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, SSC, 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)	3V ~ 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-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamv71n21b-aabt

8. Input/Output Lines

The SAM E70/S70/V70/V71 features both general purpose I/Os (GPIO) and system I/Os. GPIOs can have alternate functionality due to multiplexing capabilities of the PIO controllers. The same PIO line can be used, whether in I/O mode or by the multiplexed peripherals. System I/Os include pins such as test pins, oscillators, erase or analog inputs.

8.1 General-Purpose I/O Lines

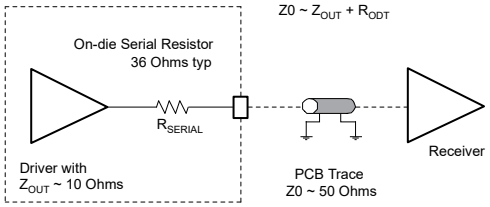
General-purpose (GPIO) lines are managed by PIO Controllers. All I/Os have several input or output modes such as pullup or pulldown, input Schmitt triggers, multi-drive (open-drain), glitch filters, debouncing or input change interrupt. Programming of these modes is performed independently for each I/O line through the PIO controller user interface. For more details, refer to 32. Parallel Input/Output Controller (PIO).

The input/output buffers of the PIO lines are supplied through VDDIO power supply rail.

The SAM E70/S70/V70/V71 embeds high-speed pads able to handle the high-speed clocks for HSMCI, SPI and QSPI (MCK/2). Refer to 58. Electrical Characteristics for SAM V70/V71 for more details. Typical pullup and pulldown value is 100 kΩ for all I/Os.

Each I/O line also embeds a R_{SERIAL} (On-die Serial Resistor), as shown in the following figure. It consists of an internal series resistor termination scheme for impedance matching between the driver output (SAM E70/S70/V70/V71) and the PCB trace impedance preventing signal reflection. The series resistor helps to reduce I/Os switching current (di/dt). thereby reducing in turn, EMI. It also decreases overshoot and undershoot (ringing) due to inductance of interconnect between devices or between boards. Finally, R_{SERIAL} helps diminish signal integrity issues.

Figure 8-1. On-Die Termination (ODT)



8.2 System I/O Lines

System I/O lines are pins used by oscillators, Test mode, reset, JTAG and other features. The following table lists the SAM E70/S70/V70/V71 system I/O lines shared with PIO lines.

These pins are software-configurable as general-purpose I/Os or system pins. At startup, the default function of these pins is always used.

19. Bus Matrix (MATRIX)

19.1 Description

The Bus Matrix (MATRIX) implements a multi-layer AHB, based on the AHB-Lite protocol, that enables parallel access paths between multiple AHB masters and slaves in a system, thus increasing the overall bandwidth. The MATRIX interconnects 13 AHB masters to 9 AHB slaves. The normal latency to connect a master to a slave is one cycle. The exception is the default master of the accessed slave which is connected directly (zero cycle latency).

The MATRIX user interface is compliant with ARM Advanced Peripheral Bus.

19.2 Embedded Characteristics

- 13 Masters
- 9 Slaves
- One Decoder for Each Master
- Several Possible Boot Memories for Each Master before Remap
- One Remap Function for Each Master
- Support for Long Bursts of 32, 64, 128 and up to the 256-beat Word Burst AHB Limit
- Enhanced Programmable Mixed Arbitration for Each Slave
  - Round-Robin
  - Fixed Priority
- Programmable Default Master for Each Slave
  - No Default Master
  - Last Accessed Default Master
  - Fixed Default Master
- Deterministic Maximum Access Latency for Masters
- Zero or One Cycle Arbitration Latency for the First Access of a Burst
- Bus Lock Forwarding to Slaves
- Master Number Forwarding to Slaves
- Configurable Automatic Clock-off Mode for Power Reduction
- One Special Function Register for Each Slave (not dedicated)
- Register Write Protection

19.2.1 Matrix Masters

The MATRIX manages the masters listed in he following table. Each master can perform an access to an available slave concurrently with other masters. lists the available masters.

Each master has its own specifically-defined decoder. To simplify addressing, all the masters have the same decodings.

