between SW-DP and JTAG boundary scan operations. A chip reset must be performed after JTAGSEL is changed.

20. USB Transmitter Macrocell Interface (UTMI)

20.1 Description

The USB Transmitter Macrocell Interface (UTMI) registers manage specific aspects of the integrated USB transmitter macrocell functionality not controlled in USB sections.

20.2 Embedded Characteristics

- 32-bit UTMI Registers Control Product-specific Behavior

21.3.2 Chip ID Extension Register

Name: CHIPID_EXID
Offset: 0x4
Reset: -
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	EXID[31:24]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	23	22	21	20	19	18	17	16
	EXID[23:16]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	15	14	13	12	11	10	9	8
	EXID[15:8]							
Access	R	R	R	R	R	R	R	R
Reset								

Bit	7	6	5	4	3	2	1	0
	EXID[7:0]							
Access	R	R	R	R	R	R	R	R
Reset								

Bits 31:0 – EXID[31:0] Chip ID Extension
This field is cleared if CHIPID_CIDR.EXT = 0.

Value	Name	Description
0xX	Reserved	Reserved

Figure 22-1. Flash Memory Areas

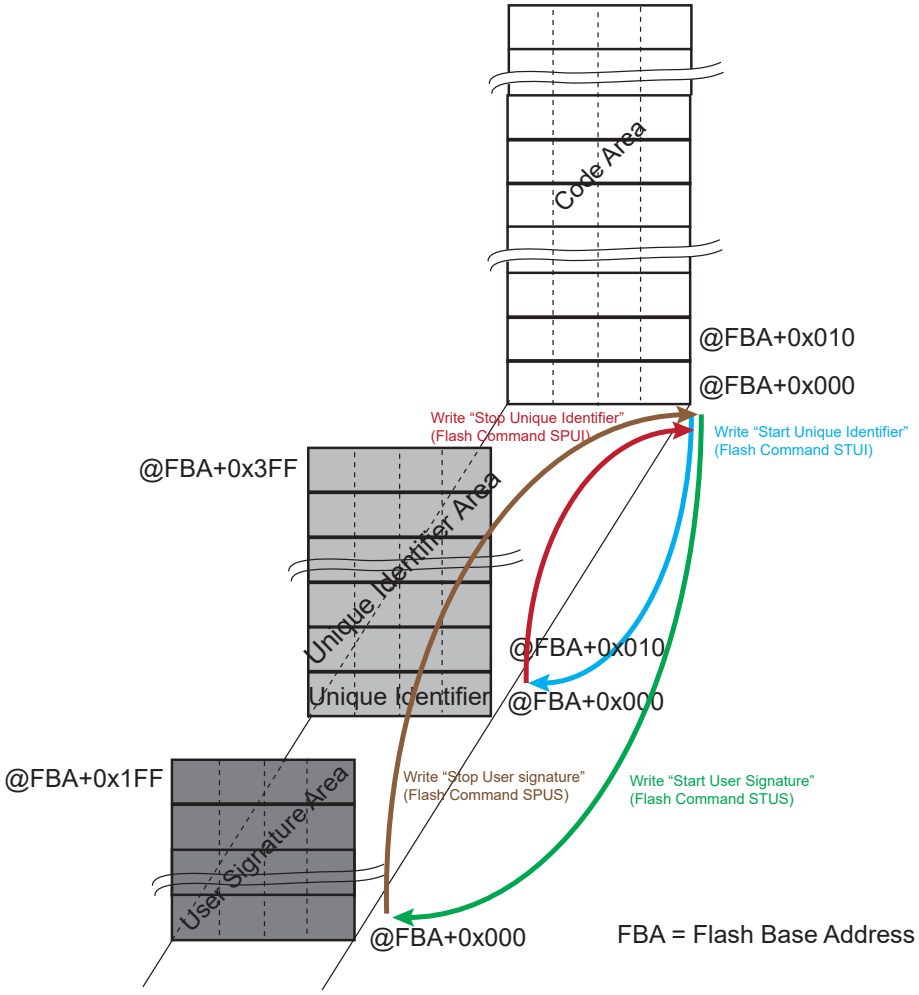
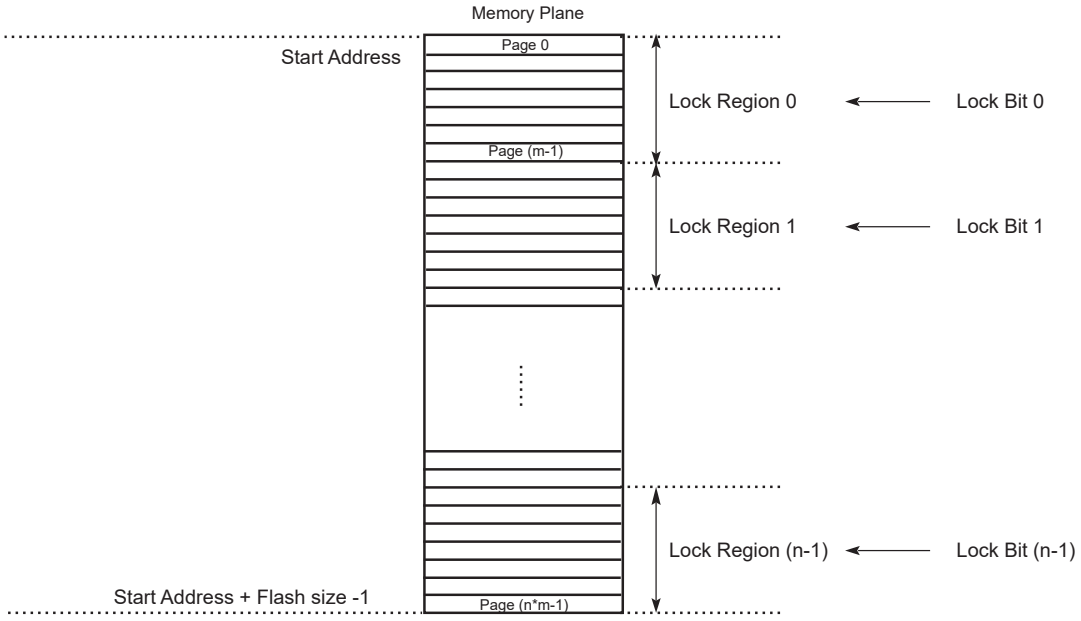


