Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

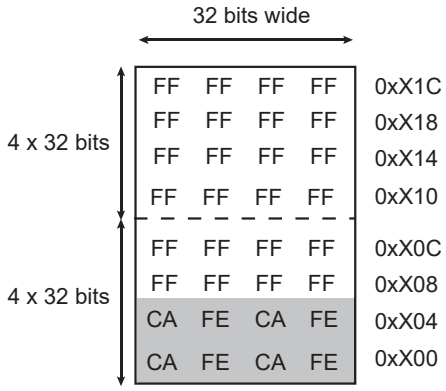
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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"

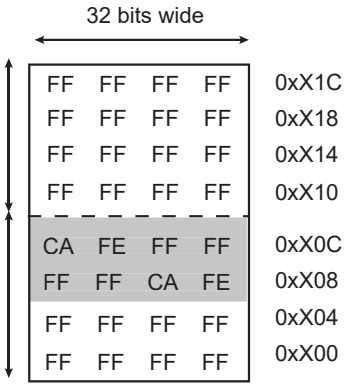
Details	
Product Status	Active
Core Processor	ARM® Cortex®-M7
Core Size	32-Bit Single-Core
Speed	300MHz
Connectivity	CANbus, Ethernet, I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	75
Program Memory Size	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 10x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TFBGA
Supplier Device Package	100-TFBGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsame70n21a-cnt

Signal Name	Function	Type	Active Level	Voltage Reference	Comments
PWMCx_PWMH0 – PWMCx_PWMH3	Waveform Output High for Channel 0–3	Output	–	–	–
PWMCx_PWML0– PWMCx_PWML3	Waveform Output Low for Channel 0–3	Output	–	–	Only output in complementary mode when dead time insertion is enabled.
PWMCx_PWMFI0 – PWMCx_PWMFI2	Fault Input	Input	–	–	–
PWMCx_PWMEX TRG0– PWMCx_PWMEX TRG1	External Trigger Input	Input	–	–	–
Serial Peripheral Interface - SPI(x=[0..1])					
SPIx_MISO	Master In Slave Out	I/O	–	–	–
SPIx_MOSI	Master Out Slave In	I/O	–	–	–
SPIx_SPCK	SPI Serial Clock	I/O	–	–	–
SPIx_NPCS0	SPI Peripheral Chip Select 0	I/O	Low	–	–
SPIx_NPCS1– SPIx_NPCS3	SPI Peripheral Chip Select	Output	Low	–	–
Quad IO SPI - QSPI					
QSCK	QSPI Serial Clock	Output	–	–	–
QCS	QSPI Chip Select	Output	–	–	–
QIO0–QIO3	QSPI I/O QIO0 is QMOSI Master Out Slave In QIO1 is QMISO Master In Slave Out	I/O	–	–	–
Two-Wire Interface - TWIHS(x=0..2)					
TWDx	TWIx Two-wire Serial Data	I/O	–	–	–
TWCKx	TWIx Two-wire Serial Clock	I/O	–	–	–
Analog					

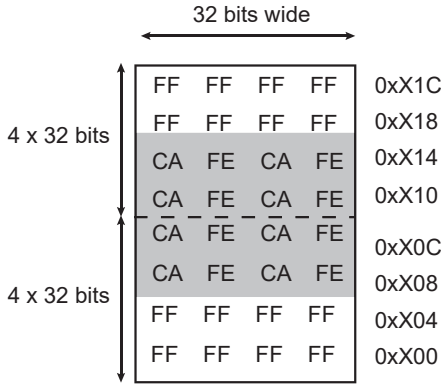
Figure 22-10. Optimized Partial Page Programming



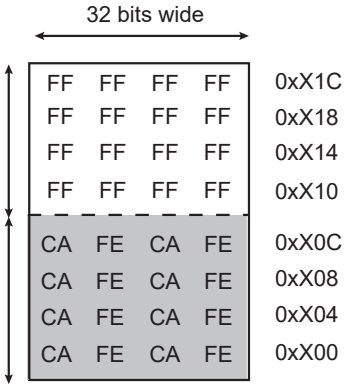
Case 1: 2 x 32 bits modified, not crossing 128-bit boundary
User programs WP, Flash Controller sends Write Word
=> Only 1 word programmed => programming period reduced



Case 2: 2 x 32 bits modified, not crossing 128-bit boundary
User programs WP, Flash Controller sends Write Word
=> Only 1 word programmed => programming period reduced



Case 3: 4 x 32 bits modified across 128-bit boundary
User programs WP, Flash Controller sends WP
=> Whole page programmed



