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, EBI/EMI, 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	114
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 24x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsame70q21a-an

31.20.19 PMC Fast Startup Polarity Register

Name: PMC_FSPR
Offset: 0x0074
Reset: 0x00000000
Property: Read/Write

This register can only be written if the WPEN bit is cleared in the PMC 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
	FSTP15	FSTP14	FSTP13	FSTP12	FSTP11	FSTP10	FSTP9	FSTP8
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	FSTP7	FSTP6	FSTP5	FSTP4	FSTP3	FSTP2	FSTP1	FSTP0
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 – FSTP Fast Startup Input Polarity x bits
Defines the active polarity of the corresponding wake-up input. If the corresponding wake-up input is enabled and at the FSTP level, it enables a fast restart signal.

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.

38.8.57 GMAC Carrier Sense Errors Register

Name: GMAC_CSE
Offset: 0x14C
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
							CSR[9:8]	
Access							R	R
Reset							0	0

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

Bits 9:0 – CSR[9:0] Carrier Sense Error
This register counts the number of frames transmitted with carrier sense was not seen during transmission or where carrier sense was de-asserted after being asserted in a transmit frame without collision (no underrun). Only incremented in half duplex mode. The only effect of a carrier sense error is to increment this register. The behavior of the other statistics registers is unaffected by the detection of a carrier sense error.

38.8.66 GMAC 128 to 255 Byte Frames Received Register

Name: GMAC_TBFR255
Offset: 0x170
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
	NFRX[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
	NFRX[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
	NFRX[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
	NFRX[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – NFRX[31:0] 128 to 255 Byte Frames Received without Error
This bit field counts the number of 128 to 255 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.

38.8.85 GMAC 1588 Timer Seconds Low Register

Name: GMAC_TSL
Offset: 0x1D0
Reset: 0x00000000
Property: -

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

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

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

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

Bits 31:0 – TCS[31:0] Timer Count in Seconds
This register is writable. It increments by 1 when the IEEE 1588 nanoseconds counter counts to one second. It may also be incremented when the Timer Adjust Register is written.

38.8.104 GMAC Receive Buffer Size Register Priority Queue x

Name: GMAC_RBSRPQx
Offset: 0x04A0 + x*0x04 [x=0..4]
Reset: 0x00000002
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
	RBS[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0

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

Bits 15:0 – RBS[15:0] Receive Buffer Size
DMA receive buffer size in AHB system memory. The value defined by these bits determines the size of buffer to use in main AHB system memory when writing received data.

The value is defined in multiples of 64 Bytes such that a value of 0x01 corresponds to buffers of 64 Bytes, 0x02 corresponds to 128 Bytes etc.

Examples:

- 0x18: 1536 Bytes (1 × max length frame/buffer)
- 0xA0: 10240 Bytes (1 × 10K jumbo frame/buffer)

Note: This value should never be written as zero.

4. USBHS_DEVCTRL.RMWKUP is cleared at the end of the upstream resume.
5. When the controller detects a valid “End of Resume” signal from the host, the End of Resume (USBHS_DEVISR.EORSM) interrupt is set.

39.5.2.10 STALL Request

For each endpoint, the STALL management is performed using:

- the STALL Request (USBHS_DEVEPTIMRx.STALLRQ) bit to initiate a STALL request,
- the STALLed Interrupt (USBHS_DEVEPTISRx.STALLEDI) bit, which is set when a STALL handshake has been sent.

To answer the next request with a STALL handshake, USBHS_DEVEPTIMRx.STALLRQ has to be set by writing a one to the STALL Request Set (USBHS_DEVEPTIERx.STALLRQS) bit. All following requests are discarded (USBHS_DEVEPTISRx.RXOUTI, etc. is not be set) and handshaked with a STALL until the USBHS_DEVEPTIMRx.STALLRQ bit is cleared, which is done when a new SETUP packet is received (for control endpoints) or when the STALL Request Clear (USBHS_DEVEPTIMRx.STALLRQC) bit is written to one.