Figure 22-2. Organization of Embedded Flash for Code



3. Write the XDMAC_CSAx register for channel x.
4. Write the XDMAC_CDAx register for channel x.
5. Program XDMAC_CUBCx.UBLEN with the number of data.
6. Program XDMAC_CCx register (see "Single Block Transfer With Single Microblock").
7. Program XDMAC_CBCx.BLEN with the number of microblocks of data.
8. Clear the following registers:
   – XDMAC_CNDCx
   – XDMAC_CDS_MSPx
   – XDMAC_CSUSx XDMAC_CDUSx
   This indicates that the linked list is disabled and striding is disabled.
9. Enable the Block interrupt by writing a '1' to XDMAC_CIEx.BIE, enable the Channel x Interrupt Enable bit by writing a '1' to XDMAC_GIEx.IEx.
10. Enable channel x by writing a '1' to the XDMAC_GE.ENx. XDMAC_GS.STx is set by hardware.
11. Once completed, the DMA channel sets XDMAC_CISx.BIS (End of Block Interrupt Status bit) and generates an interrupt. XDMAC_GS.STx is cleared by hardware. The software can either wait for an interrupt or poll the channel status bit.

36.5.4.3 Master Transfer

1. Read the XDMAC_GS register to choose a free channel.
2. Clear the pending Interrupt Status bit by reading the chosen XDMAC_CISx register.
3. Build a linked list of transfer descriptors in memory. The descriptor view is programmable on a per descriptor basis. The linked list items structure must be word aligned. MBR_UBC.NDE must be configured to 0 in the last descriptor to terminate the list.
4. Configure field NDA in the XDMAC Channel x Next Descriptor Address Register (XDMAC_CNDAx) with the first descriptor address and bit XDMAC_CNDAx.NDAIF with the master interface identifier.
5. Configure the XDMAC_CNDCx register:
   5.1. Set XDMAC_CNDCx.NDE to enable the descriptor fetch.
   5.2. Set XDMAC_CNDCx.NDSUP to update the source address at the descriptor fetch time, otherwise clear this bit.
   5.3. Set XDMAC_CNDCx.NDDUP to update the destination address at the descriptor fetch time, otherwise clear this bit.
   5.4. Configure XDMAC_CNDCx.NDVIEW to define the length of the first descriptor.
6. Enable the End of Linked List interrupt by writing a '1' to XDMAC_CIEx.LIE.
7. Enable channel x by writing a '1' to XDMAC_GE.ENx. XDMAC_GS.STx is set by hardware.
8. Once completed, the DMA channel sets XDMAC_CISx.BIS (End of Block Interrupt Status bit) and generates an interrupt. XDMAC_GS.STx is cleared by hardware. The software can either wait for an interrupt or poll the channel status bit.