Case 4: 4 x 32 bits modified, not crossing 128-bit boundary
User programs WP, Flash Controller sends Write Word
=> Only 1 word programmed => programming period reduced

32.6.1.47 PIO Write Protection Status Register

Name: PIO_WPSR
Offset: 0x00E8
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	WPVSRC[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

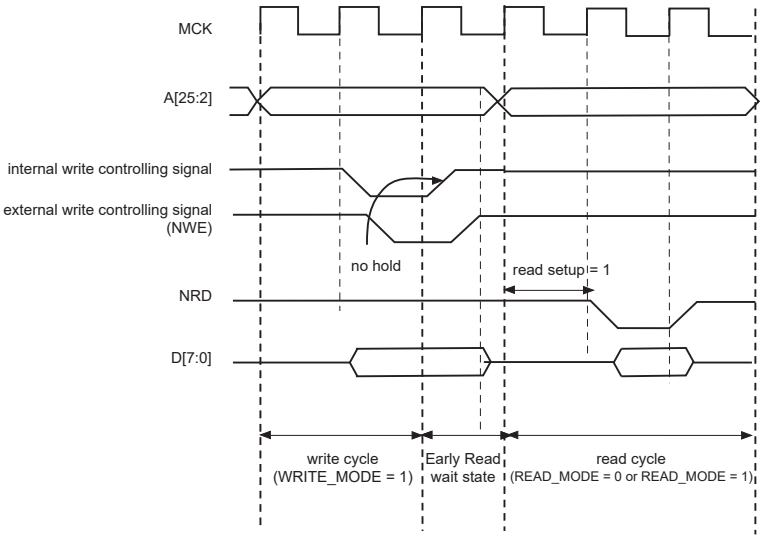
Bit	7	6	5	4	3	2	1	0
								WPVS
Access								
Reset								0

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the PIO_WPSR.
1	A write protection violation has occurred since the last read of the PIO_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

Figure 35-20. Early Read Wait State: NWE-controlled write with no hold followed by a read with one set-up cycle



35.11.3 Reload User Configuration Wait State

The user may change any of the configuration parameters by writing the SMC user interface.

When detecting that a new user configuration has been written in the user interface, the SMC inserts a wait state before starting the next access. This “reload user configuration wait state” is used by the SMC to load the new set of parameters to apply to next accesses.

The reload configuration wait state is not applied in addition to the chip select wait state. If accesses before and after re-programming the user interface are made to different devices (chip selects), then one single chip select wait state is applied.

On the other hand, if accesses before and after writing the user interface are made to the same device, a reload configuration wait state is inserted, even if the change does not concern the current chip select.

35.11.3.1 User Procedure

To insert a reload configuration wait state, the SMC detects a write access to any SMC_MODE register of the user interface. If the user only modifies timing registers (SMC_SETUP, SMC_PULSE, SMC_CYCLE registers) in the user interface, he must validate the modification by writing the SMC_MODE, even if no change was made on the mode parameters.

The user must not change the configuration parameters of an SMC chip select (Setup, Pulse, Cycle, Mode) if accesses are performed on this CS during the modification. Any change of the chip select parameters, while fetching the code from a memory connected on this CS, may lead to unpredictable behavior. The instructions used to modify the parameters of an SMC chip select can be executed from the internal RAM or from a memory connected to another CS.

35.11.3.2 Slow Clock Mode Transition

A reload configuration wait state is also inserted when the Slow Clock mode is entered or exited, after the end of the current transfer (see ”Slow Clock Mode”).

35.11.4 Read to Write Wait State

Due to an internal mechanism, a wait cycle is always inserted between consecutive read and write SMC accesses.

35.16.1.7 SMC Off-Chip Memory Scrambling Key2 Register

Name: SMC_KEY2
Offset: 0x88
Reset: 0x00000000
Property: Write-once

Notes: 1. ‘Write-once’ access indicates that the first write access after a system reset prevents any further modification of the value of this register.

Bit	31	30	29	28	27	26	25	24
	KEY2[31:24]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	KEY2[23:16]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	KEY2[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	KEY2[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – KEY2[31:0] Off-Chip Memory Scrambling (OCMS) Key Part 2
When off-chip memory scrambling is enabled, KEY1 and KEY2 values determine data scrambling.

36.9.11 XDMAC Global Channel Read Suspend Register

Name: XDMAC_GRS
Offset: 0x28
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	RS23	RS22	RS21	RS20	RS19	RS18	RS17	RS16
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
	RS15	RS14	RS13	RS12	RS11	RS10	RS9	RS8
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
	RS7	RS6	RS5	RS4	RS3	RS2	RS1	RS0
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 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23 – RSx XDMAC Channel x Read Suspend

Value	Description
0	The read channel is not suspended.
1	The source requests for channel n are no longer serviced by the system scheduler.