Each time a STALL handshake is sent, the USBHS_DEVEPTISRx.STALLEDI bit is set by the USBHS and the PEP_x interrupt is set.

Special Considerations for Control Endpoints

If a SETUP packet is received into a control endpoint for which a STALL is requested, the Received SETUP Interrupt (USBHS_DEVEPTISRx.RXSTPI) bit is set and USBHS_DEVEPTIMRx.STALLRQ and USBHS_DEVEPTISRx.STALLEDI are cleared. The SETUP has to be ACKed.

This simplifies the enumeration process management. If a command is not supported or contains an error, the user requests a STALL and can return to the main task, waiting for the next SETUP request.

STALL Handshake and Retry Mechanism

The retry mechanism has priority over the STALL handshake. A STALL handshake is sent if the USBHS_DEVEPTIMRx.STALLRQ bit is set and if no retry is required.

39.5.2.11 Management of Control Endpoints

Overview

A SETUP request is always ACKed. When a new SETUP packet is received, the USBHS_DEVEPTISRx.RXSTPI is set; the Received OUT Data Interrupt (USBHS_DEVEPTISRx.RXOUTI) bit is not.

The FIFO Control (USBHS_DEVEPTIMRx.FIFOCON) bit and the Read/Write Allowed (USBHS_DEVEPTISRx.RWALL) bit are irrelevant for control endpoints. The user never uses them on these endpoints. When read, their values are always zero.

Control endpoints are managed using:

- the USBHS_DEVEPTISRx.RXSTPI bit, which is set when a new SETUP packet is received and which is cleared by firmware to acknowledge the packet and to free the bank;
- the USBHS_DEVEPTISRx.RXOUTI bit, which is set when a new OUT packet is received and which is cleared by firmware to acknowledge the packet and to free the bank;
- the Transmitted IN Data Interrupt (USBHS_DEVEPTISRx.TXINI) bit, which is set when the current bank is ready to accept a new IN packet and which is cleared by firmware to send the packet.
  Control Write

Bit 20 – RTOE Response Time-out Error Interrupt Disable

Bit 19 – RENDE Response End Bit Error Interrupt Disable

Bit 18 – RCRCE Response CRC Error Interrupt Disable

Bit 17 – RDIRE Response Direction Error Interrupt Disable

Bit 16 – RINDE Response Index Error Interrupt Disable

Bit 13 – CSRCV Completion Signal received interrupt Disable

Bit 12 – SDIOWAIT SDIO Read Wait Operation Status Interrupt Disable

Bit 8 – SDIOIRQA SDIO Interrupt for Slot A Interrupt Disable

Bit 5 – NOTBUSY Data Not Busy Interrupt Disable

Bit 4 – DTIP Data Transfer in Progress Interrupt Disable

Bit 3 – BLKE Data Block Ended Interrupt Disable

Bit 2 – TXRDY Transmit Ready Interrupt Disable

Bit 1 – RXRDY Receiver Ready Interrupt Disable

Bit 0 – CMDRDY Command Ready Interrupt Disable

41.8.4 SPI Transmit Data Register

Name: SPI_TDR
Offset: 0x0C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
								LASTXFER
Access								W
Reset								–

Bit	23	22	21	20	19	18	17	16
					PCS[3:0]			
Access					W	W	W	W
Reset					–	–	–	–

Bit	15	14	13	12	11	10	9	8
	TD[15:8]							
Access	W	W	W	W	W	W	W	W
Reset	–	–	–	–	–	–	–	–

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

Bit 24 – LASTXFER Last Transfer
This field is only used if variable peripheral select is active (SPI_MR.PS = 1).

Value	Description
0	No effect
1	The current NPCS is deasserted after the transfer of the character written in TD. When SPI_CSRx.CSAAT is set, the communication with the current serial peripheral can be closed by raising the corresponding NPCS line as soon as TD transfer is completed.

Bits 19:16 – PCS[3:0] Peripheral Chip Select
This field is only used if variable peripheral select is active (SPI_MR.PS = 1).

