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	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 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/atsame70n20b-cnt

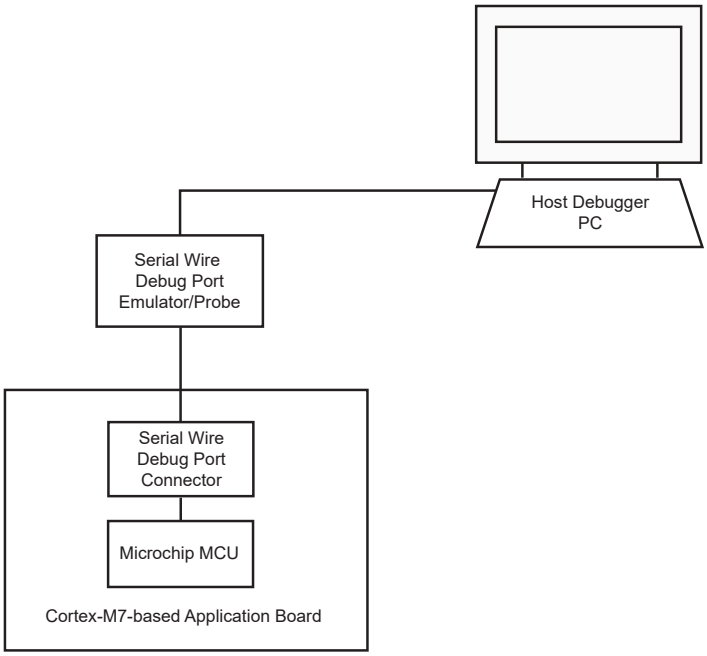
Signal Name	Function	Type	Active Level
TRACECLK	Trace Clock	Output	–
TRACED0–3	Trace Data	Output	–

16.6 Application Examples

16.6.1 Debug Environment

The figure below shows a complete debug environment example. The SW-DP interface is used for standard debugging functions, such as downloading code and single-stepping through the program and viewing core and peripheral registers.

Figure 16-2. Application Debug Environment Example



16.6.2 Test Environment

The figure below shows a test environment example (JTAG Boundary scan). Test vectors are sent and interpreted by the tester. In this example, the “board in test” is designed using a number of JTAG-compliant devices. These devices can be connected to form a single scan chain.

28.5.3 Real-time Timer Value Register

Name: RTT_VR
Offset: 0x08
Reset: 0x00000000
Property: Read-only

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

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

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

Bits 31:0 – CRTV[31:0] Current Real-time Value
Returns the current value of the RTT.
As CRTV can be updated asynchronously, it must be read twice at the same value.

Value	Description
0	Main crystal oscillator is not stabilized.
1	Main crystal oscillator is stabilized.

32.6.1.45 PIO Lock Status Register

Name: PIO_LOCKSR
Offset: 0x00E0
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	P31	P30	P29	P28	P27	P26	P25	P24
Access								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	P23	P22	P21	P20	P19	P18	P17	P16
Access								
Reset	0	0	0	0	0	0	0	0

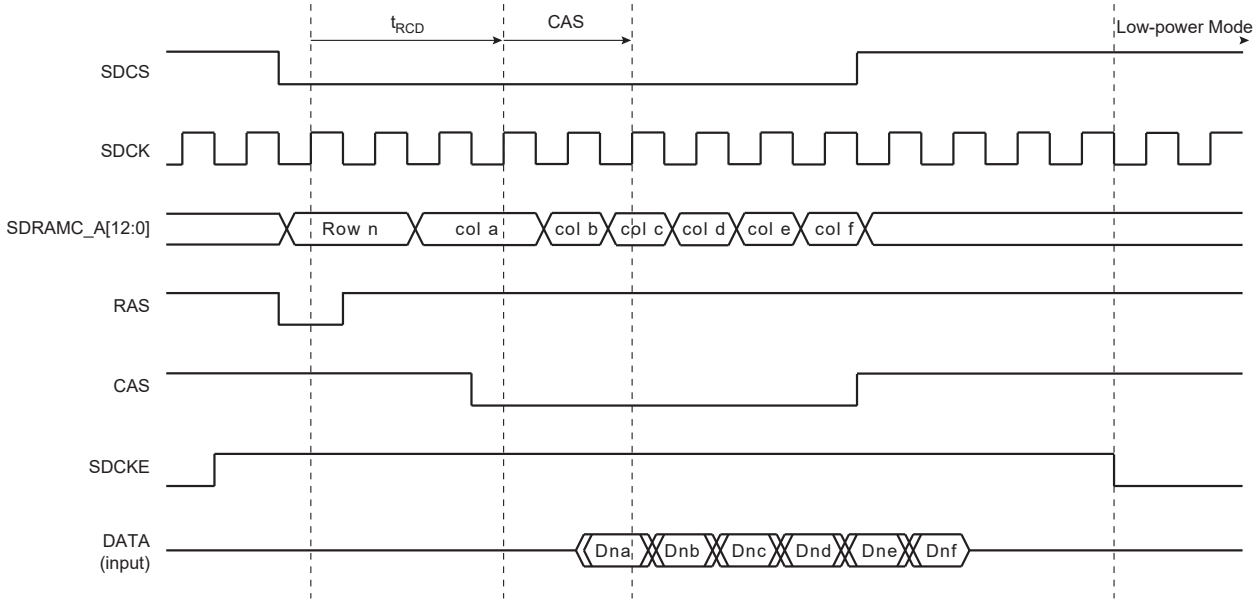
Bit	15	14	13	12	11	10	9	8
	P15	P14	P13	P12	P11	P10	P9	P8
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access								
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, 24, 25, 26, 27, 28, 29, 30, 31 – P PIO Lock Status