38.8 Register Summary

Offset	Name	Bit Pos.								
0x00	GMAC_NCR	7:0	WESTAT	INCSTAT	CLRSTAT	MPE	TXEN	RXEN	LBL	
		15:8	SRTSM			TXZQPF	TXPF	THALT	TSTART	BP
		23:16						FNP	TXPBPF	ENPBPR
		31:24								
0x04	GMAC_NCFGR	7:0	UNIHEN	MTIHEN	NBC	CAF	JFRAME	DNVLAN	FD	SPD
		15:8	RXBUFO[1:0]		PEN	RTY				MAXFS
		23:16	DCPF	DBW[1:0]		CLK[2:0]			RFCS	LFERD
		31:24		IRXER	RXBP	IPGSEN		IRXFCS	EFRHD	RXCOEN
0x08	GMAC_NSR	7:0						IDLE	MDIO	
		15:8								
		23:16								
		31:24								
0x0C	GMAC_UR	7:0								
		15:8								
		23:16								
		31:24								
0x10	GMAC_DCFGR	7:0	ESPA	ESMA		FBLDO[4:0]				
		15:8					TXCOEN	TXPBMS	RXBMS[1:0]	
		23:16	DRBS[7:0]							
		31:24								DDRP
0x14	GMAC_TSR	7:0			TXCOMP	TFC	TXGO	RLE	COL	UBR
		15:8								HRESP
		23:16								
		31:24								
0x18	GMAC_RBQB	7:0	ADDR[5:0]							
		15:8	ADDR[13:6]							
		23:16	ADDR[21:14]							
		31:24	ADDR[29:22]							
0x1C	GMAC_TBQB	7:0	ADDR[5:0]							
		15:8	ADDR[13:6]							
		23:16	ADDR[21:14]							
		31:24	ADDR[29:22]							
0x20	GMAC_RSR	7:0					HNO	RXOVR	REC	BNA
		15:8								
		23:16								
		31:24								
0x24	GMAC_ISR	7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
		15:8		PFTR	PTZ	PFNZ	HRESP	ROVR		
		23:16	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		
		31:24			TSUTIMCMP	WOL	RXLPISBC	SRI	PDRSFT	PDRQFT
0x28	GMAC_IER	7:0	TCOMP	TFC	RLEX	TUR	TXUBR	RXUBR	RCOMP	MFS
		15:8	EXINT	PFTR	PTZ	PFNZ	HRESP	ROVR		
		23:16	PDRSFR	PDRQFR	SFT	DRQFT	SFR	DRQFR		

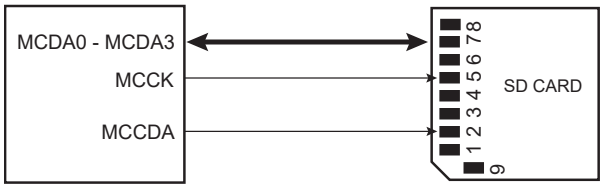
Table 40-3. SD Memory Card Bus Signals

Pin Number	Name	Type(1)	Description	HSMCI Pin Name(2) (Slot z)
1	CD/DAT[3]	I/O/PP	Card detect/ Data line Bit 3	MCDz3
2	CMD	PP	Command/response	MCCDz
3	VSS1	S	Supply voltage ground	VSS
4	VDD	S	Supply voltage	VDD
5	CLK	I/O	Clock	MCCK
6	VSS2	S	Supply voltage ground	VSS
7	DAT[0]	I/O/PP	Data line Bit 0	MCDz0
8	DAT[1]	I/O/PP	Data line Bit 1 or Interrupt	MCDz1
9	DAT[2]	I/O/PP	Data line Bit 2	MCDz2

Notes: 1. I: input, O: output, PP: Push Pull, OD: Open Drain.

2. When several HSMCI (x HSMCI) are embedded in a product, MCCK refers to HSMCIx_CK, MCCDA to HSMCIx_CDA, MCDAy to HSMCIx_DAy.

Figure 40-6. SD Card Bus Connections with One Slot



Note: When several HSMCI (x HSMCI) are embedded in a product, MCCK refers to HSMCIx_CK, MCCDA to HSMCIx_CDA MCDAy to HSMCIx_DAy.

When the HSMCI is configured to operate with SD memory cards, the width of the data bus can be selected in the HSMCI_SDCR. Clearing the SDCBUS bit in this register means that the width is one bit; setting it means that the width is four bits. In the case of High Speed MultiMedia cards, only the data line 0 is used. The other data lines can be used as independent PIOs.