36.5.4.4 Disabling A Channel Before Transfer Completion

Under normal operation, the software enables a channel by writing a '1' to XDMAC_GE.ENx, then the hardware disables a channel on transfer completion by clearing bit XDMAC_GS.STx. To disable a channel, write a '1' to bit XDMAC_GD.DIx and poll the XDMAC_GS register.

38.8.32 GMAC 1588 Timer Second Comparison Low Register

Name: GMAC_SCL
Offset: 0x0E0
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	SEC[31:24]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	SEC[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bit	7	6	5	4	3	2	1	0
	SEC[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – SEC[31:0] 1588 Timer Second Comparison Value
Value is compared to seconds value bits [31:0] of the TSU timer count value.

38.8.75 GMAC Length Field Frame Errors Register

Name: GMAC_LFFE
Offset: 0x194
Reset: 0x00000000
Property: -

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
							LFER[9:8]	
Access							R	R
Reset							0	0

Bit	7	6	5	4	3	2	1	0
	LFER[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 9:0 – LFER[9:0] Length Field Frame Errors
This bit field counts the number of frames received that have a measured length shorter than that extracted from the length field (Bytes 13 and 14). This condition is only counted if the value of the length field is less than 0x0600, the frame is not of excessive length and checking is enabled by writing a '1' to the Length Field Error Frame Discard bit in the Network Configuration Register (GMAC_NCFGR.LFERD).

Offset	Name	Bit Pos.								
		31:24								
0x0564	USBHS_HSTPIPICR1 (INTPIPES)	7:0	SHORTPACKETIC	RXSTALLDIC	OVERFIC	NAKEDIC		UNDERFIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x0564	USBHS_HSTPIPICR1 (ISOPIPES)	7:0	SHORTPACKETIC	CRCERRIC	OVERFIC	NAKEDIC		UNDERFIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x0568	USBHS_HSTPIPICR2	7:0	SHORTPACKETIC	RXSTALLDIC	OVERFIC	NAKEDIC		TXSTPIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x0568	USBHS_HSTPIPICR2 (INTPIPES)	7:0	SHORTPACKETIC	RXSTALLDIC	OVERFIC	NAKEDIC		UNDERFIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x0568	USBHS_HSTPIPICR2 (ISOPIPES)	7:0	SHORTPACKETIC	CRCERRIC	OVERFIC	NAKEDIC		UNDERFIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x056C	USBHS_HSTPIPICR3	7:0	SHORTPACKETIC	RXSTALLDIC	OVERFIC	NAKEDIC		TXSTPIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x056C	USBHS_HSTPIPICR3 (INTPIPES)	7:0	SHORTPACKETIC	RXSTALLDIC	OVERFIC	NAKEDIC		UNDERFIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x056C	USBHS_HSTPIPICR3 (ISOPIPES)	7:0	SHORTPACKETIC	CRCERRIC	OVERFIC	NAKEDIC		UNDERFIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								
0x0570	USBHS_HSTPIPICR4	7:0	SHORTPACKETIC	RXSTALLDIC	OVERFIC	NAKEDIC		TXSTPIC	TXOUTIC	RXINIC
		15:8								
		23:16								
		31:24								

Bit 1 – RXOUTIS Received OUT Data Interrupt Set

Bit 0 – TXINIS Transmitted IN Data Interrupt Set

For all the read operations, the NOTBUSY flag is cleared at the end of the host command.
For the Infinite Read Multiple Blocks, the NOTBUSY flag is set at the end of the STOP_TRANSMISSION host command (CMD12).

For the Single Block Reads, the NOTBUSY flag is set at the end of the data read block.

