Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM® Cortex®-M7
Core Size	32-Bit Single-Core
Speed	300MHz
Connectivity	I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	75
Program Memory Size	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	1.08V ~ 3.6V
Data Converters	A/D 10x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TFBGA
Supplier Device Package	100-TFBGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsams70n21a-cn

The WPVS bit is automatically cleared after reading the PMC_WPSR.

The following registers are write-protected when the WPEN bit is set in PMC_WPMR:

- PMC System Clock Disable Register
- PMC Peripheral Clock Enable Register 0
- PMC Peripheral Clock Disable Register 0
- PMC Clock Generator Main Oscillator Register
- PMC Clock Generator Main Clock Frequency Register
- PMC Clock Generator PLLA Register
- PMC UTMI Clock Configuration Register
- PMC Master Clock Register
- PMC USB Clock Register
- PMC Programmable Clock Register
- PMC Fast Startup Mode Register
- PMC Fast Startup Polarity Register
- PMC Peripheral Clock Enable Register 1
- PMC Peripheral Clock Disable Register 1
- PMC Oscillator Calibration Register
- PMC SleepWalking Enable Register 0
- PMC SleepWalking Disable Register 0
- PLL Maximum Multiplier Value Register
- PMC SleepWalking Enable Register 1
- PMC SleepWalking Disable Register 1

31.20 Register Summary

Offset	Register	Name	Access	Reset
0x0000	System Clock Enable Register	PMC_SCER	Write-only	–
0x0004	System Clock Disable Register	PMC_SCDR	Write-only	–
0x0008	System Clock Status Register	PMC_SCSR	Read-only	0x0000_0001
0x000C	Reserved	–	–	–
0x0010	Peripheral Clock Enable Register 0	PMC_PCER0	Write-only	–
0x0014	Peripheral Clock Disable Register 0	PMC_PCDR0	Write-only	–
0x0018	Peripheral Clock Status Register 0	PMC_PCSR0	Read-only	0x0000_0000
0x001C	UTMI Clock Register	CKGR_UCKR	Read/Write	0x1020_0800
0x0020	Main Oscillator Register	CKGR_MOR	Read/Write	0x0000_0008
0x0024	Main Clock Frequency Register	CKGR_MCFR	Read/Write	0x0000_0000

31.20.14 PMC Interrupt Enable Register

Name: PMC_IER
Offset: 0x0060
Property: Write-only

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

0: No effect.

1: Enables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			XT32KERR			CFDEV	MOSCRCS	MOSCSELS
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
		PCKRDY6	PCKRDY5	PCKRDY4	PCKRDY3	PCKRDY2	PCKRDY1	PCKRDY0
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		LOCKU			MCKRDY		LOCKA	MOSCXTS
Access								
Reset								

Bit 21 – XT32KERR 32.768 kHz Crystal Oscillator Error Interrupt Enable

Bit 18 – CFDEV Clock Failure Detector Event Interrupt Enable

Bit 17 – MOSCRCS Main RC Oscillator Status Interrupt Enable

Bit 16 – MOSCSELS Main Clock Source Oscillator Selection Status Interrupt Enable

Bits 8, 9, 10, 11, 12, 13, 14 – PCKRDY Programmable Clock Ready x Interrupt Enable

Bit 6 – LOCKU UTMI PLL Lock Interrupt Enable

Bit 3 – MCKRDY Master Clock Ready Interrupt Enable

Bit 1 – LOCKA PLLA Lock Interrupt Enable

Bit 0 – MOSCXTS Main Crystal Oscillator Status Interrupt Enable

32.6.1.19 PIO Multi-driver Disable Register

Name: PIO_MDDR
Offset: 0x0054
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								