Value	Description
0	The I/O line is not locked.
1	The I/O line is locked.

Figure 34-7. Low-power Mode Behavior



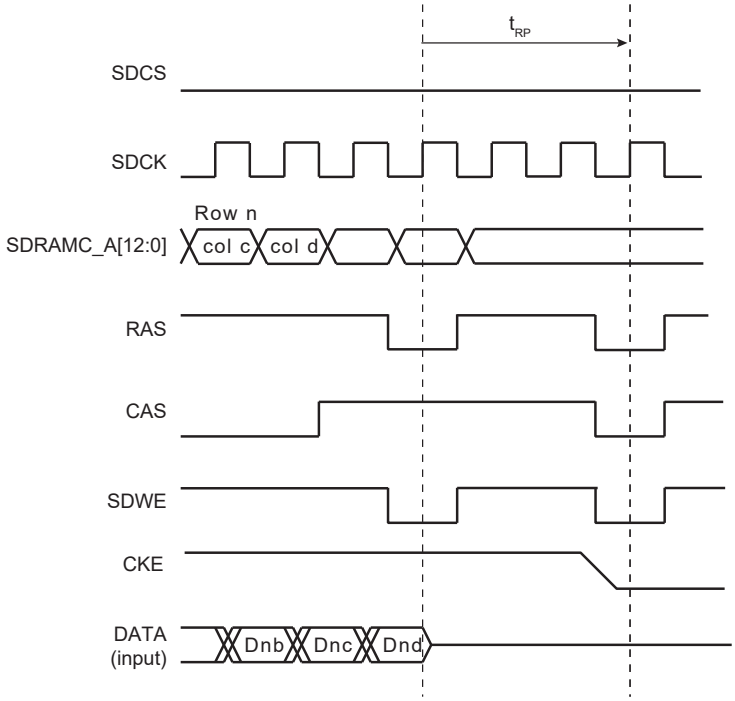
34.6.5.3 Deep Powerdown Mode

This mode is selected by configuring SDRAMC_LPR.LPCB to 3. When this mode is activated, all internal voltage generators inside the SDRAM are stopped and all data is lost.

When this mode is enabled, the application must not access the SDRAM until a new initialization sequence is done (see “SDRAM Device Initialization”).

Refer to the following figure.

