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	HCS12X
Core Size	16-Bit
Speed	50MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	119
Program Memory Size	768KB (768K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	48K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 24x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°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/nxp-semiconductors/s912xep768j4magr

Appendix A
Electrical Characteristics

1.2.3.52 PM7 / TXCAN3 / TXCAN4 / TXD3 — Port M I/O Pin 7

PM7 is a general-purpose input or output pin. It can be configured as the transmit pin TXCAN of the scalable controller area network controller 3 or 4 (CAN3 or CAN4). PM7 can be configured as the transmit pin TXD3 of the serial communication interface 3 (SCI3).

1.2.3.53 PM6 / RXCAN3 / RXCAN4 / RXD3 — Port M I/O Pin 6

PM6 is a general-purpose input or output pin. It can be configured as the receive pin RXCAN of the scalable controller area network controller 3 or 4 (CAN3 or CAN4). PM6 can be configured as the receive pin RXD3 of the serial communication interface 3 (SCI3).

1.2.3.54 PM5 / TXCAN0 / TXCAN2 / TXCAN4 / SCK0 — Port M I/O Pin 5

PM5 is a general-purpose input or output pin. It can be configured as the transmit pin TXCAN of the scalable controller area network controllers 0, 2 or 4 (CAN0, CAN2, or CAN4). It can be configured as the serial clock pin SCK of the serial peripheral interface 0 (SPI0).

1.2.3.55 PM4 / RXCAN0 / RXCAN2 / RXCAN4 / MOSI0 — Port M I/O Pin 4

PM4 is a general-purpose input or output pin. It can be configured as the receive pin RXCAN of the scalable controller area network controllers 0, 2, or 4 (CAN0, CAN2, or CAN4). It can be configured as the master output (during master mode) or slave input pin (during slave mode) MOSI for the serial peripheral interface 0 (SPI0).

1.2.3.56 PM3 / TXCAN1 / TXCAN0 / $\overline{SS0}$ — Port M I/O Pin 3

PM3 is a general-purpose input or output pin. It can be configured as the transmit pin TXCAN of the scalable controller area network controllers 1 or 0 (CAN1 or CAN0). It can be configured as the slave select pin $\overline{SS}$ of the serial peripheral interface 0 (SPI0).

1.2.3.57 PM2 / RXCAN1 / RXCAN0 / MISO0 — Port M I/O Pin 2

PM2 is a general-purpose input or output pin. It can be configured as the receive pin RXCAN of the scalable controller area network controllers 1 or 0 (CAN1 or CAN0). It can be configured as the master input (during master mode) or slave output pin (during slave mode) MISO for the serial peripheral interface 0 (SPI0).

1.2.3.58 PM1 / TXCAN0 — Port M I/O Pin 1

PM1 is a general-purpose input or output pin. It can be configured as the transmit pin TXCAN of the scalable controller area network controller 0 (CAN0).

1.2.3.59 PM0 / RXCAN0 — Port M I/O Pin 0

PM0 is a general-purpose input or output pin. It can be configured as the receive pin RXCAN of the scalable controller area network controller 0 (CAN0).

Table 2-5. PORTB Register Field Descriptions

Field	Description
7-0 PB	**Port B general purpose input/output data**—Data Register Port B pins 7 through 0 are associated with address outputs ADDR[7:0] respectively in expanded modes. In emulation modes the address is multiplexed with IVD[7:0]. In normal expanded mode pin 0 is related to the $\overline{UDS}$ input. When not used with the alternative function, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.5 Port A Data Direction Register (DDRA)

Address 0x0002 (PRR) Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset	0	0	0	0	0	0	0	0

Figure 2-3. Port A Data Direction Register (DDRA)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
   Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

Table 2-6. DDRA Register Field Descriptions

Field	Description
7-0 DDRA	**Port A Data Direction**— This register controls the data direction of pins 7 through 0. The external bus function forces the I/O state to be outputs for all associated pins. In this case the data direction bits will not change. When operating a pin as a general purpose I/O, the associated data direction bit determines whether it is an input or output. 1 Associated pin is configured as output. 0 Associated pin is configured as high-impedance input.

2.3.6 Port B Data Direction Register (DDRB)

Address 0x0003 (PRR) Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Reset	0	0	0	0	0	0	0	0

Figure 2-4. Port B Data Direction Register (DDRB)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
   Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

