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	Obsolete
Core Processor	HCS12X
Core Size	16-Bit
Speed	80MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	59
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	20K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12xdt512vaa

Appendix F
Ordering Information

Appendix G
Detailed Register Map

be configured as master output (during master mode) or slave input pin (during slave mode) MOSI of the serial peripheral interface 2 (SPI2).

1.2.3.55 PP4 / KWP4 / PWM4 / MISO2 — Port P I/O Pin 4

PP4 is a general-purpose input or output pin. It can be configured to generate an interrupt causing the MCU to exit stop or wait mode. It can be configured as pulse width modulator (PWM) channel 4 output. It can be configured as master input (during master mode) or slave output (during slave mode) pin MISO of the serial peripheral interface 2 (SPI2).

1.2.3.56 PP3 / KWP3 / PWM3 / $\overline{SS1}$ — Port P I/O Pin 3

PP3 is a general-purpose input or output pin. It can be configured to generate an interrupt causing the MCU to exit stop or wait mode. It can be configured as pulse width modulator (PWM) channel 3 output. It can be configured as slave select pin $\overline{SS}$ of the serial peripheral interface 1 (SPI1).

1.2.3.57 PP2 / KWP2 / PWM2 / SCK1 — Port P I/O Pin 2

PP2 is a general-purpose input or output pin. It can be configured to generate an interrupt causing the MCU to exit stop or wait mode. It can be configured as pulse width modulator (PWM) channel 2 output. It can be configured as serial clock pin SCK of the serial peripheral interface 1 (SPI1).

1.2.3.58 PP1 / KWP1 / PWM1 / MOSI1 — Port P I/O Pin 1

PP1 is a general-purpose input or output pin. It can be configured to generate an interrupt causing the MCU to exit stop or wait mode. It can be configured as pulse width modulator (PWM) channel 1 output. It can be configured as master output (during master mode) or slave input pin (during slave mode) MOSI of the serial peripheral interface 1 (SPI1).

1.2.3.59 PP0 / KWP0 / PWM0 / MISO1 — Port P I/O Pin 0

PP0 is a general-purpose input or output pin. It can be configured to generate an interrupt causing the MCU to exit stop or wait mode. It can be configured as pulse width modulator (PWM) channel 0 output. It can be configured as master input (during master mode) or slave output (during slave mode) pin MISO of the serial peripheral interface 1 (SPI1).

1.2.3.60 PS7 / $\overline{SS0}$ — Port S I/O Pin 7

PS7 is a general-purpose input or output pin. It can be configured as the slave select pin $\overline{SS}$ of the serial peripheral interface 0 (SPI0).

1.2.3.61 PS6 / SCK0 — Port S I/O Pin 6

PS6 is a general-purpose input or output pin. It can be configured as the serial clock pin SCK of the serial peripheral interface 0 (SPI0).

4.3.2 Register Descriptions

This section describes in address order all the ATD10B16C registers and their individual bits.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 ATDCTL0	R	0	0	0	0	WRAP3	WRAP2	WRAP1	WRAP0
	W								
0x0001 ATDCTL1	R	ETRIGSEL	0	0	0	ETRIGCH3	ETRIGCH2	ETRIGCH1	ETRIGCH0
	W								
0x0002 ATDCTL2	R	ADPU	AFFC	AWAI	ETRIGLE	ETRIGP	ETRIGE	ASCIE	ASCIF
	W								
0x0003 ATDCTL3	R	0	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
	W								
0x0004 ATDCTL4	R	SRES8	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0
	W								
0x0005 ATDCTL5	R	DJM	DSGN	SCAN	MULT	CD	CC	CB	CA
	W								
0x0006 ATDSTAT0	R	SCF	0	ETORF	FIFOR	CC3	CC2	CC1	CC0
	W								
0x0007 Unimplemented	R								
	W								
0x0008 ATDTEST0	R	Unimplemented							
	W								
0x0009 ATDTEST1	R	Unimplemented							SC
	W								
0x000A ATDSTAT2	R	CCF15	CCF14	CCF13	CCF12	CCF11	CCF10	CCF9	CCF8
	W								
0x000B ATDSTAT1	R	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
	W								
0x000C ATDDIEN0	R	IEN15	IEN14	IEN13	IEN12	IEN11	IEN10	IEN9	IEN8
	W								