Figure 34-8. Deep Powerdown Mode Behavior

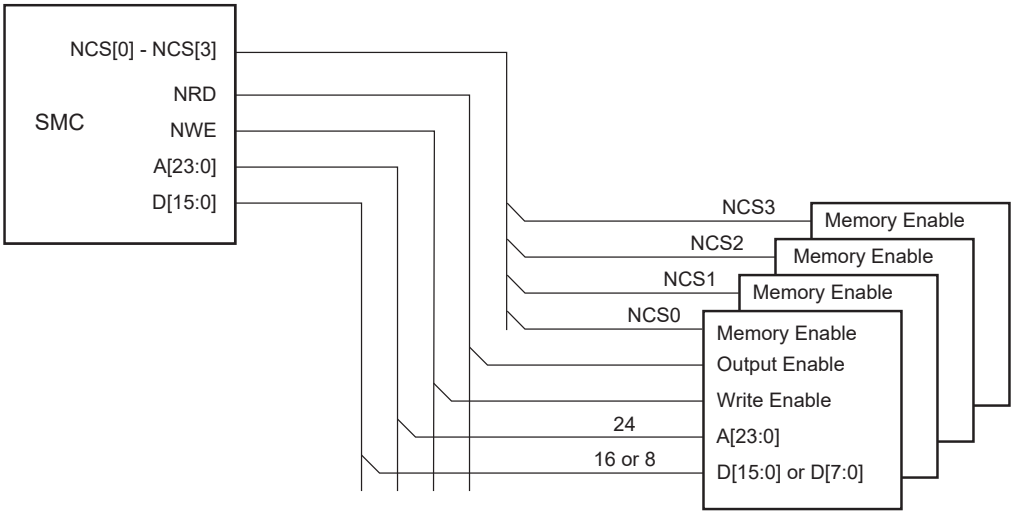


35.6 External Memory Mapping

The SMC provides up to 24 address lines, A[23:0]. This allows each chip select line to address up to 16 Mbytes of memory.

If the physical memory device connected on one chip select is smaller than 16 Mbytes, it wraps around and appears to be repeated within this space. The SMC correctly handles any valid access to the memory device within the page (see the following figure).

Figure 35-1. Memory Connections for Four External Devices



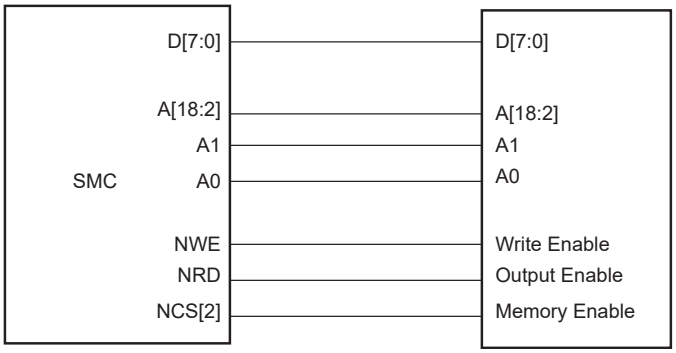
35.7 Connection to External Devices

35.7.1 Data Bus Width

A data bus width of 8 or 16 bits can be selected for each chip select. This option is controlled by the bit DBW in the Mode register (SMC_MODE) for the corresponding chip select.

Figure 35-2 shows how to connect a 512-Kbyte × 8-bit memory on NCS2. Figure 35-3 shows how to connect a 512-Kbyte × 16-bit memory on NCS2.

Figure 35-2. Memory Connection for an 8-bit Data Bus



Offset	Name	Bit Pos.								
		23:16	SA[23:16]							
		31:24	SA[31:24]							
0x03E4	XDMAC_CDA14	7:0	DA[7:0]							
		15:8	DA[15:8]							
		23:16	DA[23:16]							
		31:24	DA[31:24]							
0x03E8	XDMAC_CNDA14	7:0	NDA[5:0]							NDAIF
		15:8	NDA[13:6]							
		23:16	NDA[21:14]							
		31:24	NDA[29:22]							
0x03EC	XDMAC_CNDC14	7:0				NDVIEW[1:0]		NDDUP	NDSUP	NDE
		15:8								
		23:16								
		31:24								
0x03F0	XDMAC_CUBC14	7:0	UBLEN[7:0]							
		15:8	UBLEN[15:8]							
		23:16	UBLEN[23:16]							
		31:24								
0x03F4	XDMAC_CBC14	7:0	BLEN[7:0]							
		15:8					BLEN[11:8]			
		23:16								
		31:24								
0x03F8	XDMAC_CC14	7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		31:24		PERID[6:0]						
0x03FC	XDMAC_CDS_MSP14	7:0	SDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		31:24	DDS_MSP[15:8]							
0x0400	XDMAC_CSUS14	7:0	SUBS[7:0]							
		15:8	SUBS[15:8]							
		23:16	SUBS[23:16]							
		31:24								
0x0404	XDMAC_CDUS14	7:0	DUBS[7:0]							
		15:8	DUBS[15:8]							
		23:16	DUBS[23:16]							
		31:24								
0x0408 ... 0x040F	Reserved									
0x0410	XDMAC_CIE15	7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
		15:8								
		23:16								
		31:24								
0x0414	XDMAC_CID15	7:0		ROID	WBEID	RBEID	FID	DID	LID	BID

36.9.1 XDMAC Global Type Register

Name: XDMAC_GTYPE
Offset: 0x00
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
		NB_REQ[6:0]						
Access		R	R	R	R	R	R	R
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	FIFO_SZ[10:3]							
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
	FIFO_SZ[2:0]			NB_CH[4:0]				
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 22:16 – NB_REQ[6:0] Number of Peripheral Requests Minus One

Bits 15:5 – FIFO_SZ[10:0] Number of Bytes

Bits 4:0 – NB_CH[4:0] Number of Channels Minus One

37.6.16 DMA Channel Disable Register

Name: ISI_DMA_CHDR
Offset: 0x3C
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
							C_CH_DIS	P_CH_DIS
Access							W	W
Reset							–	–

Bit 1 – C_CH_DIS Codec Channel Disable Request

Value	Description
0	No effect.
1	Disables the channel. Poll C_CH_S in DMA_CHSR to verify that the codec channel status has been successfully modified.