19.4.10 SMC NAND Flash Chip Select Configuration Register

Name: CCFG_SMCNFCS
Offset: 0x0124
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
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				SDRAMEN	SMC_NFCS3	SMC_NFCS2	SMC_NFCS1	SMC_NFCS0
Access								
Reset				0	0	0	0	0

Bit 4 – SDRAMEN SDRAM Enable

WARNING This bit must not be used if SMC_NFCS1 is set.

WARNING: This must not be used if SMC_NFCS1 is set.

Value	Description
0	NCS1 is not assigned to SDRAM.
1	NCS1 is assigned to SDRAM.

Bit 3 – SMC_NFCS3 SMC NAND Flash Chip Select 3 Assignment

Value	Description
0	NCS3 is not assigned to a NAND Flash (NANDOE and NANWE not used for NCS3).
1	NCS3 is assigned to a NAND Flash (NANDOE and NANWE used for NCS3).

Bit 2 – SMC_NFCS2 SMC NAND Flash Chip Select 2 Assignment

Value	Description
0	NCS2 is not assigned to a NAND Flash (NANDOE and NANWE not used for NCS2).
1	NCS2 is assigned to a NAND Flash (NANDOE and NANWE used for NCS2).

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

Bit 2 – SEC Second Event

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

Bit 1 – ALARM Alarm Flag

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

Bit 0 – ACKUPD Acknowledge for Update

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

from the queue will be from immediately after the last successfully transmitted frame. As long as transmit is disabled by writing a '0' to the Transmit Enable bit in the Network Control register (GMAC_NCR.TXEN), the transmit buffer queue pointer resets to point to the address indicated by the Transmit Buffer Queue Base Address register (GMAC_TBQB).
Note: Disabling receive does not have the same effect on the receive buffer queue pointer.

Once the transmit queue is initialized, transmit is activated by writing a '1' to the Start Transmission bit of the Network Control register (GMAC_NCR.TSTART). Transmit is halted when a buffer descriptor with its used bit set is read, a transmit error occurs, or by writing to the Transmit Halt bit of the Network Control register (GMAC_NCR.THALT). Transmission is suspended if a pause frame is received while the Transmit Pause Frame bit is '1' in the Network Configuration register (GMAC_NCR.TXPF). Rewriting the Start bit (GMAC_NCR.TSTART) while transmission is active is allowed. This is implemented by the Transmit Go variable which is readable in the Transmit Status register (GMAC_TSR.TXGO). The TXGO variable is reset when:

- Transmit is disabled.
- A buffer descriptor with its ownership bit set is read.
- Bit 10, THALT, of the Network Control register is written.
- There is a transmit error such as too many retries or a transmit underrun.

To set TXGO, write a '1' to GMAC_NCR.TSTART. Transmit halt does not take effect until any ongoing transmit finishes.

If the DMA is configured for packet buffer Partial Store and Forward mode and a collision occurs during transmission of a multi-buffer frame, transmission will automatically restart from the first buffer of the frame. For packet buffer mode, the entire contents of the frame are read into the transmit packet buffer memory, so the retry attempt will be replayed directly from the packet buffer memory rather than having to re-fetch through the AHB.

If a used bit is read midway through transmission of a multi-buffer frame, this is treated as a transmit error. Transmission stops, GTXER is asserted and the FCS will be bad.

If transmission stops due to a transmit error or a used bit being read, transmission restarts from the first buffer descriptor of the frame being transmitted when the transmit start bit is rewritten.

38.6.3.5 DMA Bursting on the AHB

The DMA will always use SINGLE, or INCR type AHB accesses for buffer management operations. When performing data transfers, the AHB burst length is selected by the Fixed Burst Length for DMA Data Operations bit field in the DMA Configuration register (GMAC_DCFGR.FBLDO) so that either SINGLEor fixed length incrementing bursts (INCR4, INCR8 or INCR16) are used where possible:

When there is enough space and enough data to be transferred, the programmed fixed length bursts will be used. If there is not enough data or space available, for example when at the beginning or the end of a buffer, SINGLE type accesses are used. Also SINGLE type accesses are used at 1024 Byte boundaries, so that the 1 KByte boundaries are not burst over as per AHB requirements.