= Unimplemented or Reserved u = Unaffected

Figure 4-2. ATD Register Summary

Table 4-7. External Trigger Configurations

ETRIGLE	ETRIGP	External Trigger Sensitivity
0	0	Falling Edge
0	1	Ring Edge
1	0	Low Level
1	1	High Level

Table 5-12. Clock Prescaler Values

Prescale Value	Total Divisor Value	Max. Bus Clock[1]	Min. Bus Clock[2]
00000	Divide by 2	4 MHz	1 MHz
00001	Divide by 4	8 MHz	2 MHz
00010	Divide by 6	12 MHz	3 MHz
00011	Divide by 8	16 MHz	4 MHz
00100	Divide by 10	20 MHz	5 MHz
00101	Divide by 12	24 MHz	6 MHz
00110	Divide by 14	28 MHz	7 MHz
00111	Divide by 16	32 MHz	8 MHz
01000	Divide by 18	36 MHz	9 MHz
01001	Divide by 20	40 MHz	10 MHz
01010	Divide by 22	44 MHz	11 MHz
01011	Divide by 24	48 MHz	12 MHz
01100	Divide by 26	52 MHz	13 MHz
01101	Divide by 28	56 MHz	14 MHz
01110	Divide by 30	60 MHz	15 MHz
01111	Divide by 32	64 MHz	16 MHz
10000	Divide by 34	68 MHz	17 MHz
10001	Divide by 36	72 MHz	18 MHz
10010	Divide by 38	76 MHz	19 MHz
10011	Divide by 40	80 MHz	20 MHz
10100	Divide by 42	84 MHz	21 MHz
10101	Divide by 44	88 MHz	22 MHz
10110	Divide by 46	92 MHz	23 MHz
10111	Divide by 48	96 MHz	24 MHz
11000	Divide by 50	100 MHz	25 MHz
11001	Divide by 52	104 MHz	26 MHz
11010	Divide by 54	108 MHz	27 MHz
11011	Divide by 56	112 MHz	28 MHz
11100	Divide by 58	116 MHz	29 MHz
11101	Divide by 60	120 MHz	30 MHz
11110	Divide by 62	124 MHz	31 MHz
11111	Divide by 64	128 MHz	32 MHz

1 Maximum ATD conversion clock frequency is 2 MHz. The maximum allowed bus clock frequency is shown in this column.

2 Minimum ATD conversion clock frequency is 500 kHz. The minimum allowed bus clock frequency is shown in this column.

ADDH — Add Immediate 8 bit Constant (High Byte) — ADDH

Operation

RD + IMM8:$00 ⇒ RD

Adds the content of high byte of register RD and a signed immediate 8 bit constant using binary addition and stores the result in the high byte of the destination register RD. This instruction can be used after an ADDL for a 16 bit immediate addition.

Example:

```
ADDL    R2,#LOWBYTE
ADDH    R2,#HIGHBYTE      ; R2 = R2 + 16 bit immediate
```

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 operation; cleared otherwise.
$RD[15]_{old}$ & IMM8[7] & $\overline{RD[15]}_{new}$ | $\overline{RD[15]}_{old}$ & $\overline{IMM8[7]}$ & $RD[15]_{new}$

C: Set if there is a carry from the bit 15 of the result; cleared otherwise.
$RD[15]_{old}$ & IMM8[7] | $RD[15]_{old}$ & $\overline{RD[15]}_{new}$ | IMM8[7] & $\overline{RD[15]}_{new}$