40.8 High-Speed Multimedia Card Operations

After a power-on reset, the cards are initialized by a special message-based High-Speed Multimedia Card bus protocol. Each message is represented by one of the following tokens:

- Command—A command is a token that starts an operation. A command is sent from the host either to a single card (addressed command) or to all connected cards (broadcast command). A command is transferred serially on the CMD line.
- Response—A response is a token which is sent from an addressed card or (synchronously) from all connected cards to the host as an answer to a previously received command. A response is transferred serially on the CMD line.
- Data—Data can be transferred from the card to the host or vice versa. Data is transferred via the data line.

WARNING In SDIO Byte and Block modes (TRTYP = 4 or 5), writing the 7 last bits of BCNT field with a value which differs from 0 is forbidden and may lead to unpredictable results.

40.14.8 HSMCI Completion Signal Timeout Register

Name: HSMCI_CSTOR
Offset: 0x1C
Reset: 0x0
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the HSMCI 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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		CSTOMUL[2:0]			CSTOCYC[3:0]			
Access								
Reset		0	0	0	0	0	0	0

Bits 6:4 – CSTOMUL[2:0] Completion Signal Timeout Multiplier
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).

These fields determine the maximum number of Master Clock cycles that the HSMCI waits between the end of the data transfer and the assertion of the completion signal. The data transfer comprises data phase and the optional busy phase. If a non-DATA ATA command is issued, the HSMCI starts waiting immediately after the end of the response until the completion signal.

Multiplier is defined by CSTOMUL as shown in the following table:

If the data time-out set by CSTOCYC and CSTOMUL has been exceeded, the Completion Signal Time-out Error flag (CSTOE) in the HSMCI Status Register (HSMCI_SR) rises.

Value	Name	Description
0	1	CSTOCYC x 1
1	16	CSTOCYC x 16
2	128	CSTOCYC x 128
3	256	CSTOCYC x 256
4	1024	CSTOCYC x 1024
5	4096	CSTOCYC x 4096

Bit 7 – UNRE Underrun Error Interrupt Mask

Bit 6 – OVRE Overrun Error Interrupt Mask

Bit 5 – GACC General Call Access Interrupt Mask

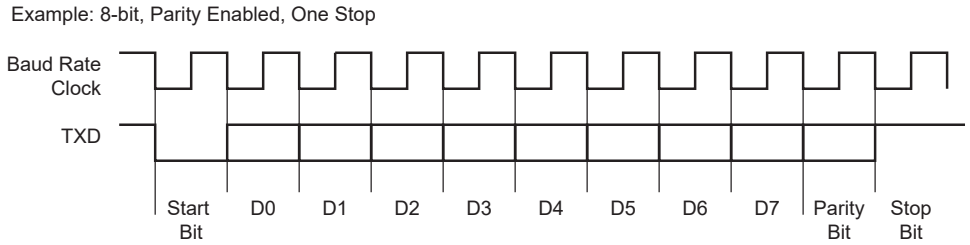
Bit 4 – SVACC Slave Access Interrupt Mask

Bit 2 – TXRDY Transmit Holding Register Ready Interrupt Mask

Bit 1 – RXRDY Receive Holding Register Ready Interrupt Mask

Bit 0 – TXCOMP Transmission Completed Interrupt Mask

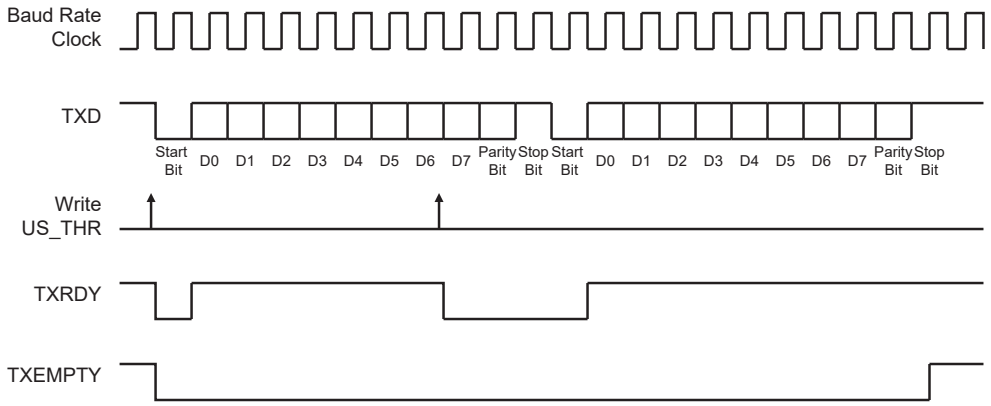
Figure 46-5. Character Transmit



The characters are sent by writing in US_THR. The transmitter reports two status bits in the Channel Status register (US_CSR): TXRDY (Transmitter Ready), which indicates that US_THR is empty, and TXEMPTY, which indicates that all the characters written in US_THR have been processed. When the current character processing is completed, the last character written in US_THR is transferred into the Shift register of the transmitter and US_THR becomes empty, thus TXRDY rises.