The DMA will not terminate a fixed length burst early, unless an error condition occurs on the AHB or if receive or transmit are disabled in the Network Control register (GMAC_NCR).

38.6.3.6 DMA Packet Buffer

The DMA uses packet buffers for both transmit and receive paths. This mode allows multiple packets to be buffered in both transmit and receive directions. This allows the DMA to withstand far greater access latencies on the AHB and make more efficient use of the AHB bandwidth. There are two modes of operation—Full Store and Forward and Partial Store and Forward.

38.8.48 GMAC 512 to 1023 Byte Frames Transmitted Register

Name: GMAC_TBFT1023
Offset: 0x128
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] 512 to 1023 Byte Frames Transmitted without Error
This register counts the number of 512 to 1023 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.

38.8.115 GMAC Screening Type 2 Compare Word 1 Register x

Name: GMAC_ST2CW1x
Offset: 0x0704 + x*0x08 [x=0..23]
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
Access								
Reset								0

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

Bits 8:7 – OFFSSTRT[1:0] Ethernet Frame Offset Start

Value	Name	Description
0	FRAMESTART	Offset from the start of the frame
1	ETHERTYPE	Offset from the byte after the EtherType field
2	IP	Offset from the byte after the IP header field
3	TCP_UDP	Offset from the byte after the TCP/UDP header field

Bits 6:0 – OFFSVAL[6:0] Offset Value in Bytes
The value of OFFSVAL ranges from 0 to 127 bytes, and is counted from either the start of the frame, the byte after the EtherType field (last EtherType in the header if the frame is VLAN tagged), the byte after the IP header (IPv4 or IPv6) or the byte after the TCP/UDP header.

- the receive and transmit bank FIFO counters,
- all registers of this endpoint (USBHS_DEVEPTCFGx, USBHS_DEVEPTISRx, the Endpoint x Control (USBHS_DEVEPTIMRx) register), except its configuration (USBHS_DEVEPTCFGx.ALLOC, USBHS_DEVEPTCFGx.EPBK, USBHS_DEVEPTCFGx.EPSIZE, USBHS_DEVEPTCFGx.EPDIR, USBHS_DEVEPTCFGx.EPTYPE) and the Data Toggle Sequence (USBHS_DEVEPTISRx.DTSEQ) field.
  Note: The interrupt sources located in USBHS_DEVEPTISRx are not cleared when a USB bus reset has been received.

The endpoint configuration remains active and the endpoint is still enabled.

The endpoint reset may be associated with a clear of the data toggle sequence as an answer to the CLEAR_FEATURE USB request. This can be achieved by writing a one to the Reset Data Toggle Set bit (RSTDTS) in the Device Endpoint x Control Set register (this sets the Reset Data Toggle bit USBHS_DEVEPTIMRx.RSTDT).

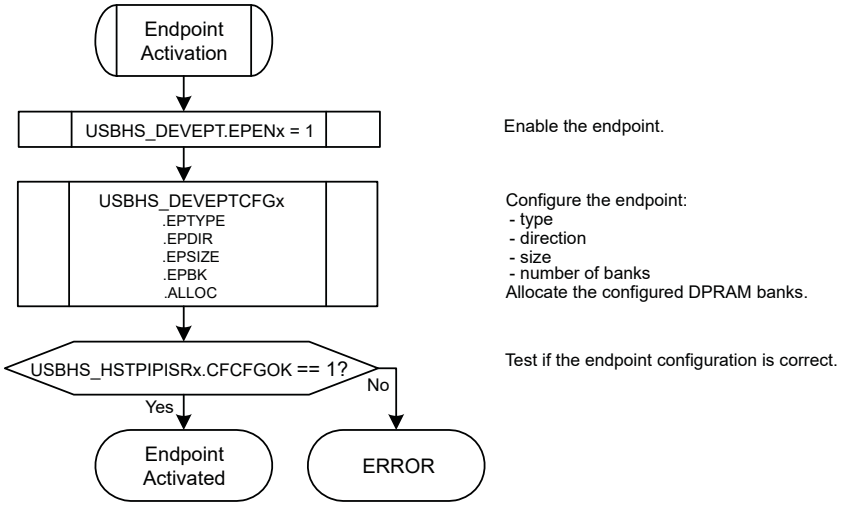
In the end, the user has to write a zero to the USBHS_DEVEPT.EPRSTx bit to complete the reset operation and to start using the FIFO.