For the Multiple Block Reads with predefined block count, the NOTBUSY flag is set at the end of the last received data block.

The NOTBUSY flag allows to deal with these different states.

Value	Description
0	The HSMCI is not ready for new data transfer. Cleared at the end of the card response.
1	The HSMCI is ready for new data transfer. Set when the busy state on the data line has ended. This corresponds to a free internal data receive buffer of the card.

Bit 4 – DTIP Data Transfer in Progress (cleared at the end of CRC16 calculation)

Value	Description
0	No data transfer in progress.
1	The current data transfer is still in progress, including CRC16 calculation.

Bit 3 – BLKE Data Block Ended (cleared on read)
This flag must be used only for Write Operations.

Refer to the MMC or SD Specification for more details concerning the CRC Status.

Value	Description
0	A data block transfer is not yet finished.
1	A data block transfer has ended, including the CRC16 Status transmission. The flag is set for each transmitted CRC Status.

Bit 2 – TXRDY Transmit Ready (cleared by writing in HSMCI_TDR)

Value	Description
0	The last data written in HSMCI_TDR has not yet been transferred in the Shift Register.
1	The last data written in HSMCI_TDR has been transferred in the Shift Register.

Bit 1 – RXRDY Receiver Ready (cleared by reading HSMCI_RDR)

Value	Description
0	Data has not yet been received since the last read of HSMCI_RDR.
1	Data has been received since the last read of HSMCI_RDR.

Bit 0 – CMDRDY Command Ready (cleared by writing in HSMCI_CMDR)

Value	Description
0	A command is in progress.
1	The last command has been sent.

last character transfer. Then, another DMA transfer can be started if SPI_CR.SPIEN has previously been written.

41.7.3.6 SPI Direct Access Memory Controller (DMAC)

In both Fixed and Variable modes, the Direct Memory Access Controller (DMAC) can be used to reduce processor overhead.

The fixed peripheral selection allows buffer transfers with a single peripheral. Using the DMAC is an optimal means, as the size of the data transfer between the memory and the SPI is either 8 bits or 16 bits. However, if the peripheral selection is modified, SPI_MR must be reprogrammed.

The variable peripheral selection allows buffer transfers with multiple peripherals without reprogramming SPI_MR. Data written in SPI_TDR is 32 bits wide and defines the real data to be transmitted and the destination peripheral. Using the DMAC in this mode requires 32-bit wide buffers, with the data in the LSBs and the PCS and LASTXFER fields in the MSBs. However, the SPI still controls the number of bits (8 to 16) to be transferred through MISO and MOSI lines with the chip select configuration registers. This is not the optimal means in terms of memory size for the buffers, but it provides a very effective means to exchange data with several peripherals without any intervention of the processor.

41.7.3.7 Peripheral Chip Select Decoding

The user can program the SPI to operate with up to 15 slave peripherals by decoding the four chip select lines, NPCS0 to NPCS3 with an external decoder/demultiplexer (see figure below). This can be enabled by setting SPI_MR.PCSDEC.

When operating without decoding, the SPI makes sure that in any case only one chip select line is activated, i.e., one NPCS line driven low at a time. If two bits are defined low in a PCS field, only the lowest numbered chip select is driven low.

When operating with decoding, the SPI directly outputs the value defined by the PCS field on the NPCS lines of either SPI_MR or SPI_TDR (depending on PS).

As the SPI sets a default value of 0xF on the chip select lines (i.e., all chip select lines at 1) when not processing any transfer, only 15 peripherals can be decoded.

The SPI has four chip select registers (SPI_CSR0...SPI_CSR3). As a result, when external decoding is activated, each NPCS chip select defines the characteristics of up to four peripherals. As an example, SPI_CRS0 defines the characteristics of the externally decoded peripherals 0 to 3, corresponding to the PCS values 0x0 to 0x3. Consequently, the user has to make sure to connect compatible peripherals on the decoded chip select lines 0 to 3, 4 to 7, 8 to 11 and 12 to 14. The following figure shows this type of implementation.

If SPI_CSRx.CSAAT bit is used, with or without the DMAC, the Mode Fault detection for NPCS0 line must be disabled. This is not needed for all other chip select lines since Mode Fault detection is only on NPCS0.