Bit 0 – P_CH_DIS Preview Channel Disable Request

Value	Description
0	No effect.
1	Disables the channel. Poll P_CH_S in DMA_CHSR to verify that the preview channel status has been successfully modified.

- No IP fragmentation
- TCP or UDP packet

When an IP, TCP or UDP frame is received, the receive buffer descriptor gives an indication if the GMAC was able to verify the checksums. There is also an indication if the frame had SNAP encapsulation. These indication bits will replace the type ID match indication bits when the receive checksum offload is enabled. For details of these indication bits refer to “Receive Buffer Descriptor Entry”.

If any of the checksums are verified as incorrect by the GMAC, the packet is discarded and the appropriate statistics counter incremented.

38.6.6.2 Transmitter Checksum Offload

The transmitter checksum offload is only available if the full store and forward mode is enabled. This is because the complete frame to be transmitted must be read into the packet buffer memory before the checksum can be calculated and written back into the headers at the beginning of the frame.

Transmitter checksum offload is enabled by setting bit [11] in the DMA Configuration register. When enabled, it will monitor the frame as it is written into the transmitter packet buffer memory to automatically detect the protocol of the frame. Protocol support is identical to the receiver checksum offload.

For transmit checksum generation and substitution to occur, the protocol of the frame must be recognized and the frame must be provided without the FCS field, by making sure that bit [16] of the transmit descriptor word 1 is clear. If the frame data already had the FCS field, this would be corrupted by the substitution of the new checksum fields.

If these conditions are met, the transmit checksum offload engine will calculate the IP, TCP and UDP checksums as appropriate. Once the full packet is completely written into packet buffer memory, the checksums will be valid and the relevant DPRAM locations will be updated for the new checksum fields as per standard IP/TCP and UDP packet structures.

If the transmitter checksum engine is prevented from generating the relevant checksums, bits [22:20] of the transmitter DMA writeback status will be updated to identify the reason for the error. Note that the frame will still be transmitted but without the checksum substitution, as typically the reason that the substitution did not occur was that the protocol was not recognized.

38.6.7 MAC Filtering Block

The filter block determines which frames should be written to the FIFO interface and on to the DMA.

Whether a frame is passed depends on what is enabled in the Network Configuration register, the state of the external matching pins, the contents of the specific address, type and Hash registers and the frame's destination address and type field.

If bit 25 of the Network Configuration register is not set, a frame will not be copied to memory if the GMAC is transmitting in half duplex mode at the time a destination address is received.

Ethernet frames are transmitted a byte at a time, least significant bit first. The first six bytes (48 bits) of an Ethernet frame make up the destination address. The first bit of the destination address, which is the LSB of the first byte of the frame, is the group or individual bit. This is one for multicast addresses and zero for unicast. The all ones address is the broadcast address and a special case of multicast.

The GMAC supports recognition of four specific addresses. Each specific address requires two registers, Specific Address register Bottom and Specific Address register Top. Specific Address register Bottom stores the first four bytes of the destination address and Specific Address register Top contains the last two bytes. The addresses stored can be specific, group, local or universal.