If SPI_MR.PCSDEC = 0:

PCS = xxx0 NPCS[3:0] = 1110

PCS = xx01 NPCS[3:0] = 1101

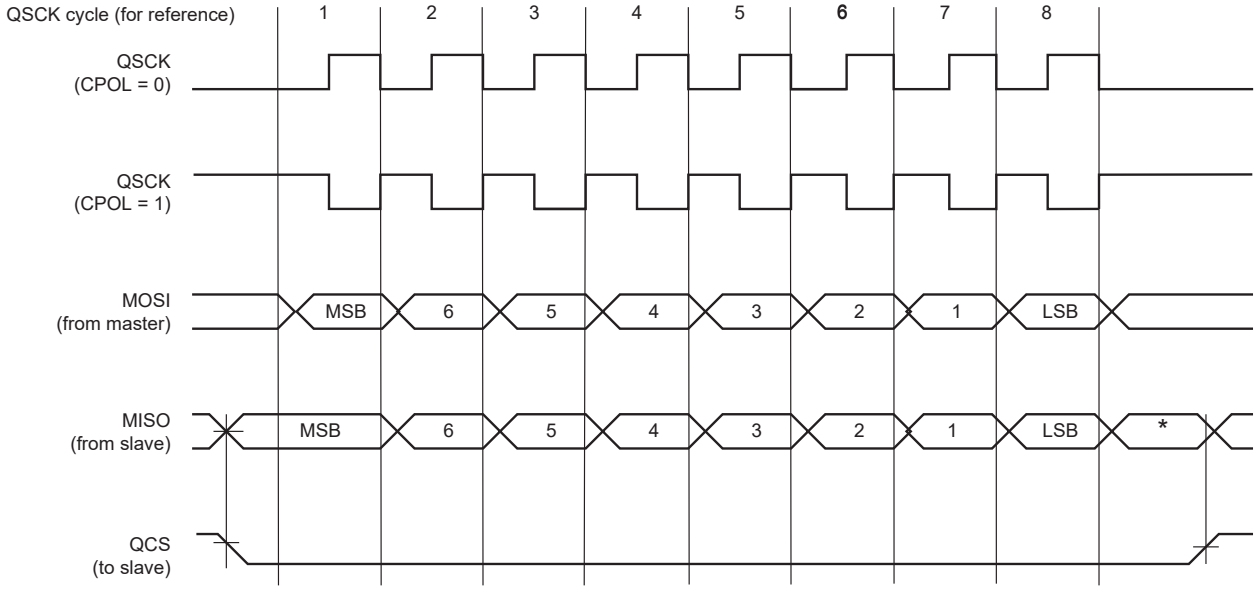
PCS = x011 NPCS[3:0] = 1011

PCS = 0111 NPCS[3:0] = 0111

PCS = 1111 forbidden (no peripheral is selected)