39.5.2.5 Endpoint Activation

The endpoint is maintained inactive and reset (see "Endpoint Reset" for more information) as long as it is disabled (USBHS_DEVEPT.EPENx = 0). USBHS_DEVEPTISRx.DTSEQ is also reset.

The algorithm represented in the following figure must be followed to activate an endpoint.

Figure 39-8. Endpoint Activation Algorithm



As long as the endpoint is not correctly configured (USBHS_HSTPIPISRx.CFGOK = 0), the controller does not acknowledge the packets sent by the host to this endpoint.

The USBHS_HSTPIPISRx.CFGOK bit is set provided that the configured size and number of banks are correct as compared to the endpoint maximal allowed values (see the Description of USB Pipes/Endpoints table) and to the maximal FIFO size (i.e., the DPRAM size).

See "DPRAM Management" for additional information.

39.5.2.6 Address Setup

The USB device address is set up according to the USB protocol.

- After all kinds of resets, the USB device address is 0.

- Device Connection (USBHS_HSTISR.DCONNI)
- Device Disconnection (USBHS_HSTISR.DDISCI)
- USB Reset Sent (USBHS_HSTISR.RSTI)
- Downstream Resume Sent (USBHS_HSTISR.RSMEDI)
- Upstream Resume Received (USBHS_HSTISR.RXRSMI)
- Host Start of Frame (USBHS_HSTISR.HSOFI)
- Host Wakeup (USBHS_HSTISR.HWUPI)
- Pipe x (USBHS_HSTISR.PEP_x)
- DMA Channel x (USBHS_HSTISR.DMAxINT)

There is no exception host global interrupt.

Pipe Interrupts

The processing host pipe interrupts are:

- Received IN Data (USBHS_HSTPIPISRx.RXINI)
- Transmitted OUT Data (USBHS_HSTPIPISRx.TXOUTI)
- Transmitted SETUP (USBHS_HSTPIPISRx.TXSTPI)
- Short Packet (USBHS_HSTPIPISRx.SHORTPACKETI)
- Number of Busy Banks (USBHS_HSTPIPISRx.NBUSYBK)

The exception host pipe interrupts are:

- Underflow (USBHS_HSTPIPISRx.UNDERFI)
- Pipe Error (USBHS_HSTPIPISRx.PERRI)
- NAKed (USBHS_HSTPIPISRx.NAKEDI)
- Overflow (USBHS_HSTPIPISRx.OVERFI)
- Received STALLed (USBHS_HSTPIPISRx.RXSTALLDI)
- CRC Error (USBHS_HSTPIPISRx.CRCERRI)
  DMA Interrupts

The processing host DMA interrupts are:

- The End of USB Transfer Status (USBHS_HSTDMASTATUSx.END_TR_ST)
- The End of Channel Buffer Status (USBHS_HSTDMASTATUSx.END_BF_ST)
- The Descriptor Loaded Status (USBHS_HSTDMASTATUSx.DESC_LDST)

There is no exception host DMA interrupt.

39.5.4 USB DMA Operation

USB packets of any length may be transferred when required by the USBHS. These transfers always feature sequential addressing. Such characteristics mean that in case of high USBHS throughput, both AHB ports benefit from “incrementing burst of unspecified length” since the average access latency of AHB slaves can then be reduced.

The DMA uses word “incrementing burst of unspecified length” of up to 256 beats for both data transfers and channel descriptor loading. A burst may last on the AHB busses for the duration of a whole USB packet transfer, unless otherwise broken by the AHB arbitration or the AHB 1-Kbyte boundary crossing.