38.8.46 GMAC 128 to 255 Byte Frames Transmitted Register

Name: GMAC_TBFT255
Offset: 0x120
Reset: 0x00000000
Property: -

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

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

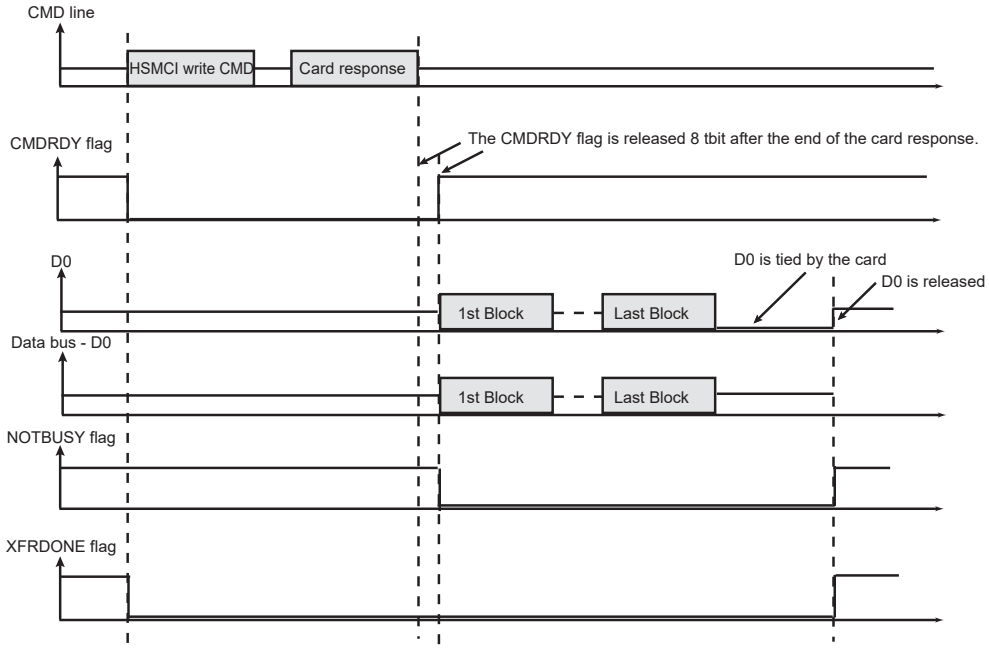
Bit	15	14	13	12	11	10	9	8
	NFTX[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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

Bits 31:0 – NFTX[31:0] 128 to 255 Byte Frames Transmitted without Error
This register counts the number of 128 to 255 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

Offset	Name	Bit Pos.								
		15:8		FIFOCON	KILLBK	NBUSYBKE		ERRORTRANSE	DATAXE	MDATAE
		23:16						RSTDT		EPDISHDMA
		31:24								
0x01E8 ... 0x01EF	Reserved									
0x01F0	USBHS_DEVEPTIER0	7:0	SHORTPACKETES	STALLEDES	OVERFES	NAKINES	NAKOUTES	RXSTPES	RXOUTES	TXINES
		15:8		FIFOCONS	KILLBKS	NBUSYBKES				
		23:16					STALLRQS	RSTDTS	NYETDISS	EPDISHDMAS
		31:24								
0x01F0	USBHS_DEVEPTIER0 (ISOENPT)	7:0	SHORTPACKETES	CRCERRES	OVERFES	HBISOFLUSHES	HBISOINERRES	UNDERFES	RXOUTES	TXINES
		15:8		FIFOCONS	KILLBKS	NBUSYBKES		ERRORTRANSES	DATAXES	MDATAES
		23:16						RSTDTS		EPDISHDMAS
		31:24								
0x01F4	USBHS_DEVEPTIER1	7:0	SHORTPACKETES	STALLEDES	OVERFES	NAKINES	NAKOUTES	RXSTPES	RXOUTES	TXINES
		15:8		FIFOCONS	KILLBKS	NBUSYBKES				
		23:16					STALLRQS	RSTDTS	NYETDISS	EPDISHDMAS
		31:24								
0x01F4	USBHS_DEVEPTIER1 (ISOENPT)	7:0	SHORTPACKETES	CRCERRES	OVERFES	HBISOFLUSHES	HBISOINERRES	UNDERFES	RXOUTES	TXINES
		15:8		FIFOCONS	KILLBKS	NBUSYBKES		ERRORTRANSES	DATAXES	MDATAES
		23:16						RSTDTS		EPDISHDMAS
		31:24								
0x01F8	USBHS_DEVEPTIER2	7:0	SHORTPACKETES	STALLEDES	OVERFES	NAKINES	NAKOUTES	RXSTPES	RXOUTES	TXINES
		15:8		FIFOCONS	KILLBKS	NBUSYBKES				
		23:16					STALLRQS	RSTDTS	NYETDISS	EPDISHDMAS
		31:24								
0x01F8	USBHS_DEVEPTIER2 (ISOENPT)	7:0	SHORTPACKETES	CRCERRES	OVERFES	HBISOFLUSHES	HBISOINERRES	UNDERFES	RXOUTES	TXINES
		15:8		FIFOCONS	KILLBKS	NBUSYBKES		ERRORTRANSES	DATAXES	MDATAES
		23:16						RSTDTS		EPDISHDMAS
		31:24								

Figure 40-12. XFRDONE During a Write Access



40.13 Register Write Protection

To prevent any single software error from corrupting HSMCI behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the HSMCI Write Protection Mode Register (HSMCI_WPMR).