Both TXRDY and TXEMPTY are low when the transmitter is disabled. Writing a character in US_THR while TXRDY is low has no effect and the written character is lost.

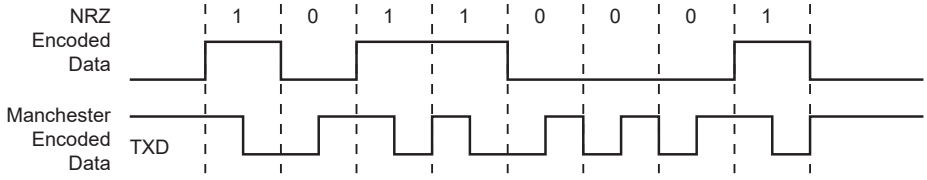
Figure 46-6. Transmitter Status



46.6.3.2 Manchester Encoder

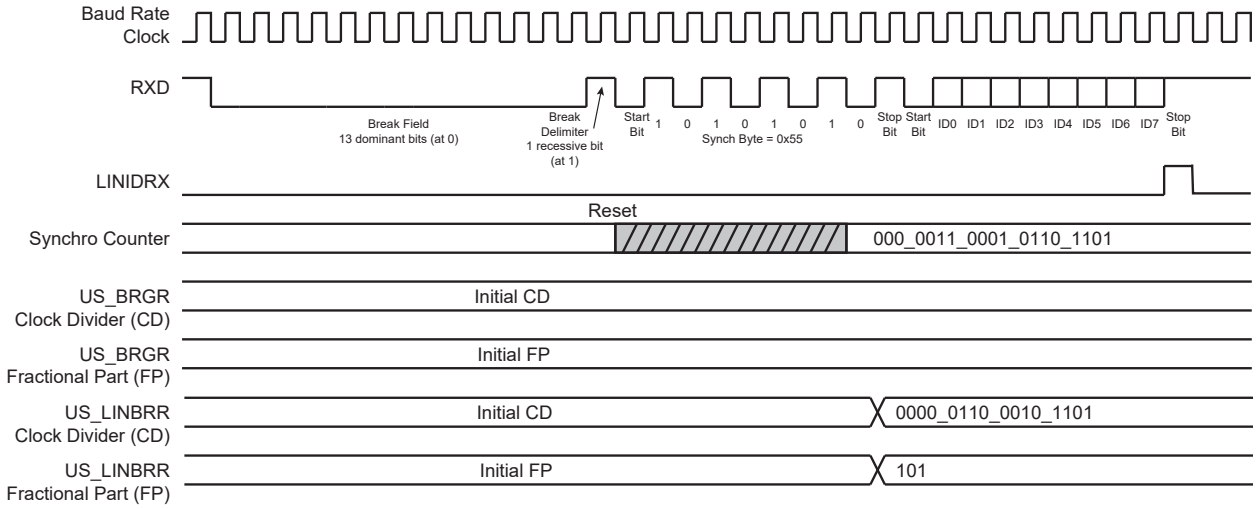
When the Manchester encoder is in use, characters transmitted through the USART are encoded based on biphase Manchester II format. To enable this mode, write a '1' to USART_MR.MAN. Depending on polarity configuration, a logic level (zero or one), is transmitted as a coded signal one-to-zero or zero-to-one. Thus, a transition always occurs at the midpoint of each bit time. It consumes more bandwidth than the original NRZ signal (2x) but the receiver has more error control since the expected input must show a change at the center of a bit cell. An example of Manchester encoded sequence is: the byte 0xB1 or 10110001 encodes to 10 01 10 10 01 01 01 10, assuming the default polarity of the encoder. Figure 46-7 illustrates this coding scheme.