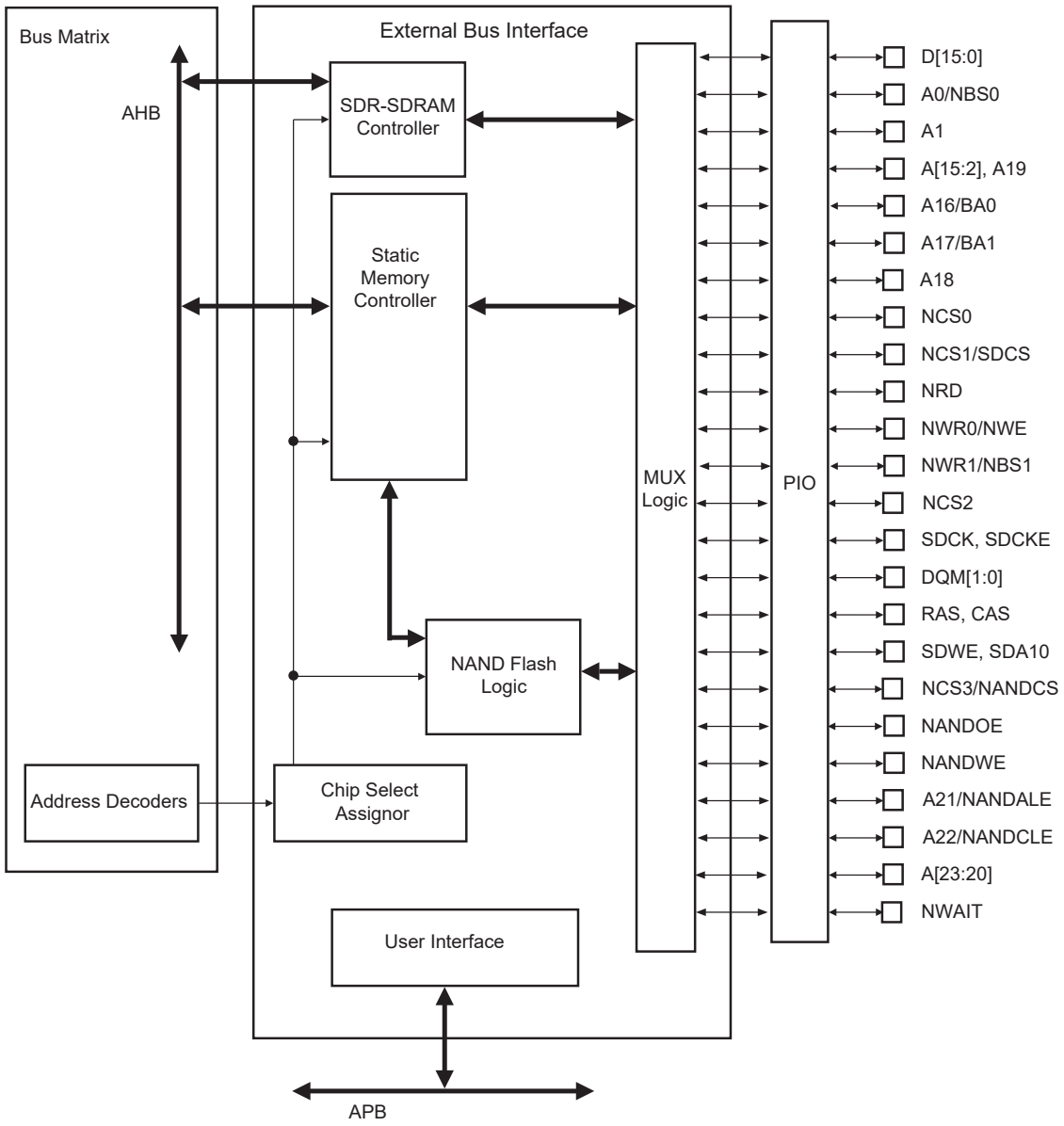
Bit	7	6	5	4	3	2	1	0
	P7	P6	P5	P4	P3	P2	P1	P0
Access								
Reset								

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 Multi-drive Disable

Value	Description
0	No effect.
1	Disables multi-drive on the I/O line.

33.3 EBI Block Diagram

Figure 33-1. Organization of the External Bus Interface



33.4 I/O Lines Description

Table 33-1. EBI I/O Lines Description

Name	Function	Type	Active Level
EBI			
D0–D15	Data Bus	I/O	
A0–A23	Address Bus	Output	