Bits 15:8 – SCBR[7:0] Serial Clock Bit Rate
In Master mode, the SPI Interface uses a modulus counter to derive the SPCK bit rate from the peripheral clock. The bit rate is selected by writing a value from1 to 255 in the SCBR field. The following equation determines the SPCK bit rate:

SCBR = $f_{peripheral\ clock}$ / SPCK Bit Rate

Programming the SCBR field to 0 is forbidden. Triggering a transfer while SCBR is at 0 can lead to unpredictable results.

If BRSRCCLK = 1 in SPI_MR, SCBR must be programmed with a value greater than 1.

At reset, SCBR is 0 and the user has to program it at a valid value before performing the first transfer.

Note: If one of the SCBR fields in SPI_CSRx is set to 1, the other SCBR fields in SPI_CSRx must be set to 1 as well, if they are used to process transfers. If they are not used to transfer data, they can be set at any value.

Bits 7:4 – BITS[3:0] Bits Per Transfer
(See Note under the register table in SPI Chip Select Register.)

The BITS field determines the number of data bits transferred. Reserved values should not be used.

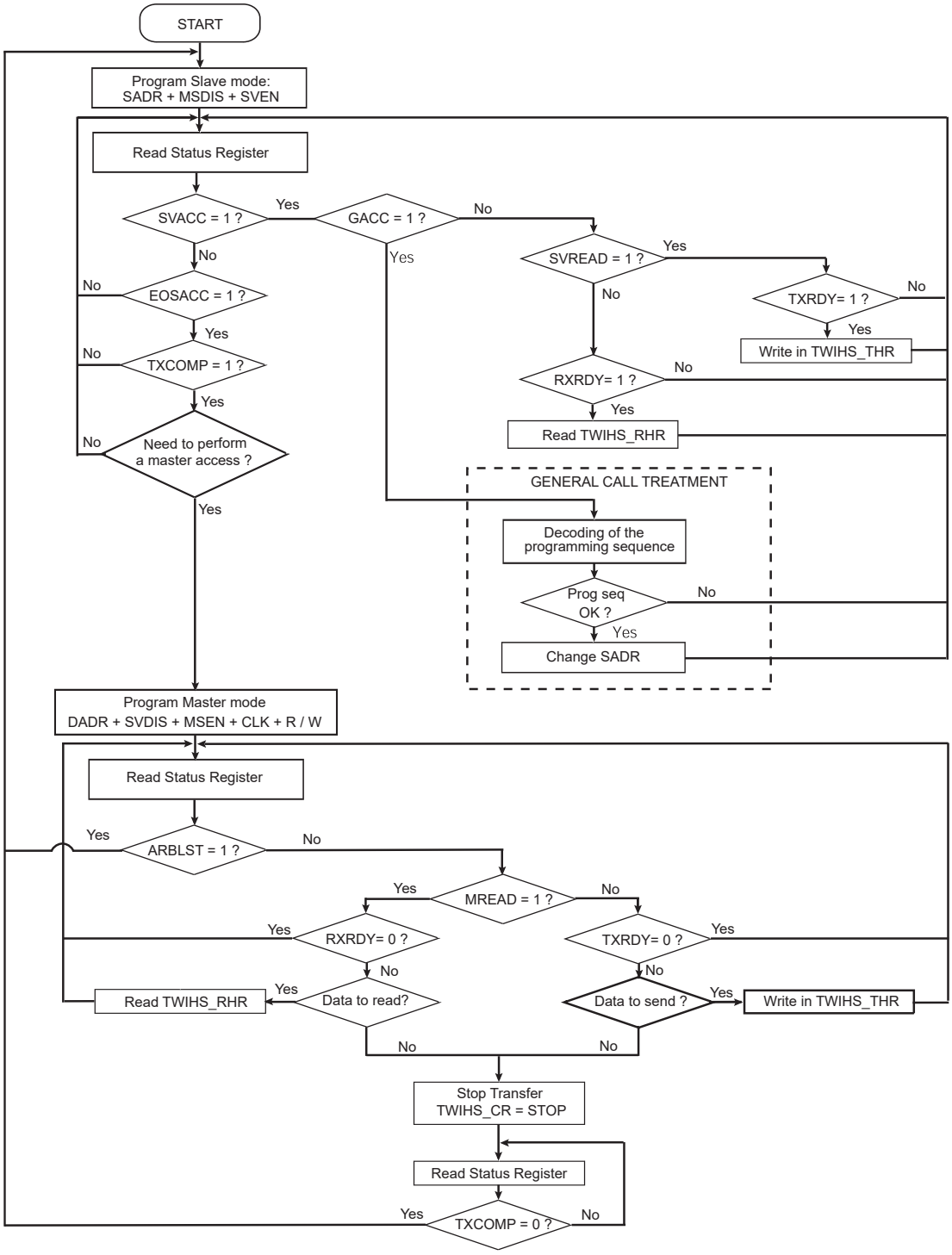
Value	Name	Description
0	8_BIT	8 bits for transfer
1	9_BIT	9 bits for transfer
2	10_BIT	10 bits for transfer
3	11_BIT	11 bits for transfer
4	12_BIT	12 bits for transfer
5	13_BIT	13 bits for transfer
6	14_BIT	14 bits for transfer
7	15_BIT	15 bits for transfer
8	16_BIT	16 bits for transfer
9	–	Reserved
10	–	Reserved
11	–	Reserved
12	–	Reserved
13	–	Reserved
14	–	Reserved
15	–	Reserved