If a write access to a write-protected register is detected, the WPVS bit in the HSMCI Write Protection Status Register (HSMCI_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the HSMCI_WPSR.

The following registers can be protected:

- HSMCI Mode Register
- HSMCI Data Timeout Register
- HSMCI SDCard/SDIO Register
- HSMCI Completion Signal Timeout Register
- HSMCI DMA Configuration Register
- HSMCI Configuration Register

44. Synchronous Serial Controller (SSC)

44.1 Description

The Synchronous Serial Controller (SSC) provides a synchronous communication link with external devices. It supports many serial synchronous communication protocols generally used in audio and telecom applications such as I2S, Short Frame Sync, Long Frame Sync, etc.

The SSC contains an independent receiver and transmitter and a common clock divider. The receiver and the transmitter each interface with three signals: the TD/RD signal for data, the TK/RK signal for the clock and the TF/RF signal for the Frame Sync. The transfers can be programmed to start automatically or on different events detected on the Frame Sync signal.

The SSC high-level of programmability and its use of DMA enable a continuous high bit rate data transfer without processor intervention.

Featuring connection to the DMA, the SSC enables interfacing with low processor overhead to:

- Codecs in Master or Slave mode
- DAC through dedicated serial interface, particularly I2S
- Magnetic card reader

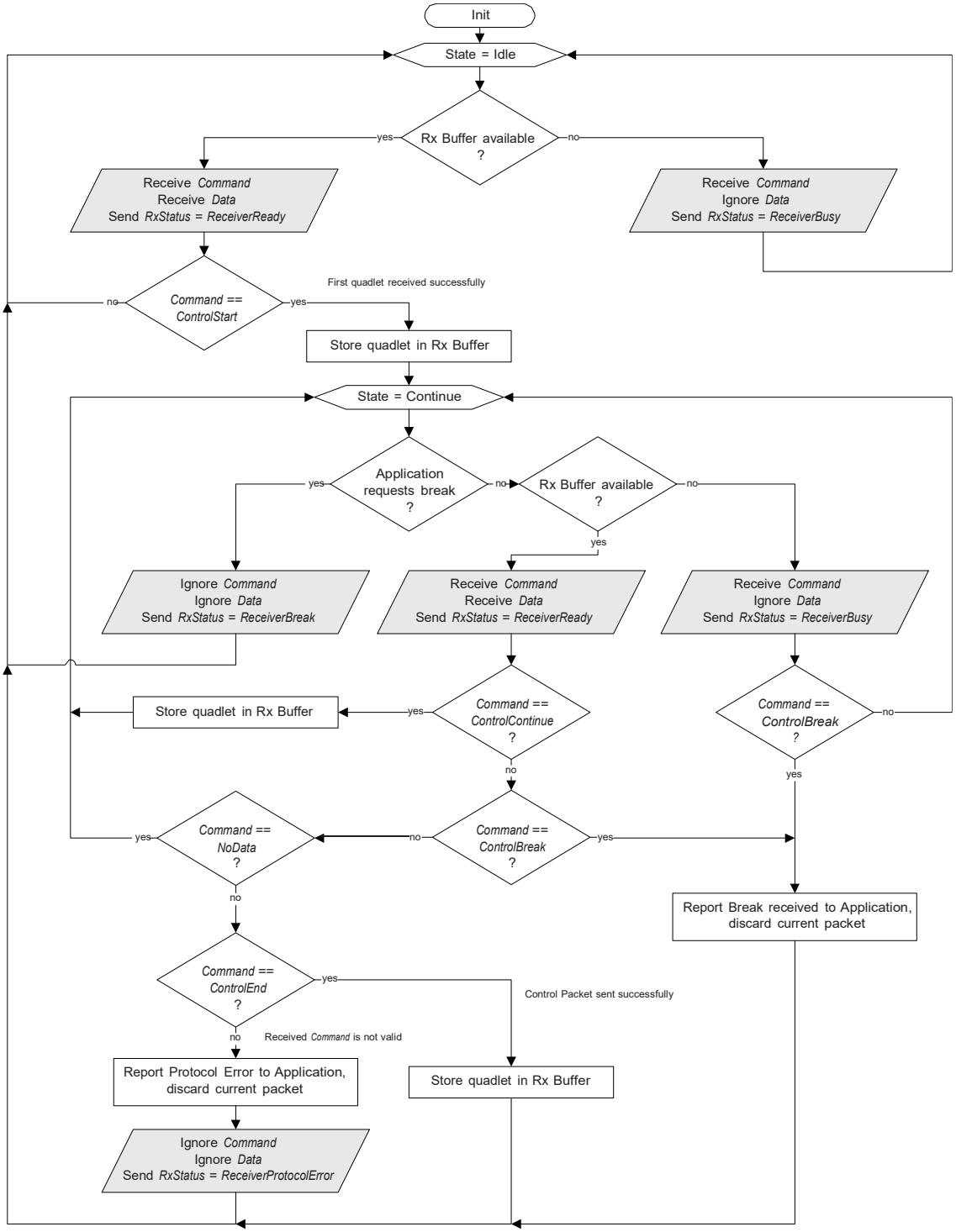
44.2 Embedded Characteristics

- Provides Serial Synchronous Communication Links Used in Audio and Telecom Applications
- Contains an Independent Receiver and Transmitter and a Common Clock Divider
- Interfaced with the DMA Controller (DMAC) to Reduce Processor Overhead
- Offers a Configurable Frame Sync and Data Length
- Receiver and Transmitter Can be Programmed to Start Automatically or on Detection of Different Events on the Frame Sync Signal
- Receiver and Transmitter Include a Data Signal, a Clock Signal and a Frame Sync Signal

Bit 4 – FILTER Receiver Digital Filter
0 (DISABLED): UART does not filter the receive line.

1 (ENABLED): UART filters the receive line using a three-sample filter (16x-bit clock) (2 over 3 majority).

Figure 48-9. Control Packet Rx Device Protocol



Synchronous

Synchronous stream data is sent in a continuous and broadcast fashion, without block information. Therefore, receiving Devices must not respond to the synchronous command; thereby leaving RxStatus in the ReceiverReady state (logic low). For 3-pin MediaLB, the required pull-down on MLBS leaves this signal in the ReceiverReady command when no synchronous data is transmitted on the MLBD line.