Figure 46-7. NRZ to Manchester Encoding



The Manchester encoded character can also be encapsulated by adding both a configurable preamble and a start frame delimiter pattern. Depending on the configuration, the preamble is a training sequence, composed of a predefined pattern with a programmable length from 1 to 15 bit times. If the preamble length is set to '0', the preamble waveform is not generated prior to any character. The preamble pattern

Figure 46-42. Slave Node Synchronization



The accuracy of the synchronization depends on several parameters:

- Nominal clock frequency (f_{Nom}) (the theoretical slave node clock frequency)
- Baud Rate
- Oversampling (OVER = 0 => 16X or OVER = 1 => 8X)

The following formula is used to compute the deviation of the slave bit rate relative to the master bit rate after synchronization (f_{SLAVE} is the real slave node clock frequency):

$$\text{Baud rate deviation} = \left(100 \times \frac{[\alpha \times 8 \times (2 - \text{OVER}) + \beta] \times \text{Baud rate}}{8 \times f_{SLAVE}}\right)\%$$

$$\text{Baud rate deviation} = \left(100 \times \frac{[\alpha \times 8 \times (2 - \text{OVER}) + \beta] \times \text{Baud rate}}{8 \times \left(\frac{f_{TOL\_UNSYNCH}}{100}\right) \times f_{Nom}}\right)\%$$

$-0.5 \leq \alpha \leq +0.5 \quad -1 < \beta < +1$

$f_{TOL\_UNSYNCH}$ is the deviation of the real slave node clock from the nominal clock frequency. The LIN Standard imposes that it must not exceed ±15%. The LIN Standard imposes also that for communication between two nodes, their bit rate must not differ by more than ±2%. This means that the baud rate deviation must not exceed ±1%.

It follows from that, a minimum value for the nominal clock frequency:

$$f_{Nom}(\text{min}) = \left(100 \times \frac{[0.5 \times 8 \times (2 - \text{OVER}) + 1] \times \text{Baud rate}}{8 \times \left(\frac{-15}{100} + 1\right) \times 1\%}\right)\text{Hz}$$

Examples:

- Baud rate = 20 kbit/s, OVER = 0 (Oversampling 16X) => f_{Nom}(min) = 2.64 MHz
- Baud rate = 20 kbit/s, OVER = 1 (Oversampling 8X) => f_{Nom}(min) = 1.47 MHz
- Baud rate = 1 kbit/s, OVER = 0 (Oversampling 16X) => f_{Nom}(min) = 132 kHz
- Baud rate = 1 kbit/s, OVER = 1 (Oversampling 8X) => f_{Nom}(min) = 74 kHz

Value	Description
0	There are characters in either US_THR or the Transmit Shift Register, or the transmitter is disabled.
1	There are no characters in US_THR, nor in the Transmit Shift Register.

Bit 8 – TIMEOUT Receiver Timeout (cleared by writing a one to bit US_CR.STTTO)

Value	Description
0	There has not been a timeout since the last Start Timeout command (STTTO in US_CR) or the Timeout Register is 0.
1	There has been a timeout since the last Start Timeout command (STTTO in US_CR).

Bit 7 – PARE Parity Error (cleared by writing a one to bit US_CR.RSTSTA)

Value	Description
0	No parity error has been detected since the last RSTSTA.
1	At least one parity error has been detected since the last RSTSTA.

Bit 6 – FRAME Framing Error (cleared by writing a one to bit US_CR.RSTSTA)

Value	Description
0	No stop bit has been detected low since the last RSTSTA.
1	At least one stop bit has been detected low since the last RSTSTA.

Bit 5 – OVRE Overrun Error (cleared by writing a one to bit US_CR.RSTSTA)

Value	Description
0	No overrun error has occurred since the last RSTSTA.
1	At least one overrun error has occurred since the last RSTSTA.

Bit 2 – RXBRK Break Received/End of Break (cleared by writing a one to bit US_CR.RSTSTA)

Value	Description
0	No break received or end of break detected since the last RSTSTA.
1	Break received or end of break detected since the last RSTSTA.

Bit 1 – TXRDY Transmitter Ready (cleared by writing US_THR)

Value	Description
0	A character is in the US_THR waiting to be transferred to the Transmit Shift Register, or an STTBRK command has been requested, or the transmitter is disabled. As soon as the transmitter is enabled, TXRDY becomes 1.
1	There is no character in the US_THR.

Bit 0 – RXRDY Receiver Ready (cleared by reading US_RHR)

Value	Description
0	No complete character has been received since the last read of US_RHR or the receiver is disabled. If characters were being received when the receiver was disabled, RXRDY changes to 1 when the receiver is enabled.
1	At least one complete character has been received and US_RHR has not yet been read.