Offset	Name	Bit Pos.								
0x050F										
0x0510	XDMAC_CIE19	7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
		15:8								
		23:16								
		31:24								
0x0514	XDMAC_CID19	7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
		15:8								
		23:16								
		31:24								
0x0518	XDMAC_CIM19	7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
		15:8								
		23:16								
		31:24								
0x051C	XDMAC_CIS19	7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
		15:8								
		23:16								
		31:24								
0x0520	XDMAC_CSA19	7:0	SA[7:0]							
		15:8	SA[15:8]							
		23:16	SA[23:16]							
		31:24	SA[31:24]							
0x0524	XDMAC_CDA19	7:0	DA[7:0]							
		15:8	DA[15:8]							
		23:16	DA[23:16]							
		31:24	DA[31:24]							
0x0528	XDMAC_CNDA19	7:0	NDA[5:0]							NDAIF
		15:8	NDA[13:6]							
		23:16	NDA[21:14]							
		31:24	NDA[29:22]							
0x052C	XDMAC_CNDC19	7:0				NDVIEW[1:0]		NDDUP	NDSUP	NDE
		15:8								
		23:16								
		31:24								
0x0530	XDMAC_CUBC19	7:0	UBLEN[7:0]							
		15:8	UBLEN[15:8]							
		23:16	UBLEN[23:16]							
		31:24								
0x0534	XDMAC_CBC19	7:0	BLEN[7:0]							
		15:8					BLEN[11:8]			
		23:16								
		31:24								
0x0538	XDMAC_CC19	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]						

37.6.25 ISI Write Protection Status Register

Name: ISI_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
	WPVSRC[15:8]							
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
	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 23:8 – WPVSRC[15:0] Write Protection Violation Source

Value	Name
0	No Write Protection Violation occurred since the last read of this register (ISI_WPSR).
1	Write access in ISI_CFG1 while Write Protection was enabled (since the last read).
2	Write access in ISI_CFG2 while Write Protection was enabled (since the last read).
3	Write access in ISI_PSIZE while Write Protection was enabled (since the last read).
4	Write access in ISI_PDECF while Write Protection was enabled (since the last read).
5	Write access in ISI_Y2R_SET0 while Write Protection was enabled (since the last read).
6	Write access in ISI_Y2R_SET1 while Write Protection was enabled (since the last read).
7	Write access in ISI_R2Y_SET0 while Write Protection was enabled (since the last read).
8	Write access in ISI_R2Y_SET1 while Write Protection was enabled (since the last read).
9	Write access in ISI_R2Y_SET2 while Write Protection was enabled (since the last read).

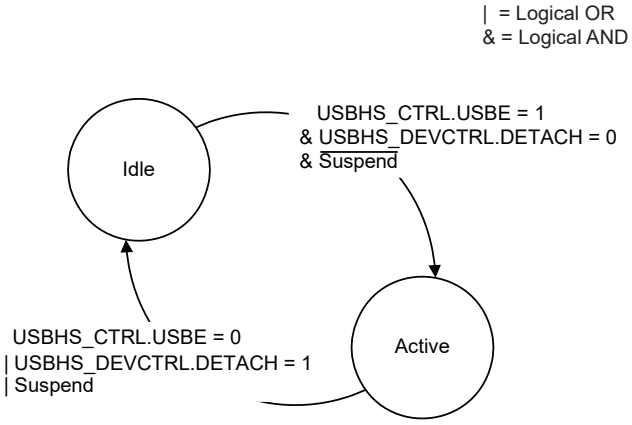
Bit 0 – WPVS Write Protection Violation Status

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

39.5.1.6 Pad Suspend

Figure 39-5 shows the pad behavior.

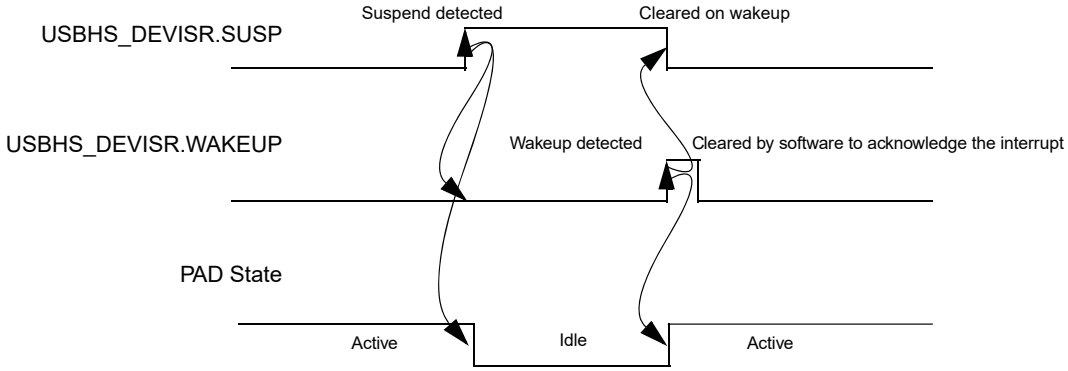
Figure 39-5. Pad Behavior



- In Idle state, the pad is put in Low-power mode, i.e., the differential receiver of the USB pad is off, and internal pull-downs with a strong value (15 K) are set in HSDP/D and HSDM/DM to avoid floating lines.
- In Active state, the pad is working.