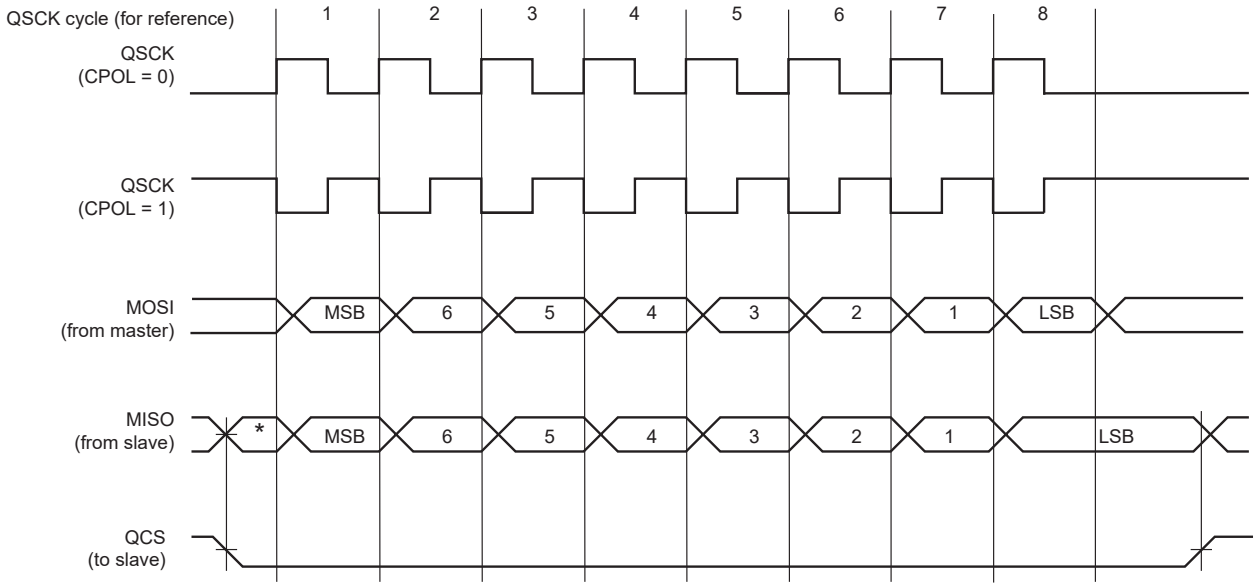
(x = don’t care)

Figure 42-2. QSPI Transfer Format (QSPI_SCR.CPHA = 0, 8 bits per transfer)



* Not defined, but normally MSB of previous character received.

Figure 42-3. QSPI Transfer Format (QSPI_SCR.CPHA = 1, 8 bits per transfer)



* Not defined but normally LSB of previous character transmitted.

42.6.3 Transfer Delays

The figure below shows several consecutive transfers while the chip select is active. Three delays can be programmed to modify the transfer waveforms:

- The delay between the deactivation and the activation of QCS, programmed by writing QSPI_MR.DLYCS. Allows to adjust the minimum time of QCS at high level.
- The delay before QSCK, programmed by writing QSPI_SR.DLYBS. Allows the start of QSCK to be delayed after the chip select has been asserted.

If a write access to a write-protected register is detected, the WPVS flag in the USART Write Protection Status Register (US_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 US_WPSR.

The following registers can be write-protected:

- USART Mode Register
- USART Baud Rate Generator Register
- USART Receiver Timeout Register
- USART Transmitter Timeguard Register
- USART Manchester Configuration Register
- USART LON Mode Register
- USART LON Beta1 Tx Register
- USART LON Beta1 Rx Register
- USART LON Priority Register
- USART LON IDT Tx Register
- USART LON IDT Rx Register
- USART IC DIFF Register

48.7.13 HBI Channel Busy 0 Register

Name: MLB_HCBR0
Offset: 0x098
Reset: 0x00000000
Property: Read-only

The HC can determine which channel(s) are busy by reading the HBI Channel Busy Registers (HCBRn). An HBI channel is busy if:

• it is currently loaded into one of the two AGUs

• the channel is enabled, CE = 1 from the Channel Allocation Table (CTR Address Mapping), and

• the DMA is active

When an HBI channel is busy, hardware may write back its local copy of the channel descriptor at any time. System software should not write a CDT descriptor for a channel that is busy. Only two HBI channels can be busy at any given time. Each bit of HCBRn is read-only.

Bit	31	30	29	28	27	26	25	24
	CHB: Bitwise Channel Busy Bit [31[31:24]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CHB: Bitwise Channel Busy Bit [31[23:16]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CHB: Bitwise Channel Busy Bit [31[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHB: Bitwise Channel Busy Bit [31[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHB: Bitwise Channel Busy Bit [31[31:0] 0]
CHB[n] = 1 indicates that channel n is busy.