Packet data AHB bursts may be locked on a DMA buffer basis for drastic overall AHB bus bandwidth performance boost with paged memories. This prevents large AHB bursts from being broken in case of conflict with other AHB bus masters, thus avoiding access latencies due to memory row changes. This

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 an IN pipe:

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 End of DMA Buffer Output Enable (USBHS_HSTDMACONTROL.END_B_EN) and Automatic Switch (USBHS_HSTPIPCFG.AUTOSW) = 1.

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

Bit 6 – RXSTALLDE Received STALLed Interrupt Enable

Value	Description
0	Cleared when USBHS_HSTPIPIDR.RXSTALLDEC = 1. This disables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.RXSTALLDE).
1	Set when USBHS_HSTPIPIER.RXSTALLDES = 1. This enables the Transmitted IN Data interrupt (USBHS_HSTPIPIMR.RXSTALLDE).

Bit 5 – OVERFIE Overflow Interrupt Enable

Bit 5 – OVERFIES Overflow Interrupt Enable

Bit 4 – NAKEDES NAKed Interrupt Enable

Bit 3 – PERRES Pipe Error Interrupt Enable

Bit 2 – UNDERFIES Underflow Interrupt Enable

Bit 1 – TXOUTES Transmitted OUT Data Interrupt Enable

Bit 0 – RXINES Received IN Data Interrupt Enable

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.

42.7.5 QSPI Status Register

Name: QSPI_SR
Offset: 0x10
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
								QSPIENS
Access								R
Reset								0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
						INSTRE	CSS	CSR
Access						R	R	R
Reset						0	0	0

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

Bit 24 – QSPIENS QSPI Enable Status

Value	Description
0	QSPI is disabled.
1	QSPI is enabled.

Bit 10 – INSTRE Instruction End Status (cleared on read)

Value	Description
0	No instruction end has been detected since the last read of QSPI_SR.
1	At least one instruction end has been detected since the last read of QSPI_SR.

Bit 9 – CSS Chip Select Status

Value	Description
0	The chip select is asserted.
1	The chip select is not asserted.

Bit 8 – CSR Chip Select Rise (cleared on read)

Value	Description
0	No chip select rise has been detected since the last read of QSPI_SR.
1	At least one chip select rise has been detected since the last read of QSPI_SR.

In Slave mode, the serial clock and word select clock are driven by an external master. I2SC_CK and I2SC_WS pins are inputs.

In Master mode, the user can configure the master clock, serial clock, and word select clock through the I2SC_MR. I2SC_MCK, I2SC_CK, and I2SC_WS pins are outputs and MCK is used to derive the I2SC clocks.

In Master mode, if the peripheral clock frequency is higher than 96 MHz, the GCLK[PID] from PMC must be selected as I2SC input clock by writing a ‘1’ in the I2SCxCC bit of the CCFG_PCCR register. Refer to the section “Bus Matrix (MATRIX)” for more details.

Audio codecs connected to the I2SC pins may require a master clock (I2SC_MCK) signal with a frequency multiple of the audio sample frequency (f_s), such as $256f_s$. When the I2SC is in Master mode, writing a ’1’ to I2SC_MR.IMCKMODE outputs MCK as master clock to the I2SC_MCK pin, and divides MCK to create the internal bit clock, output on the I2SC_CK pin. The clock division factor is defined by writing to I2SC_MR.IMCKFS and I2SC_MR.DATALENGTH, as described in the I2SC_MR.IMCKFS field description.

The master clock (I2SC_MCK) frequency is (2×16 × (IMCKFS + 1)) / (IMCKDIV + 1) times the sample frequency (f_s), i.e., I2SC_WS frequency.

Example: If the sampling rate is 44.1 kHz with an I2S master clock (I2SC_MCK) ratio of 256, the core frequency must be an integer multiple of 11.2896 MHz. Assuming an integer multiple of 4, the IMCKDIV field must be configured to 4; the field IMCKFS must then be set to 31.

The serial clock (I2SC_CK) frequency is 2 × Slot Length times the sample frequency (f_s), where Slot Length is defined in the following table.