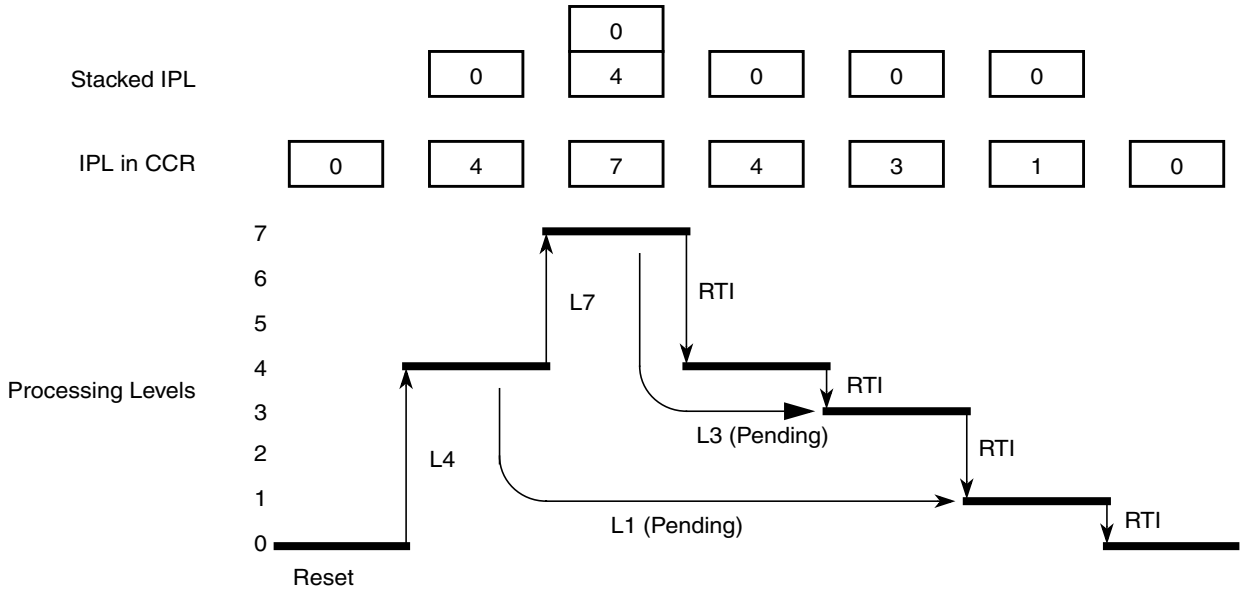

Figure 6-14. Interrupt Processing Example

6.5.3 Wake Up from Stop or Wait Mode

6.5.3.1 CPU Wake Up from Stop or Wait Mode

Only I bit maskable interrupt requests which are configured to be handled by the CPU are capable of waking the MCU from wait mode.

Since bus and core clocks are disabled in stop mode, only interrupt requests that can be generated without these clocks can wake the MCU from stop mode. These are listed in the device overview interrupt vector table. Only I bit maskable interrupt requests which are configured to be handled by the CPU are capable of waking the MCU from stop mode.

To determine whether an I bit maskable interrupt is qualified to wake up the CPU or not, the same settings as in normal run mode are applied during stop or wait mode:

- If the I bit in the CCR is set, all I bit maskable interrupts are masked from waking up the MCU.
- An I bit maskable interrupt is ignored if it is configured to a priority level below or equal to the current IPL in CCR.
- I bit maskable interrupt requests which are configured to be handled by the XGATE module are not capable of waking up the CPU.

The X bit maskable interrupt request can wake up the MCU from stop or wait mode at anytime, even if the X bit in CCR is set. If the X bit maskable interrupt request is used to wake-up the MCU with the X bit in the CCR set, the associated ISR is not called. The CPU then resumes program execution with the instruction following the WAI or STOP instruction. This features works following the same rules like any interrupt request, i.e. care must be taken that the X interrupt request used for wake-up remains active at least until the system begins execution of the instruction following the WAI or STOP instruction; otherwise, wake-up may not occur.

8.3.2.8.5 Debug Comparator Data High Register (DBGXDH)

Address: 0x002C

	7	6	5	4	3	2	1	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset	0	0	0	0	0	0	0	0

Figure 8-18. Debug Comparator Data High Register (DBGXDH)

Read: Anytime. See Table 8-29 for visible register encoding.

Write: If DBG not armed. See Table 8-29 for visible register encoding.

Table 8-35. DBGXAH Field Descriptions

Field	Description
7–0 Bits[15:8]	**Comparator Data High Compare Bits** — The Comparator data high compare bits control whether the selected comparator compares the data bus bits [15:8] to a logic one or logic zero. The comparator data compare bits are only used in comparison if the corresponding data mask bit is logic 1. This register is available only for comparators A and C. 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one

8.3.2.8.6 Debug Comparator Data Low Register (DBGXDL)

Address: 0x002D

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

Figure 8-19. Debug Comparator Data Low Register (DBGXDL)

Read: Anytime. See Table 8-29 for visible register encoding.

Write: If DBG not armed. See Table 8-29 for visible register encoding.

Table 8-36. DBGXDL Field Descriptions

Field	Description
7–0 Bits[7:0]	**Comparator Data Low Compare Bits** — The Comparator data low compare bits control whether the selected comparator compares the data bus bits [7:0] to a logic one or logic zero. The comparator data compare bits are only used in comparison if the corresponding data mask bit is logic 1. This register is available only for comparators A and C. 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one

CPU12X Information Byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CSD	CVA	0	CDV	0	0	0	0

Figure 8-25. CPU12X Information Byte CINF

Table 8-45. CINF Field Descriptions

Field	Description
7 CSD	**Source Destination Indicator** — This bit indicates if the corresponding stored address is a source or destination address. This is only used in Normal and Loop1 mode tracing. 0 Source address 1 Destination address
6 CVA	**Vector Indicator** — This bit indicates if the corresponding stored address is a vector address. Vector addresses are destination addresses, thus if CVA is set, then the corresponding CSD is also set. This is only used in Normal and Loop1 mode tracing. This bit has no meaning in Pure PC mode. 0 Indexed jump destination address 1 Vector destination address
4 CDV	**Data Invalid Indicator** — This bit indicates if the trace buffer entry is invalid. It is only used when tracing from both sources in Normal, Loop1 and Pure PC modes, to indicate that the CPU12X trace buffer entry is valid. 0 Trace buffer entry is invalid 1 Trace buffer entry is valid