47.6.2 UART Mode Register

Name: UART_MR
Offset: 0x04
Reset: 0x00000000
Property: Read/Write

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
	CHMODE[1:0]			BRSRCCK	PAR[2:0]			
Access	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	0	0		0	0	0	0	

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

Bits 15:14 – CHMODE[1:0] Channel Mode

Value	Name	Description
0	NORMAL	Normal mode
1	AUTOMATIC	Automatic echo
2	LOCAL_LOOPBACK	Local loopback
3	REMOTE_LOOPBACK	Remote loopback

Bit 12 – BRSRCCK Baud Rate Source Clock
0 (PERIPH_CLK): The baud rate is driven by the peripheral clock

1 (PMC_PCK): The baud rate is driven by a PMC-programmable clock PCK (see section "Power Management Controller (PMC)").

Bits 11:9 – PAR[2:0] Parity Type

Value	Name	Description
0	EVEN	Even Parity
1	ODD	Odd Parity
2	SPACE	Space: parity forced to 0
3	MARK	Mark: parity forced to 1
4	NO	No parity

48. Media Local Bus (MLB)

48.1 Description

The MediaLB (MLB) maps all the MOST Network data types (transport methods) into a single low-cost, scalable, and standardized hardware interface between a MediaLB Controller and at least one other MediaLB Device. The use of MediaLB simplifies the hardware interface, reduces the pin count, and facilitates the design of modular reusable hardware. From a software development perspective, the use of MediaLB relieves the system developer from the complexity of the MOST Network, which simplifies software development and enables the design of reusable software for different applications. This simplified, standardized interface shortens time-to-market and makes software maintenance effortless.

The link layer and three different physical layers are defined as part of this specification. The physical layer section describes pin configurations, operating speeds, and bus topology. The link layer section describes the compliance of the signaling and addressing protocol.

48.1.1 MediaLB Concept

The MediaLB topology supports communication among all MediaLB Devices, including the MediaLB Controller. The bus interface consists of a uni-directional line for clock (MLBC), a bi-directional line for signal information (MLBS), and a bi-directional line for data transfer (MLBD).

The MediaLB topology supports one Controller connected to one or more Devices, where the Controller is the interface between the MediaLB Devices and the MOST Network. The MediaLB Controller includes MediaLB Device functionality, and also generates the MediaLB clock (MLBC) that is synchronized to the MOST Network. This generated clock provides the timing for the entire MediaLB interface. The Controller will continue to generate MLBC even when the Controller loses lock with the MOST Network.

The MLBS line is a multiplexed signal which carries ChannelAddresses generated by the MediaLB Controller, as well as Command and RxStatus bytes from MediaLB Devices. Each ChannelAddress indicates which Device can transmit data and which Device (or Devices) can receive data on a particular logical channel.

The MLBD line is driven by the transmitting MediaLB Device and is received by all other MediaLB Devices, including the MediaLB Controller. The MLBD line carries the actual data (synchronous, asynchronous, control, or isochronous). For synchronous stream data transmission, multiple MediaLB Devices can receive the same data, in a broadcast fashion. The transmitting MediaLB Device indicates the particular type of data transmitted by sending the appropriate command on the MLBS line. The Link Layer section defines the different commands supported.

48.1.2 MediaLB Protocol

Once per MOST Network frame, the MediaLB Controller generates a unique FRAMESYNC pattern on the MLBS line. For all Devices on the bus, the end of the FRAMESYNC pattern defines the byte boundary and the channel boundary for the MLBS and MLBD lines.

Each four-byte wide block (quadlet) in a 3-pin MediaLB frame is defined as a physical channel. Physical channels can be grouped into multiple quadlets (which do not have to be consecutive) to form a logical channel. The MediaLB Controller handles channel arbitration, allocates channel bandwidth for MediaLB Devices, and manages the unique ChannelAddresses for referencing logical channels.

The MediaLB Controller initiates communication with MediaLB Devices by sending an assigned ChannelAddress on MLBS in each logical channel. This ChannelAddress indicates which MediaLB Device will transmit data and which MediaLB Devices will receive data in the following logical channel.

Offset	Name	Bit Pos.								
		31:24	F1OM	F1WM[6:0]						
0xB4	MCAN_RXF1S	7:0		F1FL[6:0]						
		15:8			F1GI[5:0]					
		23:16			F1PI[5:0]					
		31:24	DMS[1:0]						RF1L	F1F
0xB8	MCAN_RXF1A	7:0			F1AI[5:0]					
		15:8								
		23:16								
		31:24								
0xBC	MCAN_RXESC	7:0		F1DS[2:0]				F0DS[2:0]		
		15:8						RBDS[2:0]		
		23:16								
		31:24								
0xC0	MCAN_TXBC	7:0	TBSA[5:0]							
		15:8	TBSA[13:6]							
		23:16			NDTB[5:0]					
		31:24		TFQM	TFQS[5:0]					
0xC4	MCAN_TXFQS	7:0			TFFL[5:0]					
		15:8				TFGI[4:0]				
		23:16			TFQF	TFQPI[4:0]				
		31:24								
0xC8	MCAN_TXESC	7:0						TBDS[2:0]		
		15:8								
		23:16								
		31:24								
0xCC	MCAN_TXBRP	7:0	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
		15:8	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8
		23:16	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16
		31:24	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24
0xD0	MCAN_TXBAR	7:0	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
		15:8	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8
		23:16	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
		31:24	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24
0xD4	MCAN_TXBCR	7:0	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
		15:8	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8
		23:16	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16
		31:24	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24
0xD8	MCAN_TXBTO	7:0	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
		15:8	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8
		23:16	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16
		31:24	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24
0xDC	MCAN_TXBCF	7:0	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
		15:8	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8
		23:16	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16
		31:24	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24
0xE0	MCAN_TXBTIE	7:0	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0