Bit 3 – CSAAT Chip Select Active After Transfer

Value	Description
0	The Peripheral Chip Select Line rises as soon as the last transfer is achieved.
1	The Peripheral Chip Select Line does not rise after the last transfer is achieved. It remains active until a new transfer is requested on a different chip select.

Bit 2 – CSNAAT Chip Select Not Active After Transfer (ignored if CSAAT = 1)

Figure 43-30. Multimaster Flowchart



43.6.5 Slave Mode

43.6.5.1 Definition

Slave mode is defined as a mode where the device receives the clock and the address from another device called the master.

Bit 9 – RXDMA Single or Multiple DMA Controller Channels for Receiver

Value	Description
0	The receiver uses only one DMA Controller channel for all audio channels.
1	The receiver uses one DMA Controller channel per audio channel.

Bit 8 – RXMONO Receive Mono

Value	Description
0	Stereo
1	Mono, with left audio samples duplicated to right audio channel by the I2SC.

Bits 7:6 – FORMAT[1:0] Data Format

Value	Name	Description
0	I2S	I^2S format, stereo with I2SC_WS low for left channel, and MSB of sample starting one I2SC_CK period after I2SC_WS edge
1	LJ	Left-justified format, stereo with I2SC_WS high for left channel, and MSB of sample starting on I2SC_WS edge
2	–	Reserved
3	–	Reserved

Bits 4:2 – DATALENGTH[2:0] Data Word Length

Value	Name	Description
0	32_BITS	Data length is set to 32 bits.
1	24_BITS	Data length is set to 24 bits.
2	20_BITS	Data length is set to 20 bits.
3	18_BITS	Data length is set to 18 bits.
4	16_BITS	Data length is set to 16 bits.
5	16_BITS_COMPACT	Data length is set to 16-bit compact stereo. Left sample in bits 15:0 and right sample in bits 31:16 of same word.
6	8_BITS	Data length is set to 8 bits.
7	8_BITS_COMPACT	Data length is set to 8-bit compact stereo. Left sample in bits 7:0 and right sample in bits 15:8 of the same word.

Bit 0 – MODE Inter-IC Sound Controller Mode

Value	Name	Description
0	SLAVE	I2SC_CK and I2SC_WS pin inputs used as bit clock and word select/frame synchronization.
1	MASTER	Bit clock and word select/frame synchronization generated by I2SC from MCK and output to I2SC_CK and I2SC_WS pins. Peripheral clock or GCLK is output as master clock on I2SC_MCK if I2SC_MR.IMCKMODE is set.

47.6.8 UART Transmit Holding Register

Name: UART_THR
Offset: 0x1C
Reset: –
Property: Write-only

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
Access								
Reset								

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

Bits 7:0 – TXCHR[7:0] Character to be Transmitted
Next character to be transmitted after the current character if TXRDY is not set.

49.5.7.3 Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS and TXBC.NDTB. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC.