Figure 39-6 illustrates the pad events leading to a PAD state change.

Figure 39-6. Pad Events



The USBHS_DEVISR.SUSP bit is set and the Wakeup Interrupt (USBHS_DEVISR.WAKEUP) bit is cleared when a USB “Suspend” state has been detected on the USB bus. This event automatically puts the USB pad in Idle state. The detection of a non-idle event sets USBHS_DEVISR.WAKEUP, clears USBHS_DEVISR.SUSP and wakes up the USB pad.

The pad goes to the Idle state if the USBHS is disabled or if the USBHS_DEVCTRL.DETACH bit = 1. It returns to the Active state when USBHS_CTRL.USBE = 1 and USBHS_DEVCTRL.DETACH = 0.

39.5.2 USB Device Operation

39.5.2.1 Introduction

In Device mode, the USBHS supports high-, full- and low-speed data transfers.

In addition to the default control endpoint, 10 endpoints are provided, which can be configured with an isochronous, bulk or interrupt type, as described in Table 39-1.

As the Device mode starts in Idle state, the pad consumption is reduced to the minimum.

Value	Description
	• (INRQ+1) In requests have been processed. • A Pipe Reset (USBHS_HSTPIP.PRSTx rising) has occurred. • A Pipe Enable (USBHS_HSTPIP.PEN rising) has occurred.

Bit 16 – PDISHDMA Pipe Interrupts Disable HDMA Request Enable
See the USBHS_DEVEPTIMR.EPDISHDMA bit description.

Bit 14 – FIFOCON FIFO Control
For OUT and SETUP pipes:

0: Cleared when USBHS_HSTPIPIDR.FIFOCONC = 1. This sends the FIFO data and switches the bank.

1: Set when the current bank is free, at the same time as USBHS_HSTPIPISR.TXOUTI or TXSTPI.

For IN pipes:

0: Cleared when USBHS_HSTPIPIDR.FIFOCONC = 1. This frees the current bank and switches to the next bank.

1: Set when a new IN message is stored in the current bank, at the same time as USBHS_HSTPIPISR.RXINI.

Bit 12 – NBUSYBKE Number of Busy Banks Interrupt Enable

Value	Description
0	Cleared when USBHS_HSTPIPIDR.NBUSYBKEC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.NBUSYBKE).
1	Set when USBHS_HSTPIPIER.NBUSYBKES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.NBUSYBKE).

Bit 7 – SHORTPACKETIE Short Packet Interrupt Enable
If this bit is set for non-control OUT pipes, a short packet transmission is guaranteed upon ending a DMA transfer, thus signaling an end of transfer, provided that the End of DMA Buffer Output Enable (USBHS_HSTDMACONTROL.END_B_EN) bit and the Automatic Switch (USBHS_HSTPIPCFG.AUTOSW) bit = 1.

Value	Description
0	Cleared when USBHS_HSTPIPIDR.SHORTPACKETEC = 1. This disables the Transmitted interrupt Data IT (USBHS_HSTPIPIMR.SHORTPACKETE).
1	Set when USBHS_HSTPIPIER.SHORTPACKETIES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.SHORTPACKETIE).

Bit 6 – CRCERRE CRC Error Interrupt Enable

Value	Description
0	Cleared when USBHS_HSTPIPIDR.CRCERREC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.CRCERRE).
1	Set when USBHS_HSTPIPIER.CRCERRES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.CRCERRE).

Bit 5 – OVERFIE Overflow Interrupt Enable

40.14.19 HSMCI Write Protection Status Register

Name: HSMCI_WPSR
Offset: 0xE8
Reset: 0x0
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 HSMCI_WPSR.
1	A write protection violation has occurred since the last read of the HSMCI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

41.8.8 SPI Interrupt Mask Register

Name: SPI_IMR
Offset: 0x1C
Reset: 0x0
Property: Read-only

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								

Bit	15	14	13	12	11	10	9	8
						UNDES	TXEMPTY	NSSR
Access						R	R	R
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
					OVRES	MODF	TDRE	RDRF
Access					R	R	R	R
Reset					0	0	0	0