53.7.14 DACC Write Protection Status Register

Name: DACC_WPSR
Offset: 0xE8
Reset: 0x00000000
Property: Read-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
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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

Bits 15:8 – WPVSRC[7:0] Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

Bit 0 – WPVS Write Protection Violation Status

Value	Description
0	No write protection violation has occurred since the last read of the DACC_WPSR.
1	A write protection violation has occurred since the last read of the DACC_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

55.6.11 ICM User Initial Hash Value Register

Name: ICM_UIHVALx
Offset: 0x38 + x*0x04 [x=0..7]
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
	VAL[31:24]							
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
	VAL[23:16]							
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
	VAL[15:8]							
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
	VAL[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	–

Bits 31:0 – VAL[31:0] Initial Hash Value
When ICM_CFG.UIHASH is set, the Initial Hash Value is user-programmable.

To meet the desired standard, use the following example values.

For ICM_UIHVAL0 field:

Example	Comment
0x67452301	SHA1 algorithm
0xC1059ED8	SHA224 algorithm
0x6A09E667	SHA256 algorithm

For ICM_UIHVAL1 field:

Example	Comment
0xEFCDAB89	SHA1 algorithm
0x367CD507	SHA224 algorithm
0xBB67AE85	SHA256 algorithm

For ICM_UIHVAL2 field:

Date	Changes
	Added Figure 53-2, “Conversion Sequence in Trigger Mode”and Figure 53-3, “Conversion Sequence in Free-running Mode”. Section 53.6.4.1 “Trigger Mode”: removed fragment ‘(either DATRG pin or timer counter events)’. Section 53.6.4.2 “Free-Running Mode”: added sentence on FIFO. Updated Figure 53-3, “Conversion Sequence in Free-running Mode”. Updated Section 53.6.4.3 “Max Speed Mode” and added Figure 53-4, “Conversion Sequence in Max Speed Mode”. Updated Section 53.6.4.4 “Bypass Mode”. Deleted section “DACC Timings”. Table 53-4 “Register Mapping”: modified reset value for DACC_MR. Section 53.7.2 “DACC Mode Register”: added bit ZERO (bit 5) and bit description. Section 53.7.3 “DACC Trigger Register”: bit description changed for TRGSEL bit. Removed bits ENDTX0, ENDTX1, TXBUFE0 and TXBUFE1 from Section 53.7.8 “DACC Interrupt Enable Register”, Section 53.7.9 “DACC Interrupt Disable Register”, Section 53.7.10 “DACC Interrupt Mask Register” and Section 53.7.11 “DACC Interrupt Status Register”.
	Section 55. “Integrity Check Monitor (ICM)” Section 55.5.2.2 “ICM Region Configuration Structure Member”: removed MRPROT field. Section 55.6.1 “ICM Configuration Register”: removed fields HAPROT and DAPROT; updated description DUALBUFF field
	Updated Section 58. “Electrical Characteristics”.
	Updated Section 59. “Mechanical Characteristics”.
	Added Section 60. “Schematic Checklist”.
	Added Section 63. “Errata”.

Table 62-5. SAM E70/S70/V70/V71 Datasheet Rev. 44003B – Revision History

Date	Changes
24-Feb-15	“Description”: updated details on PWM, 16-bit timers, RTC, RTT and Backup mode. Added note to QFN64 package on availability.
	“Features”: updated details on PWM. Added note to QFN64 package on availability.
	Section 1. “Configuration Summary” Table 1-1 “Configuration Summary”: Modifications made to Timer Counter Channels I/O, USART/UART, QSPI, SPI, USART SPI.
	Section 2. “Block Diagram”: added AHBP block. Added Backup RAM block. Removed TRACECTL. Changed block name to Serial Wire Debug/JTAG Boundary Scan (was JTAG