Offset	Name	Bit Pos.								
0x84	MCAN_SIDFC	7:0	FLSSA[5:0]							
		15:8	FLSSA[13:6]							
		23:16	LSS[7:0]							
		31:24								
0x88	MCAN_XIDFC	7:0	FLESA[5:0]							
		15:8	FLESA[13:6]							
		23:16		LSE[6:0]						
		31:24								
0x8C ... 0x8F	Reserved									
0x90	MCAN_XIDAM	7:0	EIDM[7:0]							
		15:8	EIDM[15:8]							
		23:16	EIDM[23:16]							
		31:24				EIDM[28:24]				
0x94	MCAN_HPMS	7:0	MSI[1:0]		BIDX[5:0]					
		15:8	FLST	FIDX[6:0]						
		23:16								
		31:24								
0x98	MCAN_NDAT1	7:0	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
		15:8	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
		23:16	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
		31:24	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
0x9C	MCAN_NDAT2	7:0	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
		15:8	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
		23:16	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
		31:24	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
0xA0	MCAN_RXF0C	7:0	F0SA[5:0]							
		15:8	F0SA[13:6]							
		23:16		F0S[6:0]						
		31:24	F0OM	F0WM[6:0]						
0xA4	MCAN_RXF0S	7:0		F0FL[6:0]						
		15:8			F0GI[5:0]					
		23:16			F0PI[5:0]					
		31:24							RF0L	F0F
0xA8	MCAN_RXF0A	7:0			F0AI[5:0]					
		15:8								
		23:16								
		31:24								
0xAC	MCAN_RXBC	7:0	RBSA[5:0]							
		15:8	RBSA[13:6]							
		23:16								
		31:24								
0xB0	MCAN_RXF1C	7:0	F1SA[5:0]							
		15:8	F1SA[13:6]							
		23:16		F1S[6:0]						

54.7 Register Summary

Offset	Name	Bit Pos.								
0x00	ACC_CR	7:0								SWRST
		15:8								
		23:16								
		31:24								
0x04	ACC_MR	7:0		SELPLUS[2:0]				SELMINUS[2:0]		
		15:8		FE	SELFS	INV		EDGETYP[1:0]		ACEN
		23:16								
		31:24								
0x08 ... 0x23	Reserved									
0x24	ACC_IER	7:0								CE
		15:8								
		23:16								
		31:24								
0x28	ACC_IDR	7:0								CE
		15:8								
		23:16								
		31:24								
0x2C	ACC_IMR	7:0								CE
		15:8								
		23:16								
		31:24								
0x30	ACC_ISR	7:0							SCO	CE
		15:8								
		23:16								
		31:24	MASK							
0x34 ... 0x93	Reserved									
0x94	ACC_ACR	7:0						HYST[1:0]		ISEL
		15:8								
		23:16								
		31:24								
0x98 ... 0xE3	Reserved									
0xE4	ACC_WPMR	7:0								WPEN
		15:8	WPKEY[7:0]							
		23:16	WPKEY[15:8]							
		31:24	WPKEY[23:16]							
0xE8	ACC_WPSR	7:0								WPVS
		15:8								