Code and CPU Cycles

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

BFINSX Bit Field Insert and XNOR BFINSX

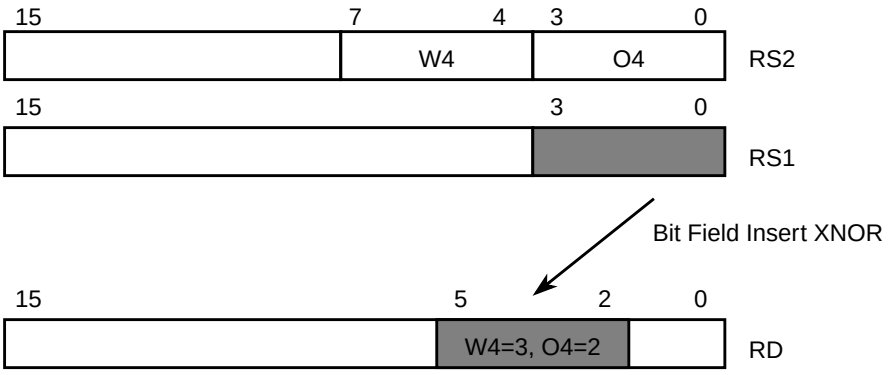
Operation

!(RS1[*w*:*0*] ^ RD[w+o:o]) ⇒ RD[*w*+o:o];

w = (RS2[7:4])

o = (RS2[3:0])

Extracts *w+1* bits from register RS1 starting at position 0, performs an XNOR with RD[w+o:o] and writes the bits back io RD. The remaining bits in RD are not affected. If (o+w) > 15 the upper bits are ignored. Using R0 as a RS1, this command can be used to toggle bits.



CCR Effects

N	Z	V	C
Δ	Δ	0	—

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

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

V: 0; cleared.

C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code										Cycles
BFINSX RD, RS1, RS2	TRI	0	1	1	1	1	RD	RS1	RS2	1	1	P

XNORL — Logical Exclusive NOR Immediate 8 bit Constant (Low Byte) — XNORL

Operation

~(RD.L ^ IMM8) ⇒ RD.L

Performs a bit wise logical exclusive NOR between the low byte of register RD and an immediate 8 bit constant and stores the result in the destination register RD.L. The high byte of RD is not affected.

CCR Effects

N	Z	V	C
Δ	Δ	0	—

N: Set if bit 7 of the result is set; cleared otherwise.
Z: Set if the 8 bit result is $00; cleared otherwise.
V: 0; cleared.
C: Not affected.

Code and CPU Cycles

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

7.3.2.2 Timer Compare Force Register (CFORC)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	FOC7	FOC6	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0
Reset	0	0	0	0	0	0	0	0

Figure 7-4. Timer Compare Force Register (CFORC)

Read or write: Anytime but reads will always return 0x0000 (1 state is transient).

All bits reset to zero.

Table 7-3. CFORC Field Descriptions

Field	Description
7:0 FOC[7:0]	**Force Output Compare Action for Channel 7:0** — A write to this register with the corresponding data bit(s) set causes the action which is programmed for output compare “x” to occur immediately. The action taken is the same as if a successful comparison had just taken place with the TCx register except the interrupt flag does not get set. **Note:** A successful channel 7 output compare overrides any channel 6:0 compares. If a forced output compare on any channel occurs at the same time as the successful output compare, then the forced output compare action will take precedence and the interrupt flag will not get set.

7.3.2.3 Output Compare 7 Mask Register (OC7M)

	7	6	5	4	3	2	1	0
R W	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0
Reset	0	0	0	0	0	0	0	0

Figure 7-5. Output Compare 7 Mask Register (OC7M)

Read or write: Anytime

All bits reset to zero.

Table 7-4. OC7M Field Descriptions

Field	Description
7:0 OC7M[7:0]	**Output Compare Mask Action for Channel 7:0** 0 The corresponding OC7Dx bit in the output compare 7 data register will not be transferred to the timer port on a successful channel 7 output compare, even if the corresponding pin is setup for output compare. 1 The corresponding OC7Dx bit in the output compare 7 data register will be transferred to the timer port on a successful channel 7 output compare. **Note:** The corresponding channel must also be setup for output compare (IOSx = 1) for data to be transferred from the output compare 7 data register to the timer port.

Table 7-23. Modulus Counter Prescaler Select

MCPR1	MCPR0	Prescaler Division
0	0	1
0	1	4
1	0	8
1	1	16

7.3.2.20 16-Bit Modulus Down-Counter FLAG Register (MCFLG)

	7	6	5	4	3	2	1	0
R	MCZF	0	0	0	POLF3	POLF2	POLF1	POLF0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-42. 16-Bit Modulus Down-Counter FLAG Register (MCFLG)

Read: Anytime

Write only used in the flag clearing mechanism for bit 7. Writing a one to bit 7 clears the flag. Writing a zero will not affect the current status of the bit.