CXINF Information Byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CFREE	CSZ	CRW	COCF	XACK	XSZ	XRW	XOCF

Figure 8-26. Information Byte CXINF

This describes the format of the information byte used only when tracing in Detail Mode. When tracing from the CPU12X in Detail Mode, information is stored to the trace buffer on all cycles except opcode fetch and free cycles. The XGATE entry stored on the same line is a snapshot of the XGATE program counter. In this case the CSZ and CRW bits indicate the type of access being made by the CPU12X, whilst the XACK and XOCF bits indicate if the simultaneous XGATE cycle is a free cycle (no bus acknowledge) or opcode fetch cycle. Similarly when tracing from the XGATE in Detail Mode, information is stored to the trace buffer on all cycles except opcode fetch and free cycles. The CPU12X entry stored on the same line is a snapshot of the CPU12X program counter. In this case the XSZ and XRW bits indicate the type of access being made by the XGATE, whilst the CFREE and COCF bits indicate if the simultaneous CPU12X cycle is a free cycle or opcode fetch cycle.

Table 8-46. CXINF Field Descriptions

Field	Description
7 CFREE	**CPU12X Free Cycle Indicator** — This bit indicates if the stored CPU12X address corresponds to a free cycle. This bit only contains valid information when tracing the XGATE accesses in Detail Mode. 0 Stored information corresponds to free cycle 1 Stored information does not correspond to free cycle

ADDL — Add Immediate 8 bit Constant (Low Byte) — ADDL

Operation

RD + $00:IMM8 ⇒ RD

Adds the content of register RD and an unsigned immediate 8 bit constant using binary addition and stores the result in the destination register RD. This instruction must be used first for a 16 bit immediate addition in conjunction with the ADDH instruction.

CCR Effects

N	Z	V	C
Δ	Δ	Δ	Δ

N: Set if bit 15 of the result is set; cleared otherwise.

Z: Set if the result is $0000; cleared otherwise.

V: Set if a two´s complement overflow resulted from the 8 bit operation; cleared otherwise.
$\overline{RD[15]}_{old}$ & $RD[15]_{new}$

C: Set if there is a carry from the bit 15 of the result; cleared otherwise.
$RD[15]_{old}$ & $\overline{RD[15]}_{new}$

Code and CPU Cycles

Source Form	Address Mode	Machine Code							Cycles
ADDL RD, #IMM8	IMM8	1	1	1	0	0	RD	IMM8	P

Table 11-14. Examples of IPLL Divider Settings[1]

f_{OSC}	REFDIV[5:0]	f_{REF}	REFFRQ[1:0]	SYNDIV[5:0]	f_{VCO}	VCOFRQ[1:0]	POSTDIV[4:0]	f_{PLL}	f_{BUS}
4MHz	$01	2MHz	01	$18	100MHz	11	$00	100MHz	50 MHz
8MHz	$03	2MHz	01	$18	100MHz	11	$00	100MHz	50 MHz
4MHz	$00	4MHz	01	$09	80MHz	01	$00	80MHz	40MHz
8MHz	$00	8MHz	10	$04	80MHz	01	$00	80MHz	40MHz
4MHz	$00	4MHz	01	$03	32MHz	00	$01	16MHz	8MHz
4MHz	$01	2MHz	01	$18	100MHz	11	$01	50MHz	25MHz

1. f_{PLL} and f_{BUS} values in this table may exceed maximum allowed frequencies for some devices. Refer to device information for maximum values.

11.4.1.1.1 IPLL Operation

The oscillator output clock signal (OSCCLK) is fed through the reference programmable divider and is divided in a range of 1 to 64 (REFDIV+1) to output the REFCLK. The VCO output clock, (VCOCLK) is fed back through the programmable loop divider and is divided in a range of 2 to 128 in increments of [2 x (SYNDIV +1)] to output the FBCLK. The VCOCLK is fed to the final programmable divider and is divided in a range of 1,2,4,6,8,... to 62 (2*POSTDIV) to output the PLLCLK. See Figure 11-15.

The phase detector then compares the FBCLK, with the REFCLK. Correction pulses are generated based on the phase difference between the two signals. The loop filter then slightly alters the DC voltage on the internal filter capacitor, based on the width and direction of the correction pulse.

The user must select the range of the REFCLK frequency and the range of the VCOCLK frequency to ensure that the correct IPLL loop bandwidth is set.

The lock detector compares the frequencies of the FBCLK, and the REFCLK. Therefore, the speed of the lock detector is directly proportional to the reference clock frequency. The circuit determines the lock condition based on this comparison.