Bit 10 – UNDES Underrun Error Interrupt Mask

Bit 9 – TXEMPTY Transmission Registers Empty Mask

Bit 8 – NSSR NSS Rising Interrupt Mask

Bit 3 – OVRES Overrun Error Interrupt Mask

Bit 2 – MODF Mode Fault Error Interrupt Mask

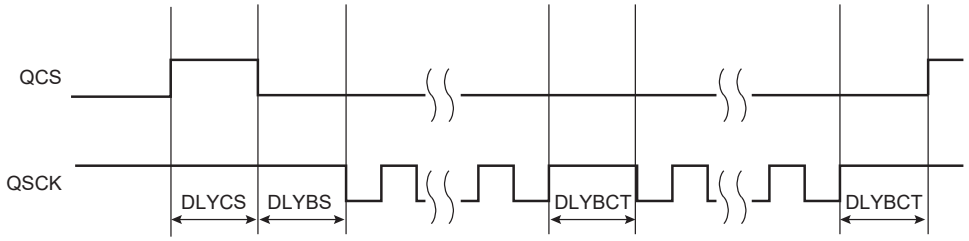
Bit 1 – TDRE SPI Transmit Data Register Empty Interrupt Mask

Bit 0 – RDRF Receive Data Register Full Interrupt Mask

- The delay between consecutive transfers, programmed by writing QSPI_MR.DLYBCT. Allows insertion of a delay between two consecutive transfers. In Serial Memory mode, this delay is not programmable and DLYBCT is ignored. In this mode, DLYBCT must be written to ‘0’.

These delays allow the QSPI to be adapted to the interfaced peripherals and their speed and bus release time.

Figure 42-4. Programmable Delays



42.6.4 QSPI SPI Mode

In SPI mode, the QSPI acts as a standard SPI Master.

To activate this mode, QSPI_MR.SMM must be written to ‘0’ in QSPI_MR.

42.6.4.1 SPI Mode Operations

The QSPI in standard SPI mode operates on the clock generated by the internal programmable baud rate generator. It fully controls the data transfers to and from the slave connected to the SPI bus. The QSPI drives the chip select line to the slave (QCS) and the serial clock signal (QSCK).

The QSPI features two holding registers, the Transmit Data register (QSPI_TDR) and the Receive Data register (QSPI_RDR), and a single internal shift register. The holding registers maintain the data flow at a constant rate.

After enabling the QSPI, a data transfer begins when the processor writes to the QSPI_TDR. The written data is immediately transferred to the internal shift register and transfer on the SPI bus starts. While the data in the internal shift register is shifted on the MOSI line, the MISO line is sampled and shifted to the internal shift register. Receiving data cannot occur without transmitting data. If receiving mode is not needed, for example when communicating with a slave receiver only (such as an LCD), the receive status flags in the Status register (QSPI_SR) can be discarded.

If new data is written in QSPI_TDR during the transfer, it is retained there until the current transfer is completed. Then, the received data is transferred from the internal shift register to the QSPI_RDR, the data in QSPI_TDR is loaded in the internal shift register and a new transfer starts.

The transfer of a data written in QSPI_TDR in the internal shift register is indicated by the Transmit Data Register Empty (TDRE) bit in the QSPI_SR. When new data is written in QSPI_TDR, this bit is cleared. QSPI_SR.TDRE is used to trigger the Transmit DMA channel.

The end of transfer is indicated by the TXEMPTY flag in the QSPI_SR. If a transfer delay (DLYBCT) is greater than 0 for the last transfer, QSPI_SR.TXEMPTY is set after the completion of this delay. The peripheral clock can be switched off at this time.

The transfer of received data from the internal shift register in QSPI_RDR is indicated by the Receive Data Register Full (RDRF) bit in the QSPI_SR. When the received data is read, QSPI_SR.RDRF bit is cleared.

45.8.8 I2SC Interrupt Mask Register

Name: I2SC_IMR
Offset: 0x1C
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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
		TXUR	TXRDY			RXOR	RXRDY	
Access		R	R			R	R	
Reset		0	0			0	0	