Table 49-8. Tx Buffer Element

	31			24	23				16	15	8	7	0
T0	ESI	XTD	RTR	ID[28:0]									
T1	MM[7:0]				EFC	reserved	FDF	BRS	DLC[3:0]	reserved			
T2	DB3[7:0]				DB2[7:0]					DB1[7:0]		DB0[7:0]	
T3	DB7[7:0]				DB6[7:0]					DB5[7:0]		DB4[7:0]	
...	...				...					...		...	
Tn	DBm[7:0]				DBm-1[7:0]					DBm-2[7:0]		DBm-3[7:0]	

• T0 Bit 30 ESI: Error State Indicator

T0 Bit 31 ESI: Error State Indicator

0: ESI bit in CAN FD format depends only on error passive flag

1: ESI bit in CAN FD format transmitted recessive

Note: The ESI bit of the transmit buffer is or’ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive. This feature can be used in gateway applications when a message from an error passive node is routed to another CAN network.

• T0 Bit 30 XTD: Extended Identifier

0: 11-bit standard identifier.

1: 29-bit extended identifier.

• T0 Bit 29 RTR: Remote Transmission Request

0: Transmit data frame.

1: Transmit remote frame.

Note: When RTR = 1, the MCAN transmits a remote frame according to ISO11898-1, even if MCAN_CCCR.FDOE enables the transmission in CAN FD format.

• T0 Bits 28:0 ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].

• T1 Bits 31:24 MM[7:0]: Message Marker

Written by processor during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.

Register	UPDM = 0	UPDM = 1	UPDM = 2
	Update is triggered at the next PWM period as soon as the bit UPDULOCK is set to '1'	Update is triggered at the next PWM period as soon as the update period counter has reached the value UPR.	
Update Period Value (PWM_SCUPUPD)	Not applicable	Write by the processor	
	Not applicable	Update is triggered at the next PWM period as soon as the update period counter has reached the value UPR.	

51.6.2.9.1 Method 1: Manual write of duty-cycle values and manual trigger of the update

In this mode, the update of the period value, the duty-cycle values and the dead-time values must be done by writing in their respective update registers with the processor (respectively PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx).

To trigger the update, the user must use the bit UPDULOCK in the PWM_SCUC register which allows to update synchronously (at the same PWM period) the synchronous channels:

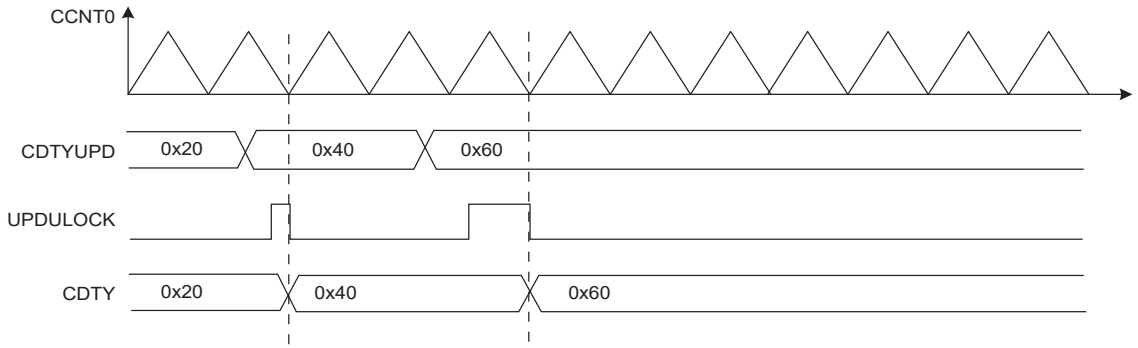
- If the bit UPDULOCK is set to '1', the update is done at the next PWM period of the synchronous channels.
- If the UPDULOCK bit is not set to '1', the update is locked and cannot be performed.

After writing the UPDULOCK bit to '1', it is held at this value until the update occurs, then it is read 0.

Sequence for Method 1:

1. Select the manual write of duty-cycle values and the manual update by setting the UPDM field to '0' in the PWM_SCM register.
2. Define the synchronous channels by the SYNCx bits in the PWM_SCM register.
3. Enable the synchronous channels by writing CHID0 in the PWM_ENA register.
4. If an update of the period value and/or the duty-cycle values and/or the dead-time values is required, write registers that need to be updated (PWM_CPRDUPDx, PWM_CDTYUPDx and PWM_DTUPDx).
5. Set UPDULOCK to '1' in PWM_SCUC.
6. The update of the registers will occur at the beginning of the next PWM period. When the UPDULOCK bit is reset, go to Step 4. for new values.