If IPLL LOCK interrupt requests are enabled, the software can wait for an interrupt request and then check the LOCK bit. If interrupt requests are disabled, software can poll the LOCK bit continuously (during IPLL start-up, usually) or at periodic intervals. In either case, only when the LOCK bit is set, the PLLCLK can be selected as the source for the system and core clocks. If the IPLL is selected as the source for the system and core clocks and the LOCK bit is clear, the IPLL has suffered a severe noise hit and the software must take appropriate action, depending on the application.

- The LOCK bit is a read-only indicator of the locked state of the IPLL.
- The LOCK bit is set when the VCO frequency is within a certain tolerance, Δ_{Lock}, and is cleared when the VCO frequency is out of a certain tolerance, Δ_{unl}.
- Interrupt requests can occur if enabled (LOCKIE = 1) when the lock condition changes, toggling the LOCK bit.

Write: Anytime (No effect on (CC3, CC2, CC1, CC0))

Table 13-17. ATDSTAT0 Field Descriptions

Field	Description
7 SCF	**Sequence Complete Flag** — This flag is set upon completion of a conversion sequence. If conversion sequences are continuously performed (SCAN=1), the flag is set after each one is completed. This flag is cleared when one of the following occurs: A) Write “1” to SCF B) Write to ATDCTL5 (a new conversion sequence is started) C) If AFFC=1 and read of a result register 0 Conversion sequence not completed 1 Conversion sequence has completed
5 ETORF	**External Trigger Overrun Flag** — While in edge trigger mode (ETRIGLE=0), if additional active edges are detected while a conversion sequence is in process the overrun flag is set. This flag is cleared when one of the following occurs: A) Write “1” to ETORF B) Write to ATDCTL0,1,2,3,4, ATDCMPE or ATDCMPHT (a conversion sequence is aborted) C) Write to ATDCTL5 (a new conversion sequence is started) 0 No External trigger over run error has occurred 1 External trigger over run error has occurred
4 FIFOR	**Result Register Over Run Flag** — This bit indicates that a result register has been written to before its associated conversion complete flag (CCF) has been cleared. This flag is most useful when using the FIFO mode because the flag potentially indicates that result registers are out of sync with the input channels. However, it is also practical for non-FIFO modes, and indicates that a result register has been over written before it has been read (i.e. the old data has been lost). This flag is cleared when one of the following occurs: A) Write “1” to FIFOR B) Write to ATDCTL0,1,2,3,4, ATDCMPE or ATDCMPHT (a conversion sequence is aborted) C) Write to ATDCTL5 (a new conversion sequence is started) 0 No over run has occurred 1 Overrun condition exists (result register has been written while associated CCFx flag was still set)
3–0 CC[3:0]	**Conversion Counter** — These 4 read-only bits are the binary value of the conversion counter. The conversion counter points to the result register that will receive the result of the current conversion. E.g. CC3=0, CC2=1, CC1=1, CC0=0 indicates that the result of the current conversion will be in ATD Result Register 6. If in non-FIFO mode (FIFO=0) the conversion counter is initialized to zero at the begin and end of the conversion sequence. If in FIFO mode (FIFO=1) the register counter is not initialized. The conversion counters wraps around when its maximum value is reached. Aborting a conversion or starting a new conversion clears the conversion counter even if FIFO=1.

13.3.2.8 ATD Compare Enable Register (ATDCMPE)

Writes to this register will abort current conversion sequence.