Bit 6 – TXUR Transmit Underflow Interrupt Disable

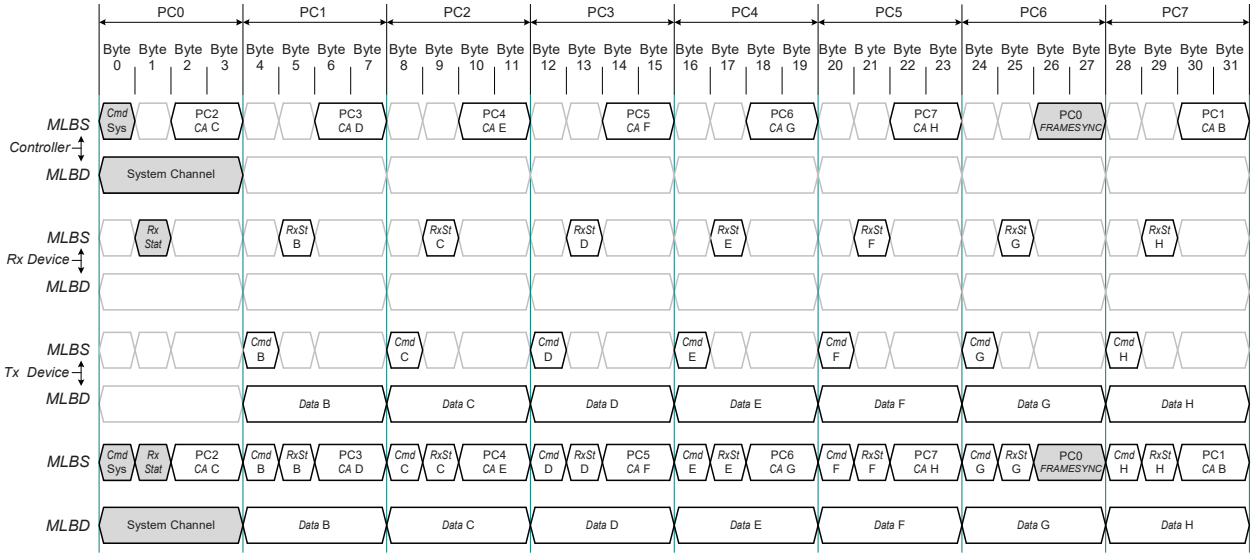
Value	Description
0	The corresponding interrupt is disabled. This bit is cleared when the corresponding bit in I2SC_IDR is written to ’1’.
1	The corresponding interrupt is enabled. This bit is set when the corresponding bit in I2SC_IER is written to ’1’.

Bit 5 – TXRDY Transmit Ready Interrupt Disable

Value	Description
0	The corresponding interrupt is disabled. This bit is cleared when the corresponding bit in I2SC_IDR is written to ’1’.
1	The corresponding interrupt is enabled. This bit is set when the corresponding bit in I2SC_IER is written to ’1’.

Bit 2 – RXOR Receiver Overrun Interrupt Disable

Figure 48-5. 3-pin MediaLB 256Fs Interface Example



48.6.1.7 Initialization

At power up, the MediaLB Controller might output a MLBReset command in the System Channel (all System commands are optional). Upon reception of the MLBReset command, all MediaLB Devices will cancel any current transmissions or receptions and clear their buffers.

Two scenarios are supported to configure MediaLB Devices and ChannelAddresses:

- Static pre-configured before startup. The system implementor decides which ChannelAddresses are to be used for every communication path on MediaLB. This static MediaLB configuration can be communicated by the EHC to the Controller through pre-defined power-up logical channels or through a secondary port.
- Dynamically at run-time. Dynamic configuration allows the board designer to support multiple build options where the EHC can query to find out if a particular Device is present or not on a particular board. The EHC instructs the Controller to scan for a particular DeviceAddress in the System Channel. The Controller uses the MLBScan command to look for a Device. The Controller then notifies the EHC whether the Device is present or not. If the Device is present, then the EHC can instruct the Controller to set the ChannelAddresses for the Device found. The EHC sends messages to the Controller to set each Indices/Logical channel, and waits the appropriate amount of time between each message as specified in the Devices documentation. When that particular Device is configured, the EHC can instruct the Controller to scan for the next Device.

Since the MediaLB Controller is the interface between the MediaLB Devices and the MOST Network, the Controller provides the MLBC signal and will also continue to operate even when the MOST Network is unlocked. When no activity exists on MediaLB, the Controller can shut off the MLBC placing MediaLB in a low-power state. The ChannelAddress assignments are not affected in low-power state; therefore, the same communication paths exists once MLBC is restarted.