Figure 51-19. Method 1 (UPDM = 0)



51.7.49 PWM External Trigger Register

Name: PWM_ETRGx
Offset: 0x042C + x*0x20 [x=0..1]
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	RFEN	TRGSRC	TRGFILT	TRGEDGE			TRGMODE[1:0]	
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

Bit	23	22	21	20	19	18	17	16
	MAXCNT[23:16]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

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

Bit	7	6	5	4	3	2	1	0
	MAXCNT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 31 – RFEN Recoverable Fault Enable

Value	Description
0	The TRGINx signal does not generate a recoverable fault.
1	The TRGINx signal generate a recoverable fault in place of the fault x input.

Bit 30 – TRGSRC Trigger Source

Value	Description
0	The TRGINx signal is driven by the PWMEXTRGx input.
1	The TRGINx signal is driven by the Analog Comparator Controller.

Bit 29 – TRGFILT Filtered input

Value	Description
0	The external trigger input x is not filtered.
1	The external trigger input x is filtered.

Bit 28 – TRGEDGE Edge Selection

Value	Name	Description
0	FALLING_ZERO	TRGMODE = 1: TRGINx event detection on falling edge.

52.5.7 Fault Output

The AFEC has the Fault output connected to the FAULT input of PWM. See Fault Output and implementation of the PWM in the product.

52.5.8 Conversion Performances

For performance and electrical characteristics of the AFE, refer to the AFE Characteristics in the section “Electrical Characteristics”.

Related Links

58. Electrical Characteristics for SAM V70/V71

52.6 Functional Description

52.6.1 Analog Front-End Conversion

The AFE embeds programmable gain amplifiers that must be enabled prior to any conversion. The bits PGA0EN and PGA1EN in the Analog Control register (AFEC_ACR) must be set.

The AFE uses the AFE clock to perform conversions. In order to guarantee a conversion with minimum error, after any start of conversion, the AFEC waits a number of AFE clock cycles (called transfer time) before changing the channel selection again (and so starts a new tracking operation).

AFE conversions are sequenced by two operating times: the tracking time and the conversion time.

- The tracking time represents the time between the channel selection change and the time for the controller to start the AFEC. The AFEC allows a minimum tracking time of 15 AFE clock periods.
- The conversion time represents the time for the AFEC to convert the analog signal.

The AFE clock frequency is selected in the PRESCAL field of the AFEC_MR. The tracking phase starts during the conversion of the previous channel. If the tracking time is longer than the conversion time of the12-bit AD converter (t_{CONV}), the tracking phase is extended to the end of the previous conversion.

The AFE clock frequency ranges from $f_{peripheral\ clock}/2$ if PRESCAL is 1, and $f_{peripheral\ clock}/256$ if PRESCAL is set to 255 (0xFF). PRESCAL must be programmed to provide the AFE clock frequency given in the section “Electrical Characteristics”.

The AFE conversion time (t_{AFE_conv}) is applicable for all modes and is calculated as follows:

$$t_{\mathrm{AFE\_conv}} = 23 \times t_{\mathrm{AFE\ Clock}}$$

When the averager is activated, the AFE conversion time is multiplied by the OSR value.

In Free Run mode, the sampling frequency (f_S) is calculated as $1/t_{AFE\_conv}$.

54. Analog Comparator Controller (ACC)

54.1 Description

The Analog Comparator Controller (ACC) configures the analog comparator and generates an interrupt depending on user settings. The analog comparator embeds two 8-to-1 multiplexers that generate two internal inputs. These inputs are compared, resulting in a compare output. The hysteresis level, edge detection and polarity are configurable.

The ACC also generates a compare event which can be used by the Pulse Width Modulator (PWM).

54.2 Embedded Characteristics

- ANA_INPUTS User Analog Inputs Selectable for Comparison
- VOLT_REF Voltage References Selectable for Comparison: External Voltage Reference, DAC0, DAC1, Temperature Sensor (TS)
- Interrupt Generation
- Compare Event Fault Generation for PWM

54.3 Block Diagram

Figure 54-1. Analog Comparator Controller Block Diagram