Read: Anytime

Write: Anytime

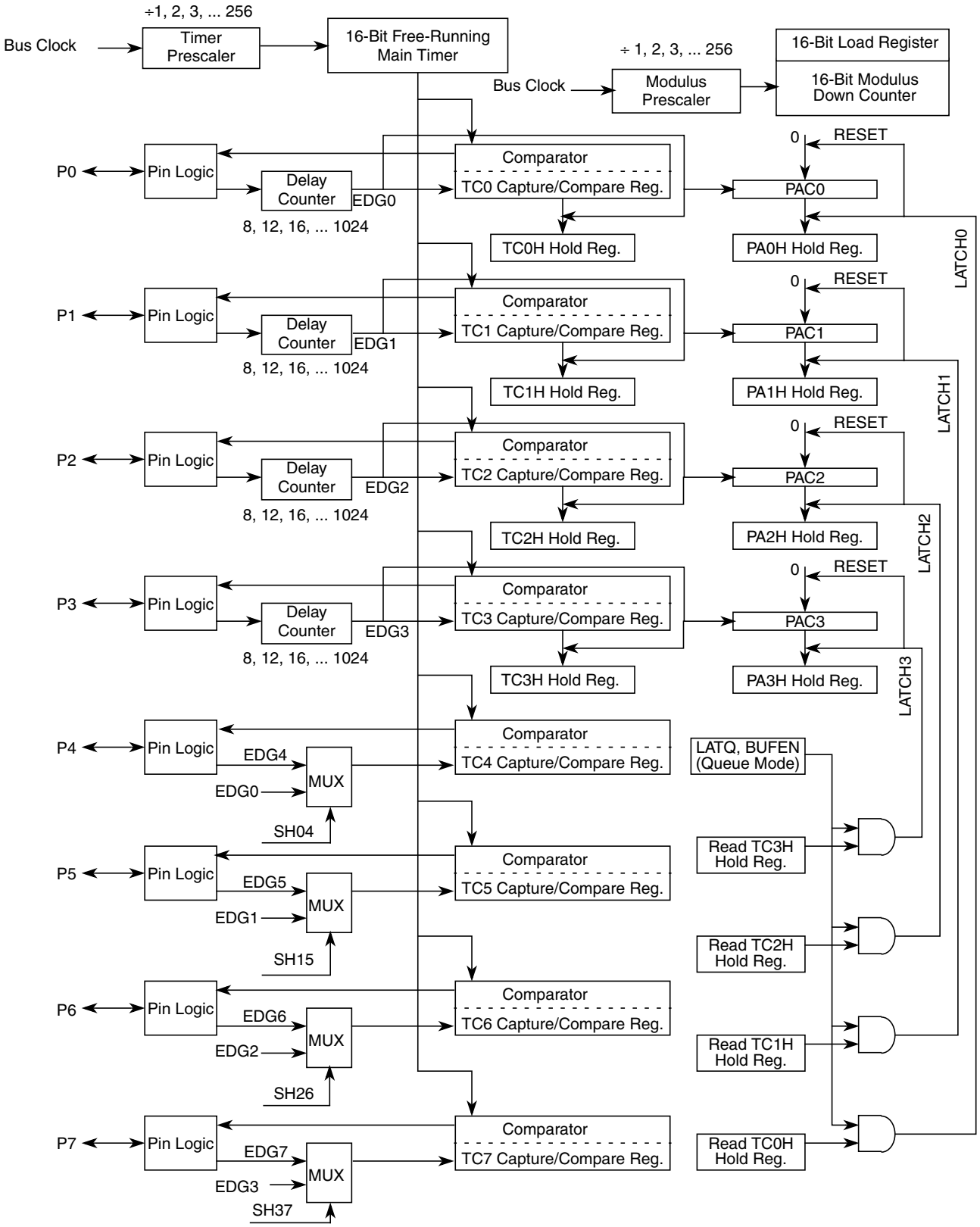

Figure 14-70. Detailed Timer Block Diagram in Queue Mode when PRNT = 1

Table 16-7. Baud Rate Prescaler

BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	Prescaler value (P)
0	0	0	0	0	0	1
0	0	0	0	0	1	2
0	0	0	0	1	0	3
0	0	0	0	1	1	4
:	:	:	:	:	:	:
1	1	1	1	1	1	64

16.3.2.4 MSCAN Bus Timing Register 1 (CANBTR1)

The CANBTR1 register configures various CAN bus timing parameters of the MSCAN module.

Module Base + 0x0003 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	SAMP	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
Reset:	0	0	0	0	0	0	0	0

Figure 16-7. MSCAN Bus Timing Register 1 (CANBTR1)

1. Read: Anytime
   Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 16-8. CANBTR1 Register Field Descriptions

Field	Description
7 SAMP	**Sampling** — This bit determines the number of CAN bus samples taken per bit time. 0 One sample per bit. 1 Three samples per bit(1). If SAMP = 0, the resulting bit value is equal to the value of the single bit positioned at the sample point. If SAMP = 1, the resulting bit value is determined by using majority rule on the three total samples. For higher bit rates, it is recommended that only one sample is taken per bit time (SAMP = 0).
6-4 TSEG2[2:0]	**Time Segment 2** — Time segments within the bit time fix the number of clock cycles per bit time and the location of the sample point (see Figure 16-44). Time segment 2 (TSEG2) values are programmable as shown in Table 16-9.
3-0 TSEG1[3:0]	**Time Segment 1** — Time segments within the bit time fix the number of clock cycles per bit time and the location of the sample point (see Figure 16-44). Time segment 1 (TSEG1) values are programmable as shown in Table 16-10.