55.5.2.2 ICM Region Configuration Structure Member

Name: ICM_RCFG
Property: Read/Write

Register offset is calculated as ICM_DSCR+0x004+RID*(0x10).

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
		ALGO[2:0]				PROCDLY	SUIEN	ECIEN
Access		R/W	R/W	R/W		R/W	R/W	R/W
Reset								

Bit	7	6	5	4	3	2	1	0
	WCIEN	BEIEN	DMIEN	RHIEN		EOM	WRAP	CDWBN
Access	R/W	R/W	R/W	R/W		R/W	R/W	R/W
Reset								

Bits 14:12 – ALGO[2:0] SHA Algorithm
Values which are not listed in the table must be considered as “reserved”.

Value	Name	Description
0	SHA1	SHA1 algorithm processed
1	SHA256	SHA256 algorithm processed

Bit 10 – PROCDLY Processing Delay
When SHA1 algorithm is processed, the runtime period is either 85 or 209 clock cycles.
When SHA256 algorithm is processed, the runtime period is either 72 or 194 clock cycles.

Value	Name	Description
0	SHORTEST	SHA processing runtime is the shortest one.
1	LONGEST	SHA processing runtime is the longest one.

Bit 9 – SUIEN Monitoring Status Updated Condition Interrupt (Default Enabled)

Value	Name	Description
0		The ICM_ISR.RSU[i] flag is set when the corresponding descriptor is loaded from memory to ICM.
1		The ICM_ISR.RSU[i] flag remains cleared even if the setting condition is met.

Bit 8 – ECIEN End Bit Condition Interrupt (Default Enabled)

Date	Comments
01-June-16	Section 31. “Clock Generator” Oscillator naming changed throughout to: - Slow RC oscillator - 32.768 kHz crystal oscillator - Main RC oscillator - Main crystal oscillator Main RC oscillator default frequency changed to 12 MHz throughout. Updated Figure 31-1 “Clock Generator Block Diagram”. Section 31.4 “Slow Clock”: changed ‘powered up’ to ‘powered’. Updated Section 31.4.1 “Slow RC Oscillator (32 kHz typical)”. Section 31.4.2 “32.768 kHz Crystal Oscillator”: identified default state. Updated details on XIN32 and XOUT32. Deleted sentence on external capacitors and figure “Typical 32.768 kHz Crystal Oscillator Connection”. Updated paragraph on selecting the source of Slow clock. Updated Figure 31-2 “Main Clock (MAINCK) Block Diagram”. Added Figure 31-3 “Main Frequency Counter Block Diagram”. Section 31.5.1 “Main RC Oscillator”: added Note in paragraph on output frequency. Corrected two occurrences of ‘Main clock’ to ‘Main RC oscillator’. Deleted recommendation to disable oscillators under certain conditions. Moved paragraph on adjusting Main RC osc frequency to Section 31.5.2 “Main RC Oscillator Frequency Adjustment”. Section 31.5.2 “Main RC Oscillator Frequency Adjustment”: updated 1st and last paragraphs. Deleted some redundant content. Section 31.5.3 “Main Crystal Oscillator”: updated information on programming startup time. Section 31.5.4 “Main Clock Source Selection”: updated list of selectable main clock sources. Section 31.5.6 “Main Frequency Counter” renamed section (was “Main Clock Frequency Counter”). Updated 1st paragraph. Section 31.5.7 “Switching Main Clock between the RC Oscillator and Crystal Oscillator” now part of Section 31.5.6 “Main Frequency Counter”. Section 31.6 now titled ”PLLA Clock” (was “Divider and PLL Block”). Section 31.6.1 “Divider and Phase Lock Loop Programming”: updated information on changing MAINCK characteristics. Section 31.7 now titled ”UTMI PLL Clock” (was UTMI Phase Lock Loop Programming). Added paragraph on multiplying factors.
cont’d	
01-June-16	Section 32. “Power Management Controller (PMC)” Main RC oscillator default frequency changed to 12 MHz throughout.