49.5.5.4 Tx Queue

Tx Queue operation is configured by programming MCAN_TXBC.TFQM to '1'. Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New messages have to be written to the Tx Buffer referenced by the Put Index MCAN_TXFQS.TFQPI. An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (MCAN_TXFQS.TFQF = '1'), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

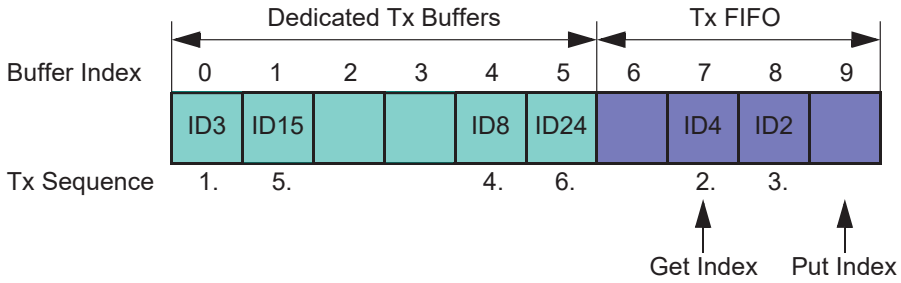
The application may use register MCAN_TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (see the table Tx Buffer / FIFO / Queue Element Size). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS.TFQPI (0…31) × Element Size to the Tx Buffer Start Address MCAN_TXBC.TBSA.

49.5.5.5 Mixed Dedicated Tx Buffers / Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of dedicated Tx Buffers and a Tx FIFO. The number of dedicated Tx Buffers is configured by MCAN_TXBC.NDTB. The number of Tx Buffers assigned to the Tx FIFO is configured by MCAN_TXBC.TFQS. In case MCAN_TXBC.TFQS is programmed to zero, only dedicated Tx Buffers are used.

Figure 49-10. Example of Mixed Configuration Dedicated Tx Buffers / Tx FIFO



Tx prioritization:

- Scan dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by MCAN_TXFS.TFGI)
- Buffer with lowest Message ID gets highest priority and is transmitted next

49.5.5.6 Mixed Dedicated Tx Buffers / Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of dedicated Tx Buffers and a Tx Queue. The number of dedicated Tx Buffers is configured by MCAN_TXBC.NDTB. The number of Tx Queue Buffers is configured by MCAN_TXBC.TFQS. In case MCAN_TXBC.TFQS is programmed to zero, only dedicated Tx Buffers are used.

Bit 0 – RF0N Receive FIFO 0 New Message

Value	Description
0	No new message written to Receive FIFO 0.
1	New message written to Receive FIFO 0.

49.6.18 MCAN Interrupt Line Select Register

Name: MCAN_ILS
Offset: 0x58
Reset: 0x00000000
Property: Read/Write

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register to one of the two module interrupt lines.

0: Interrupt assigned to interrupt line MCAN_INT0.

1: Interrupt assigned to interrupt line MCAN_INT1.

Bit	31	30	29	28	27	26	25	24
			ARAL	PEDL	PEAL	WDIL	BOL	EWL
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	EPL	ELOL			DRXL	TOOL	MRAFL	TSWL
Access	R/W	R/W			R/W	R/W	R/W	R/W
Reset	0	0			0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML
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
	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
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 29 – ARAL Access to Reserved Address Line

Bit 28 – PEDL Protocol Error in Data Phase Line

Bit 27 – PEAL Protocol Error in Arbitration Phase Line

Bit 26 – WDIL Watchdog Interrupt Line

Bit 25 – BOL Bus_Off Status Interrupt Line

Bit 24 – EWL Warning Status Interrupt Line

Bit 23 – EPL Error Passive Interrupt Line

Bit 22 – ELOL Error Logging Overflow Interrupt Line

Bit 19 – DRXL Message stored to Dedicated Receive Buffer Interrupt Line

50.7.10 TC Interrupt Status Register

Name: TC_SRx
Offset: 0x20 + x*0x40 [x=0..2]
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
						MTIOB	MTIOA	CLKSTA
Access						R	R	R
Reset						0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit 18 – MTIOB TIOBx Mirror

Value	Description
0	TIOBx is low. If TC_CMRx.WAVE = 0, TIOBx pin is low. If TC_CMRx.WAVE = 1, TIOBx is driven low.
1	TIOBx is high. If TC_CMRx.WAVE = 0, TIOBx pin is high. If TC_CMRx.WAVE = 1, TIOBx is driven high.