1. In this case, PHASE_SEG1 must be at least 2 time quanta (Tq).

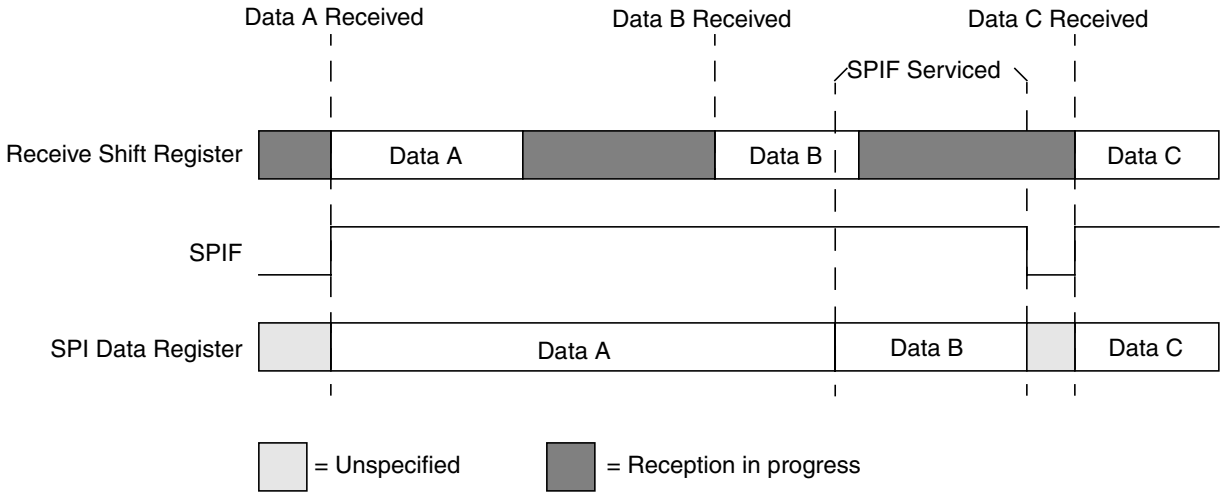

Figure 21-9. Reception with SPIF serviced in Time

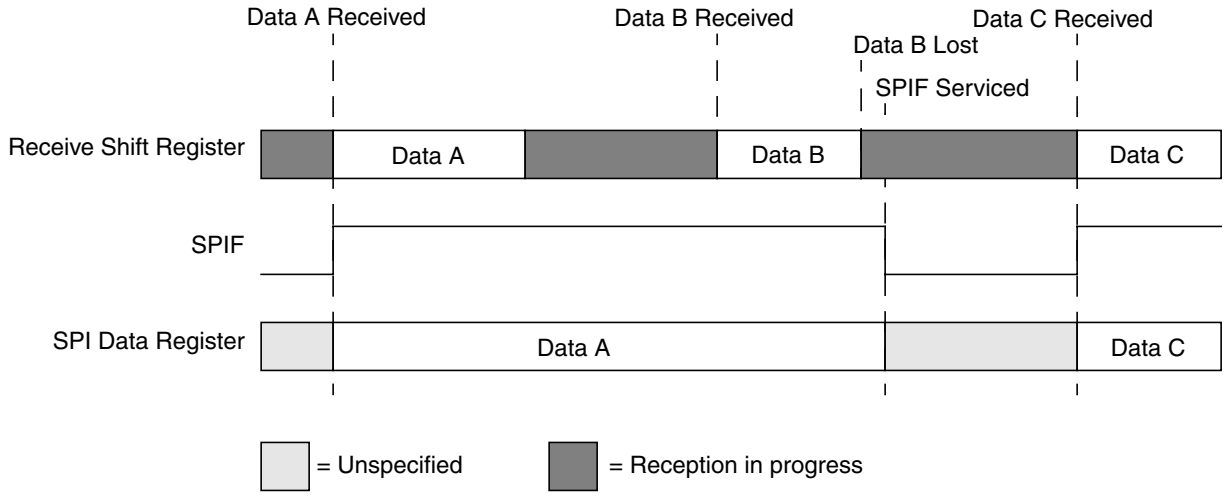

Figure 21-10. Reception with SPIF serviced too late

21.4 Functional Description

The SPI module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or SPI operation can be interrupt driven.

The SPI system is enabled by setting the SPI enable (SPE) bit in SPI control register 1. While SPE is set, the four associated SPI port pins are dedicated to the SPI function as:

- Slave select ($\overline{SS}$)
- Serial clock (SCK)
- Master out/slave in (MOSI)
- Master in/slave out (MISO)

Table 24-15. FCNFG Field Descriptions (continued)

Field	Description
1 FDFD	**Force Double Bit Fault Detect** — The FDFD bit allows the user to simulate a double bit fault during Flash array read operations and check the associated interrupt routine. The FDFD bit is cleared by writing a 0 to FDFD. The FECCR registers will not be updated during the Flash array read operation with FDFD set unless an actual double bit fault is detected. 0 Flash array read operations will set the DFDIF flag in the FERSTAT register only if a double bit fault is detected 1 Any Flash array read operation will force the DFDIF flag in the FERSTAT register to be set (see Section 24.3.2.7) and an interrupt will be generated as long as the DFDIE interrupt enable in the FERCNFG register is set (see Section 24.3.2.6)
0 FSFD	**Force Single Bit Fault Detect** — The FSFD bit allows the user to simulate a single bit fault during Flash array read operations and check the associated interrupt routine. The FSFD bit is cleared by writing a 0 to FSFD. The FECCR registers will not be updated during the Flash array read operation with FSFD set unless an actual single bit fault is detected. 0 Flash array read operations will set the SFDIF flag in the FERSTAT register only if a single bit fault is detected 1 Flash array read operation will force the SFDIF flag in the FERSTAT register to be set (see Section 24.3.2.7) and an interrupt will be generated as long as the SFDIE interrupt enable in the FERCNFG register is set (see Section 24.3.2.6)

24.3.2.6 Flash Error Configuration Register (FERCNFG)

The FERCNFG register enables the Flash error interrupts for the FERSTAT flags.

Offset Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	ERSERIE	PGMERIE	0	EPVIOLIE	ERSVIE1	ERSVIE0	DFDIE	SFDIE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 24-10. Flash Error Configuration Register (FERCNFG)

All assigned bits in the FERCNFG register are readable and writable.