MediaLB Devices are synchronously slaved to the MediaLB Controller through the MLBC signal. Since the Controller is synchronized to the MOST Network, the MLBC signal provides Network synchronization to all MediaLB Devices. Once the Controller starts up MLBC, all MediaLB Devices must synchronize to the MediaLB frame before communication can commence. When not frame-locked, Devices must search for the FRAMESYNC pattern, which defines a byte and physical channel boundary. Additionally, the start of the MediaLB frame (PC0) occurs one quadlet after FRAMESYNC is present on the bus. Even when a

49.5.4.1 Acceptance Filtering

The MCAN offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to an Rx Buffer or to Rx FIFO 0,1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are:

- Each filter element can be configured as
  - range filter (from - to)
  - filter for one or two dedicated IDs
  - classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled / disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are:

- Global Filter Configuration (MCAN_GFC)
- Standard ID Filter Configuration (MCAN_SIDFC)
- Extended ID Filter Configuration (MCAN_XIDFC)
- Extended ID and Mask (MCAN_XIDAM)

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag (MCAN_IR.HPM)
- Set High Priority Message interrupt flag (MCAN_IR.HPM) and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the effected Rx Buffer or Rx FIFO:

- Rx Buffer
  New Data flag of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC.
- Rx FIFO
  Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type, see MCAN_PSR.LEC and MCAN_PSR.DLEC. In case the matching Rx FIFO is operated in Overwrite mode, the boundary conditions described in Rx FIFO Overwrite Mode have to be considered.

49.6.8 MCAN Nominal Bit Timing and Prescaler Register

Name: MCAN_NBTP
Offset: 0x1C
Reset: 0x06000A03
Property: Read/Write

This register can only be written if the bits CCE and INIT are set in MCAN_CCCR.

The CAN bit time may be programmed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 CAN core clock periods. $t_q = t_{core\ clock}$ x (NBRP + 1).

NTSEG1 is the sum of Prop_Seg and Phase_Seg1. NTSEG2 is Phase_Seg2.

Therefore the length of the bit time is (programmed values) [NTSEG1 + NTSEG2 + 3] t_q
or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

With a CAN core clock frequency of 8 MHz, the reset value of 0x06000A03 configures the MCAN for a bit rate of 500 kbit/s.

Bit	31	30	29	28	27	26	25	24
	NSJW[6:0]							NBRP[8:8]
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	0

Bit	23	22	21	20	19	18	17	16
	NBRP[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	15	14	13	12	11	10	9	8
	NTSEG1[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	1	0

Bit	7	6	5	4	3	2	1	0
		NTSEG2[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	1	1

Bits 31:25 – NSJW[6:0] Nominal (Re) Synchronization Jump Width
0 to 127: The duration of a synchronization jump is t_q x (NSJW + 1).

Bits 24:16 – NBRP[8:0] Nominal Bit Rate Prescaler
0 to 511: The value by which the oscillator frequency is divided for generating the CAN time quanta. The CAN time is built up from a multiple of this quanta. CAN time quantum (tq) = $t_{core\ clock}$ x (NBRP + 1)

Bits 15:8 – NTSEG1[7:0] Nominal Time Segment Before Sample Point

49.6.26 MCAN New Data 2

Name: MCAN_NDAT2
Offset: 0x9C
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56
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
	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
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
	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40
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
	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
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, 24, 25, 26, 27, 28, 29, 30, 31 – NDx New Data
The register holds the New Data flags of Receive Buffers 32 to 63. The flags are set when the respective Receive Buffer has been updated from a received frame. The flags remain set until the processor clears them. A flag is cleared by writing a ‘1’ to the corresponding bit position. Writing a ‘0’ has no effect. A hard reset will clear the register.

Value	Description
0	Receive Buffer not updated.
1	Receive Buffer updated from new message.

50.7.17 TC QDEC Interrupt Enable Register

Name: TC_QIER
Offset: 0xC8
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
					MPE	QERR	DIRCHG	IDX
Access					W	W	W	W
Reset					–	–	–	–

Bit 3 – MPE Consecutive Missing Pulse Error

Value	Description
0	No effect.
1	Enables the interrupt when an occurrence of MAXCMP consecutive missing pulses is detected.

Bit 2 – QERR Quadrature Error

Value	Description
0	No effect.
1	Enables the interrupt when a quadrature error occurs on PHA, PHB.

Bit 1 – DIRCHG Direction Change

Value	Description
0	No effect.
1	Enables the interrupt when a change on rotation direction is detected.