NOTE

When TFFCA = 1, the flag cannot be cleared via the normal flag clearing mechanism (writing a one to the flag). Reference Section 7.3.2.6, “Timer System Control Register 1 (TSCR1)”.

All bits reset to zero.

Table 7-24. MCFLG Field Descriptions

Field	Description
7 MCZF	**Modulus Counter Underflow Flag** — The flag is set when the modulus down-counter reaches 0x0000. The flag indicates when interrupt conditions have occurred. The flag can be cleared via the normal flag clearing mechanism (writing a one to the flag) or via the fast flag clearing mechanism (Reference TFFCA bit in Section 7.3.2.6, “Timer System Control Register 1 (TSCR1)”).
3:0 POLF[3:0]	**First Input Capture Polarity Status** — These are read only bits. Writes to these bits have no effect. Each status bit gives the polarity of the first edge which has caused an input capture to occur after capture latch has been read. Each POLFx corresponds to a timer PORTx input. 0 The first input capture has been caused by a falling edge. 1 The first input capture has been caused by a rising edge.

7.4.1 Enhanced Capture Timer Modes of Operation

The enhanced capture timer has 8 input capture, output compare (IC/OC) channels, same as on the HC12 standard timer (timer channels TC0 to TC7). When channels are selected as input capture by selecting the IOSx bit in TIOS register, they are called input capture (IC) channels.

Four IC channels (channels 7–4) are the same as on the standard timer with one capture register each that memorizes the timer value captured by an action on the associated input pin.

Four other IC channels (channels 3–0), in addition to the capture register, also have one buffer each called a holding register. This allows two different timer values to be saved without generating any interrupts.

Four 8-bit pulse accumulators are associated with the four buffered IC channels (channels 3–0). Each pulse accumulator has a holding register to memorize their value by an action on its external input. Each pair of pulse accumulators can be used as a 16-bit pulse accumulator.

The 16-bit modulus down-counter can control the transfer of the IC registers and the pulse accumulators contents to the respective holding registers for a given period, every time the count reaches zero.

The modulus down-counter can also be used as a stand-alone time base with periodic interrupt capability.

7.4.1.1 IC Channels

The IC channels are composed of four standard IC registers and four buffered IC channels.

- An IC register is empty when it has been read or latched into the holding register.
- A holding register is empty when it has been read.

7.4.1.1.1 Non-Buffered IC Channels

The main timer value is memorized in the IC register by a valid input pin transition. If the corresponding NOVWx bit of the ICOVW register is cleared, with a new occurrence of a capture, the contents of IC register are overwritten by the new value. If the corresponding NOVWx bit of the ICOVW register is set, the capture register cannot be written unless it is empty. This will prevent the captured value from being overwritten until it is read.

7.4.1.1.2 Buffered IC Channels

There are two modes of operations for the buffered IC channels:

1. IC latch mode (LATQ = 1)

   The main timer value is memorized in the IC register by a valid input pin transition (see Figure 7-65 and Figure 7-66).

   The value of the buffered IC register is latched to its holding register by the modulus counter for a given period when the count reaches zero, by a write 0x0000 to the modulus counter or by a write to ICLAT in the MCCTL register.

   If the corresponding NOVWx bit of the ICOVW register is cleared, with a new occurrence of a capture, the contents of IC register are overwritten by the new value. In case of latching, the contents of its holding register are overwritten.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
PWMPER7	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMDTY0	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMDTY1	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMDTY2	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMDTY3	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMDTY4	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMDTY5	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMDTY6	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMDTY7	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMSDN	R W	PWMIF	PWMIE	0 PWMRSTRT	PWMLVL	0	PWM7IN	PWM7INL	PWM7ENA

= Unimplemented or Reserved

Figure 8-2. PWM Register Summary (Sheet 3 of 3)

1 Intended for factory test purposes only.

8.3.2.1 PWM Enable Register (PWME)

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx = 1), the associated PWM output is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle due to the synchronization of PWMEx and the clock source.

NOTE

The first PWM cycle after enabling the channel can be irregular.