Table 24-16. FERCNFG Field Descriptions

Field	Description
7 ERSERIE	**EEE Erase Error Interrupt Enable** — The ERSERIE bit controls interrupt generation when a failure is detected during an EEE erase operation. 0 ERSERIF interrupt disabled 1 An interrupt will be requested whenever the ERSERIF flag is set (see Section 24.3.2.8)
6 PGMERIE	**EEE Program Error Interrupt Enable** — The PGMERIE bit controls interrupt generation when a failure is detected during an EEE program operation. 0 PGMERIF interrupt disabled 1 An interrupt will be requested whenever the PGMERIF flag is set (see Section 24.3.2.8)
4 EPVIOLIE	**EEE Protection Violation Interrupt Enable** — The EPVIOLIE bit controls interrupt generation when a protection violation is detected during a write to the buffer RAM EEE partition. 0 EPVIOLIF interrupt disabled 1 An interrupt will be requested whenever the EPVIOLIF flag is set (see Section 24.3.2.8)

24.4 Functional Description

24.4.1 Flash Command Operations

Flash command operations are used to modify Flash memory contents or configure module resources for EEE operation.

The next sections describe:

- How to write the FCLKDIV register that is used to generate a time base (FCLK) derived from OSCCLK for Flash program and erase command operations
- The command write sequence used to set Flash command parameters and launch execution
- Valid Flash commands available for execution

24.4.1.1 Writing the FCLKDIV Register

Prior to issuing any Flash program or erase command after a reset, the user is required to write the FCLKDIV register to divide OSCCLK down to a target FCLK of 1 MHz. Table 24-9 shows recommended values for the FDIV field based on OSCCLK frequency.

NOTE

Programming or erasing the Flash memory cannot be performed if the bus clock runs at less than 1 MHz. Setting FDIV too high can destroy the Flash memory due to overstress. Setting FDIV too low can result in incomplete programming or erasure of the Flash memory cells.

When the FCLKDIV register is written, the FDIVLD bit is set automatically. If the FDIVLD bit is 0, the FCLKDIV register has not been written since the last reset. If the FCLKDIV register has not been written, any Flash program or erase command loaded during a command write sequence will not execute and the ACCERR bit in the FSTAT register will set.

24.4.1.2 Command Write Sequence

The Memory Controller will launch all valid Flash commands entered using a command write sequence.

Before launching a command, the ACCERR and FPVIOL bits in the FSTAT register must be clear (see Section 24.3.2.7) and the CCIF flag should be tested to determine the status of the current command write sequence. If CCIF is 0, the previous command write sequence is still active, a new command write sequence cannot be started, and all writes to the FCCOB register are ignored.

CAUTION

Writes to any Flash register must be avoided while a Flash command is active (CCIF=0) to prevent corruption of Flash register contents and Memory Controller behavior.

Table 25-64. Full Partition D-Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if a Load Data Field command sequence is currently active
		Set if command not available in current mode (see Table 25-30)
		Set if an invalid DFPART or ERPART selection is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read
FERSTAT	EPVIOLIF	None

25.4.2.16 Erase Verify D-Flash Section Command

The Erase Verify D-Flash Section command will verify that a section of code in the D-Flash user partition is erased. The Erase Verify D-Flash Section command defines the starting point of the data to be verified and the number of words.

Table 25-65. Erase Verify D-Flash Section Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x10	Global address [22:16] to identify the D-Flash block
001	Global address [15:0] of the first word to be verified	
010	Number of words to be verified	

Upon clearing CCIF to launch the Erase Verify D-Flash Section command, the Memory Controller will verify the selected section of D-Flash memory is erased. The CCIF flag will set after the Erase Verify D-Flash Section operation has completed.

Table 27-41. Load Data Field Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x05	Global address [22:16] to identify P-Flash block
001	Global address [15:0] of phrase location to be programmed(1)	
010	Word 0	
011	Word 1	
100	Word 2	
101	Word 3	

1. Global address [2:0] must be 000

Upon clearing CCIF to launch the Load Data Field command, the FCCOB registers will be transferred to the Memory Controller and be programmed in the block specified at the global address given with a future Program P-Flash command launched on a P-Flash block. The CCIF flag will set after the Load Data Field operation has completed. Note that once a Load Data Field command sequence has been initiated, the Load Data Field command sequence will be cancelled if any command other than Load Data Field or the future Program P-Flash is launched. Similarly, if an error occurs after launching a Load Data Field or Program P-Flash command, the associated Load Data Field command sequence will be cancelled.

Table 27-42. Load Data Field Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 101 at command launch
		Set if command not available in current mode (see Table 27-30)
		Set if an invalid global address [22:0] is supplied
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
		Set if a Load Data Field command sequence is currently active and the selected block has previously been selected in the same command sequence
		Set if a Load Data Field command sequence is currently active and global address [17:0] does not match that previously supplied in the same command sequence
	FPVIOL	Set if the global address [22:0] points to a protected area
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

27.4.2.6 Program P-Flash Command

The Program P-Flash operation will program a previously erased phrase in the P-Flash memory using an embedded algorithm.

28.4.3 Interrupts

The Flash module can generate an interrupt when a Flash command operation has completed or when a Flash command operation has detected an EEE error or an ECC fault.