Bit 0 – IDX Index

  - Independent Clock Selection for Each Channel
  - Independent Period, Duty-Cycle and Dead-Time for Each Channel
  - Independent Double Buffering of Period, Duty-Cycle and Dead-Times for Each Channel
  - Independent Programmable Selection of The Output Waveform Polarity for Each Channel, with Double Buffering
  - Independent Programmable Center- or Left-aligned Output Waveform for Each Channel
  - Independent Output Override for Each Channel
  - Independent Interrupt for Each Channel, at Each Period for Left-Aligned or Center-Aligned Configuration
  - Independent Update Time Selection of Double Buffering Registers (Polarity, Duty Cycle) for Each Channel, at Each Period for Left-Aligned or Center-Aligned Configuration
- External Trigger Input Management (e.g., for DC/DC or Lighting Control)
  - External PWM Reset Mode
  - External PWM Start Mode
  - Cycle-By-Cycle Duty Cycle Mode
  - Leading-Edge Blanking
- 2 2-bit Gray Up/Down Channels for Stepper Motor Control
- Spread Spectrum Counter to Allow a Constantly Varying Duty Cycle (only for Channel 0)
- Synchronous Channel Mode
  - Synchronous Channels Share the Same Counter
  - Mode to Update the Synchronous Channels Registers after a Programmable Number of Periods
  - Synchronous Channels Supports Connection of one DMA Controller Channel Which Offers Buffer Transfer Without Processor Intervention To Update Duty-Cycle Registers
- 2 Independent Events Lines Intended to Synchronize ADC Conversions
  - Programmable delay for Events Lines to delay ADC measurements
- 8 Comparison Units Intended to Generate Interrupts, Pulses on Event LinesDMA Controller Transfer Requests
- 8 Programmable Fault Inputs Providing an Asynchronous Protection of PWM Outputs
  - 3 User Driven through PIO Inputs
  - PMC Driven when Crystal Oscillator Clock Fails
  - ADC Controller Driven through Configurable Comparison Function
  - Analog Comparator Controller Driven
  - Timer/Counter Driven through Configurable Comparison Function
- Register Write Protection

Bit 0 – WRDY Write Ready for Synchronous Channels Update

Value	Description
0	New duty-cycle and dead-time values for the synchronous channels cannot be written.
1	New duty-cycle and dead-time values for the synchronous channels can be written.

where:

- Z_{IN} is input impedance in Single-ended or Differential mode
- C_{IN} = 2 to 8 pF ±20% depending on the gain value and mode (SE or DIFF); temperature dependency is negligible
- R_{ON} is typical 2 kΩ and 8 kΩ max (worst case process and high temperature)

The following formula is used to calculate input impedance:

$$Z_{\mathrm{IN}} = \frac{1}{f_S \times C_{\mathrm{IN}}}$$

where:

- f_S is the sampling frequency of the AFE channel
- Typ values are used to compute AFE input impedance Z_{IN}

Table 59-37. Input Capacitance (C_{IN}) Values

Gain Selection	Single-ended	Differential	Unit
1	2	2	pF
2	4	4	
4	8	8	

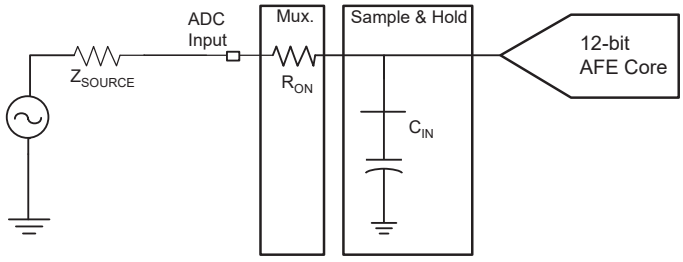
Table 59-38. Z_{IN} Input Impedance

f_S (MHz)	1	0.5	0.25	0.125	0.0625	0.03125	0.015625	0.007813
C_{IN} = 2 pF								
Z_{IN} (MΩ)	0.5	1	2	4	8	16	32	64
C_{IN} = 4 pF								
Z_{IN} (MΩ)	0.25	0.5	1	2	4	8	16	32
C_{IN} = 8 pF								
Z_{IN} (MΩ)	0.125	0.25	0.5	1	2	4	8	16

59.8.6.1 Track and Hold Time versus Source Output Impedance

The figure below shows a simplified acquisition path.

Figure 59-16. Simplified Acquisition Path



During the tracking phase, the AFE tracks the input signal during the tracking time shown below:

$t_{TRACK} = n \times C_{IN} \times (R_{ON}+Z_{SOURCE})/1000$

- Tracking time expressed in ns and Z_{SOURCE} expressed in Ω.