Table 45-2. Slot Length

I2SC_MR.DATALENGTH	Word Length	Slot Length
0	32 bits	32
1	24 bits	32 if I2SC_MR.IWS = 0 24 if I2SC_MR.IWS = 1
2	20 bits	
3	18 bits	
4	16 bits	16
5	16 bits compact stereo	
6	8 bits	8
7	8 bits compact stereo	

⚠WARNING I2SC_MR.IMCKMODE must be written to ’1’ if the master clock frequency is strictly higher than the serial clock.

If a master clock output is not required, the MCK clock is used as I2SC_CK by clearing I2SC_MR.IMCKMODE. Alternatively, if the frequency of the MCK clock used is a multiple of the required I2SC_CK frequency, the I2SC_MCK to I2SC_CK divider can be used with the ratio defined by writing the I2SC_MR.IMCKFS field.

Bit 9 – TXEMPTY TXEMPTY Interrupt Mask

Bit 8 – TIMEOUT Timeout Interrupt Mask

Bit 7 – PARE Parity Error Interrupt Mask

Bit 6 – FRAME Framing Error Interrupt Mask

Bit 5 – OVRE Overrun Error Interrupt Mask

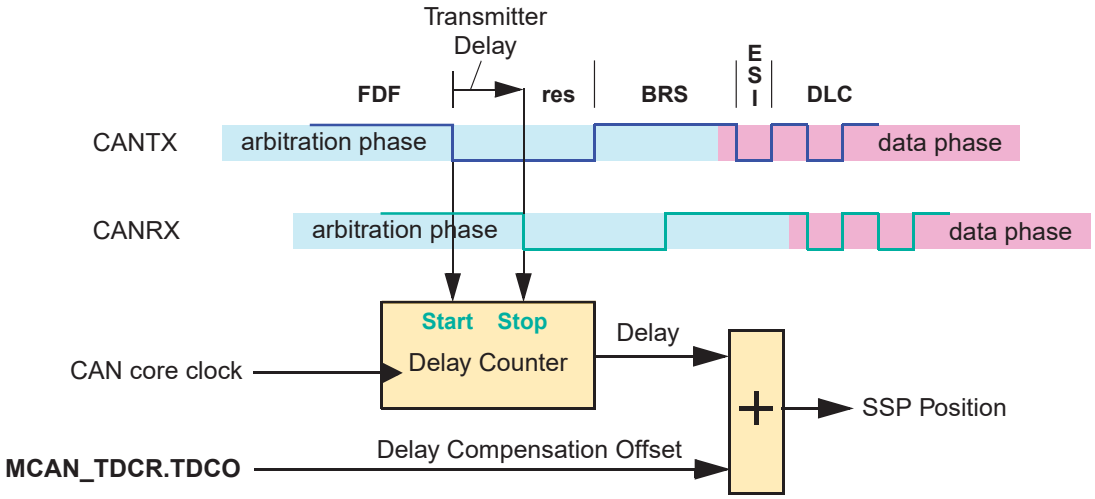
Bit 2 – RXBRK Receiver Break Interrupt Mask

Bit 1 – TXRDY TXRDY Interrupt Mask

Bit 0 – RXRDY RXRDY Interrupt Mask

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.

Figure 49-2. Transmitter Delay Measurement



To avoid that a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in a to early SSP position, the use of a transmitter delay compensation filter window can be enabled by programming MCAN_TDCR.TDCF.

This defines a minimum value for the SSP position. Dominant edges on CANRX, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least MCAN_TDCR.TDCF AND CANRX is low.

49.5.1.5 Restricted Operation Mode

In Restricted Operation mode, the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters are not incremented. The processor can set the MCAN into Restricted Operation mode by setting bit MCAN_CCCR.ASM. The bit can only be set by the processor when both MCAN_CCCR.CCE and MCAN_CCCR.INIT are set to '1'. The bit can be reset by the processor at any time.

Restricted Operation mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation mode, the processor has to reset MCAN_CCCR.ASM.

The Restricted Operation mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation mode after it has received a valid frame.

Note: The Restricted Operation Mode must not be combined with the Loop Back mode (internal or external).