Table 28-79. Flash Interrupt Sources

Interrupt Source	Interrupt Flag	Local Enable	Global (CCR) Mask
Flash Command Complete	CCIF (FSTAT register)	CCIE (FCNFG register)	I Bit
Flash EEE Erase Error	ERSERIF (FERSTAT register)	ERSERIE (FERCNFG register)	I Bit
Flash EEE Program Error	PGMERIF (FERSTAT register)	PGMERIE (FERCNFG register)	I Bit
Flash EEE Protection Violation	EPVIOLIF (FERSTAT register)	EPVIOLIE (FERCNFG register)	I Bit
Flash EEE Error Type 1 Violation	ERSVIF1 (FERSTAT register)	ERSVIE1 (FERCNFG register)	I Bit
Flash EEE Error Type 0 Violation	ERSVIF0 (FERSTAT register)	ERSVIE0 (FERCNFG register)	I Bit
ECC Double Bit Fault on Flash Read	DFDIF (FERSTAT register)	DFDIE (FERCNFG register)	I Bit
ECC Single Bit Fault on Flash Read	SFDIF (FERSTAT register)	SFDIE (FERCNFG register)	I Bit

NOTE

Vector addresses and their relative interrupt priority are determined at the MCU level.

28.4.3.1 Description of Flash Interrupt Operation

The Flash module uses the CCIF flag in combination with the CCIE interrupt enable bit to generate the Flash command interrupt request. The Flash module uses the ERSEIF, PGMEIF, EPVIOLIF, ERSVIF1, ERSVIF0, DFDIF and SFDIF flags in combination with the ERSEIE, PGMEIE, EPVIOLIE, ERSVIE1, ERSVIE0, DFDIE and SFDIE interrupt enable bits to generate the Flash error interrupt request. For a detailed description of the register bits involved, refer to Section 28.3.2.5, “Flash Configuration Register (FCNFG)”, Section 28.3.2.6, “Flash Error Configuration Register (FERCNFG)”, Section 28.3.2.7, “Flash Status Register (FSTAT)”, and Section 28.3.2.8, “Flash Error Status Register (FERSTAT)”.

The logic used for generating the Flash module interrupts is shown in Figure 28-27.

29.4.2.20 Disable EEPROM Emulation Command

The Disable EEPROM Emulation command causes the Memory Controller to suspend current EEE activity.

Table 29-73. Disable EEPROM Emulation Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x14	Not required

Upon clearing CCIF to launch the Disable EEPROM Emulation command, the Memory Controller will halt EEE operations at the next convenient point without clearing the EEE tag RAM or tag counter before setting the CCIF flag.

Table 29-74. Disable EEPROM Emulation Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if a Load Data Field command sequence is currently active
		Set if Full Partition D-Flash or Partition D-Flash command not previously run
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

29.4.2.21 EEPROM Emulation Query Command

The EEPROM Emulation Query command returns EEE partition and status variables.

Table 29-75. EEPROM Emulation Query Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x15	Not required
001	Return DFPART	
010	Return ERPART	
011	Return ECOUNT(1)	
100	Return Dead Sector Count	Return Ready Sector Count

1. Indicates sector erase count

Upon clearing CCIF to launch the EEPROM Emulation Query command, the CCIF flag will set after the EEE partition and status variables are stored in the FCCOBIX register.If the Emulation Query command is executed prior to partitioning (Partition D-Flash Command Section 29.4.2.15), the following reset values are returned: DFPART = 0x_FFFF, ERPART = 0x_FFFF, ECOUNT = 0x_FFFF, Dead Sector Count = 0x_00, Ready Sector Count = 0x_00.

5. Maximum partitioning

A.3.2 NVM Reliability Parameters

The reliability of the NVM blocks is guaranteed by stress test during qualification, constant process monitors and burn-in to screen early life failures.

The data retention and program/erase cycling failure rates are specified at the operating conditions noted. The program/erase cycle count on the sector is incremented every time a sector or mass erase event is executed.

0x0300–0x0327 Pulse Width Modulator 8-Bit 8-Channel (PWM) Map (Sheet 3 of 3)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0321	PWMDTY5	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0322	PWMDTY6	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0323	PWMDTY7	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x0324	PWMSDN	R	PWMIF	PWMIE	0	PWMLVL	0	PWM7IN	PWM7INL	PWM7 ENA
		W			PWM RSTRT					
0x0325	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0326	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0327	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x0328–0x032F Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0328– 0x032F	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x00330–0x0337 Asynchronous Serial Interface (SCI6) Map

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0330	SCI6BDH[(1)]	R	IREN	TNP1	TNP0	SBR12	SBR11	SBR10	SBR9	SBR8
		W								
0x0331	SCI6BDL[1]	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
		W								
0x0332	SCI6CR1[1]	R	LOOPS	SCISWAI	RSRC	M	WAKE	ILT	PE	PT
		W								
0x0330	SCI6ASR1[(2)]	R	RXEDGIF	0	0	0	0	BERRV	BERRIF	BKDIF
		W								
0x0331	SCI6ACR1[2]	R	RXEDGIE	0	0	0	0	0	BERRIE	BKDIE
		W								
0x0332	SCI6ACR2[2]	R	0	0	0	0	0	BERRM1	BERRM0	BKDFE
		W								
0x0333	SCI6CR2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
		W								
0x0334	SCI6SR1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
		W								