Bit 17 – MTIOA TIOAx Mirror

Value	Description
0	TIOAx is low. If TC_CMRx.WAVE = 0, TIOAx pin is low. If TC_CMRx.WAVE = 1, TIOAx is driven low.
1	TIOAx is high. If TC_CMRx.WAVE = 0, TIOAx pin is high. If TC_CMRx.WAVE = 1, TIOAx is driven high.

Bit 16 – CLKSTA Clock Enabling Status

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

51.7.8 PWM Interrupt Status Register 1

Name: PWM_ISR1
Offset: 0x1C
Reset: 0x00000000
Property: Read-only

Note: Reading PWM_ISR1 automatically clears CHIDx and FCHIDx flags.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
					FCHID3	FCHID2	FCHID1	FCHID0
Access					R	R	R	R
Reset					0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					R	R	R	R
Reset					0	0	0	0

Bits 16, 17, 18, 19 – FCHIDx Fault Protection Trigger on Channel x

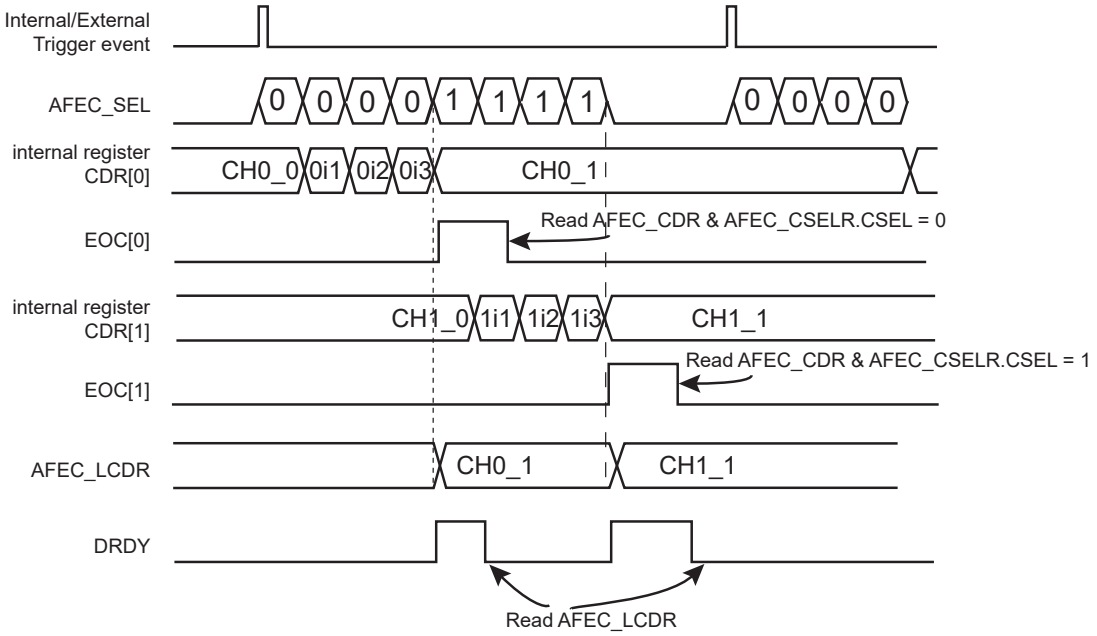
Value	Description
0	No new trigger of the fault protection since the last read of PWM_ISR1.
1	At least one trigger of the fault protection since the last read of PWM_ISR1.

Bits 0, 1, 2, 3 – CHIDx Counter Event on Channel x

Value	Description
0	No new counter event has occurred since the last read of PWM_ISR1.
1	At least one counter event has occurred since the last read of PWM_ISR1.

Figure 52-13. Digital Averaging Function Waveforms on a Single Trigger Event, Non-interleaved

AFEC_EMR.EMR = 2, STM = 1, AFEC_CHSR[7:0] = 0xFF and AFEC_MR.USEQ = 1
AFEC_SEQ1R = 0x1111_0000



Note: 0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0/1_0/1 are final results of average function.