49.5.1.6 Bus Monitoring Mode

The MCAN is set in Bus Monitoring mode by setting MCAN_CCCR.MON. In Bus Monitoring mode (see ISO11898-1, 10.12 Bus monitoring), the MCAN is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the MCAN is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the MCAN monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring mode, the Tx Buffer Request Pending register (MCAN_TXBRP) is held in reset state.

50.7.19 TC QDEC Interrupt Mask Register

Name: TC_QIMR
Offset: 0xD0
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
					MPE	QERR	DIRCHG	IDX
Access					R	R	R	R
Reset					0	0	0	0

Bit 3 – MPE Consecutive Missing Pulse Error

Value	Description
0	The interrupt on the maximum number of consecutive missing pulses specified in MAXCMP is disabled.
1	The interrupt on the maximum number of consecutive missing pulses specified in MAXCMP is enabled.

Bit 2 – QERR Quadrature Error

Value	Description
0	The interrupt on quadrature error is disabled.
1	The interrupt on quadrature error is enabled.

Bit 1 – DIRCHG Direction Change

Value	Description
0	The interrupt on rotation direction change is disabled.
1	The interrupt on rotation direction change is enabled.

Bit 0 – IDX Index

53.7.5 DACC Channel Disable Register

Name: DACC_CHDR
Offset: 0x14
Reset: –
Property: Write-only

This register can only be written if the WPEN bit is cleared in the DACC 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
							CH1	CH0
Access							W	W
Reset							0	–

Bits 0, 1 – CHx Channel x Disable

WARNING If the corresponding channel is disabled during a conversion or if it is disabled then re-enabled during a conversion, its associated analog value and its corresponding EOC flags in DACC_ISR are unpredictable.

Value	Description
0	No effect.
1	Disables the corresponding channel.

Symbol	Parameter	Conditions	Min	Max	Unit
SPI_{15}	NPCS hold after SPCK falling (slave)	3.3V domain	0	–	ns
		1.8V domain	0	–	ns

Timings are given for the 3.3V domain, with V_{DDIO} from 2.85V to 3.6V, maximum external capacitor = 40 pF.

Table 59-57. SPI Timings

Symbol	Parameter	Conditions	Min	Max	Unit
SPI_0	MISO Setup time before SPCK rises (master)	3.3V domain	12.4	–	ns
		1.7V domain	14.6	–	ns
SPI_1	MISO Hold time after SPCK rises (master)	3.3V domain	0	–	ns
		1.7V domain	0	–	ns
SPI_2	SPCK rising to MOSI Delay (master)	3.3V domain	-3.7	2.2	ns
		1.7V domain	-3.8	2.4	ns
SPI_3	MISO Setup time before SPCK falls (master)	3.3V domain	12.6	–	ns
		1.7V domain	15.13	–	ns
SPI_4	MISO Hold time after SPCK falls (master)	3.3V domain	0	–	ns
		1.7V domain	0	–	ns
SPI_5	SPCK falling to MOSI Delay (master)	3.3V domain	-3.6	2.0	ns
		1.7V domain	-3.3	2.8	ns
SPI_6	SPCK falling to MISO Delay (slave)	3.3V domain	3.0	11.9	ns
		1.7V domain	3.5	13.9	ns
SPI_7	MOSI Setup time before SPCK rises (slave)	3.3V domain	1.2	–	ns
		1.7V domain	1.5	–	ns
SPI_8	MOSI Hold time after SPCK rises (slave)	3.3V domain	0.6	–	ns
		1.7 domain	0.8	–	ns
SPI_9	SPCK rising to MISO Delay (slave)	3.3V domain	3.0	12.0	ns
		1.7V domain	3.4	13.7	ns
SPI_{10}	MOSI Setup time before SPCK falls (slave)	3.3V domain	1.2	–	ns
		1.7V domain	1.5	–	ns
SPI_{11}	MOSI Hold time after SPCK falls (slave)	3.3V domain	0.6	–	ns
		1.7V domain	0.8	–	ns
SPI_{12}	NPCS setup to SPCK rising (slave)	3.3V domain	3.9	–	ns
		1.7V domain	4.4	–	ns