52.6.15 Automatic Error Correction

The AFEC features automatic error correction of conversion results. Offset and gain error corrections are available. The correction can be enabled for each channel and correction values (offset and gain) are defined per Sample & Hold unit.

To enable error correction, the ECORR bit must be set in the AFEC Channel Error Correction register (AFEC_CECR). The offset and gain values used to compensate the results are set per Sample & Hold unit basis using the AFEC Correction Select register (AFEC_COSR) and the AFEC Correction Values register (AFEC_CVR). AFEC_COSR is used to select the Sample & Hold unit to be displayed in AFEC_CVR. This selection applies both to read and write operations in AFEC_CVR.

AFEC_CVR.OFFSETCORR and AFEC_CVR.GAINCORR must be filled with the values of corrective data. This data is computed from two measurement points in signed format. The correction is the same for all functional modes.

The final conversion result after error correction is obtained using the following formula, which is implemented after averaging in 2's complement format, with:

- OFFSETCORR—the offset correction value. OFFSETCORR is a signed value.
- GAINCORR—the gain correction value
- Gs—the value 15

$$\text{Corrected Data} = (\text{Converted Data} + \text{OFFSETCORR}) \times \frac{\text{GAINCORR}}{2^{(\text{Gs})}}$$

Symbol	VDDIO Supply	1.8V Domain	3.3V Domain	1.8V Domain	3.3V Domain	Unit
	Parameter	Min		Max		
SMC_{18}	NCS low before NWE high	(NWE_SETUP - NCS_RD_SETUP + NWE_PULSE) × t_{CPMCK} - 3.2	(NWE_SETUP - NCS_RD_SETUP + NWE_PULSE) × t_{CPMCK} - 2.2	–	–	ns
HOLD Settings (NWE_HOLD ≠ 0)						
SMC_{19}	NWE High to Data OUT, NBS0/A0 NBS1, A1, A2–A25 change	NWE_HOLD × t_{CPMCK} - 4.6	NWE_HOLD × t_{CPMCK} - 3.9	–	–	ns
SMC_{20}	NWE High to NCS Inactive (1)	(NWE_HOLD - NCS_WR_HOLD) × t_{CPMCK} - 3.9	(NWE_HOLD - NCS_WR_HOLD) × t_{CPMCK} - 3.6	–	–	ns
NO HOLD Settings (NWE_HOLD = 0)						
SMC_{21}	NWE High to Data OUT, NBS0/A0 NBS1, A1, A2–A25, NCS change(1)	2.1	1.5	–	–	ns

Note:
Hold length = total cycle duration - setup duration - pulse duration. “hold length” is for “NCS_WR_HOLD length” or “NWE_HOLD length”

Table 58-63. SMC Write NCS Controlled (WRITE_MODE = 0)

Symbol	VDDIO Supply	1.8V Domain	3.3V Domain	1.8V Domain	3.3V Domain	Unit
	Parameter	Min		Max		
SMC_{22}	Data Out Valid before NCS High	NCS_WR_PULSE × t_{CPMCK} - 2.8	NCS_WR_PULSE × t_{CPMCK} - 3.9	—	—	ns
SMC_{23}	NCS Pulse Width	NCS_WR_PULSE × t_{CPMCK} - 0.9	NCS_WR_PULSE × t_{CPMCK} - 0.2	—	—	ns
SMC_{24}	A0–A22 valid before NCS low	NCS_WR_SETUP × t_{CPMCK} - 4.0	NCS_WR_SETUP × t_{CPMCK} - 4.6	—	—	ns
SMC_{25}	NWE low before NCS high	(NCS_WR_SETUP - NWE_SETUP + NCS pulse) × t_{CPMCK} - 4.6	(NCS_WR_SETUP - NWE_SETUP + NCS pulse) × t_{CPMCK} - 4.6	—	—	ns

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Trademarks