An exception to this is when channels are concatenated. Once concatenated mode is enabled (CONxx bits set in PWMCTL register), enabling/disabling the corresponding 16-bit PWM channel is controlled by the

Table 10-1. CANCTL0 Register Field Descriptions (continued)

Field	Description
1 SLPRQ[5]	**Sleep Mode Request** — This bit requests the MSCAN to enter sleep mode, which is an internal power saving mode (see Section 10.4.5.4, "MSCAN Sleep Mode"). The sleep mode request is serviced when the CAN bus is idle, i.e., the module is not receiving a message and all transmit buffers are empty. The module indicates entry to sleep mode by setting SLPAK = 1 (see Section 10.3.2.2, "MSCAN Control Register 1 (CANCTL1)"). SLPRQ cannot be set while the WUPIF flag is set (see Section 10.3.2.5, "MSCAN Receiver Flag Register (CANRFLG)"). Sleep mode will be active until SLPRQ is cleared by the CPU or, depending on the setting of WUPE, the MSCAN detects activity on the CAN bus and clears SLPRQ itself. 0 Running — The MSCAN functions normally 1 Sleep mode request — The MSCAN enters sleep mode when CAN bus idle
0 INITRQ[6,7]	**Initialization Mode Request** — When this bit is set by the CPU, the MSCAN skips to initialization mode (see Section 10.4.5.5, "MSCAN Initialization Mode"). Any ongoing transmission or reception is aborted and synchronization to the CAN bus is lost. The module indicates entry to initialization mode by setting INITAK = 1 (Section 10.3.2.2, "MSCAN Control Register 1 (CANCTL1)"). The following registers enter their hard reset state and restore their default values: CANCTL0[8], CANRFLG[9], CANRIER[10], CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL. The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0-7, and CANIDMR0-7 can only be written by the CPU when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1). The values of the error counters are not affected by initialization mode. When this bit is cleared by the CPU, the MSCAN restarts and then tries to synchronize to the CAN bus. If the MSCAN is not in bus-off state, it synchronizes after 11 consecutive recessive bits on the CAN bus; if the MSCAN is in bus-off state, it continues to wait for 128 occurrences of 11 consecutive recessive bits. Writing to other bits in CANCTL0, CANRFLG, CANRIER, CANTFLG, or CANTIER must be done only after initialization mode is exited, which is INITRQ = 0 and INITAK = 0. 0 Normal operation 1 MSCAN in initialization mode

1 The MSCAN must be in normal mode for this bit to become set.

2 See the Bosch CAN 2.0A/B specification for a detailed definition of transmitter and receiver states.

3 In order to protect from accidentally violating the CAN protocol, the TXCAN pin is immediately forced to a recessive state when the CPU enters wait (CSWAI = 1) or stop mode (see Section 10.4.5.2, "Operation in Wait Mode" and Section 10.4.5.3, "Operation in Stop Mode").

4 The CPU has to make sure that the WUPE register and the WUPIE wake-up interrupt enable register (see Section 10.3.2.6, "MSCAN Receiver Interrupt Enable Register (CANRIER)) is enabled, if the recovery mechanism from stop or wait is required.

5 The CPU cannot clear SLPRQ before the MSCAN has entered sleep mode (SLPRQ = 1 and SLPAK = 1).

6 The CPU cannot clear INITRQ before the MSCAN has entered initialization mode (INITRQ = 1 and INITAK = 1).

7 In order to protect from accidentally violating the CAN protocol, the TXCAN pin is immediately forced to a recessive state when the initialization mode is requested by the CPU. Thus, the recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before requesting initialization mode.

8 Not including WUPE, INITRQ, and SLPRQ.

9 TSTAT1 and TSTAT0 are not affected by initialization mode.

10 RSTAT1 and RSTAT0 are not affected by initialization mode.

10.3.2.2 MSCAN Control Register 1 (CANCTL1)

The CANCTL1 register provides various control bits and handshake status information of the MSCAN module as described below.

- If there are one or more message buffers scheduled for transmission (TXEx = 0), the MSCAN will continue to transmit until all transmit message buffers are empty (TXEx = 1, transmitted successfully or aborted) and then goes into sleep mode.
- If the MSCAN is receiving, it continues to receive and goes into sleep mode as soon as the CAN bus next becomes idle.
- If the MSCAN is neither transmitting nor receiving, it immediately goes into sleep mode.

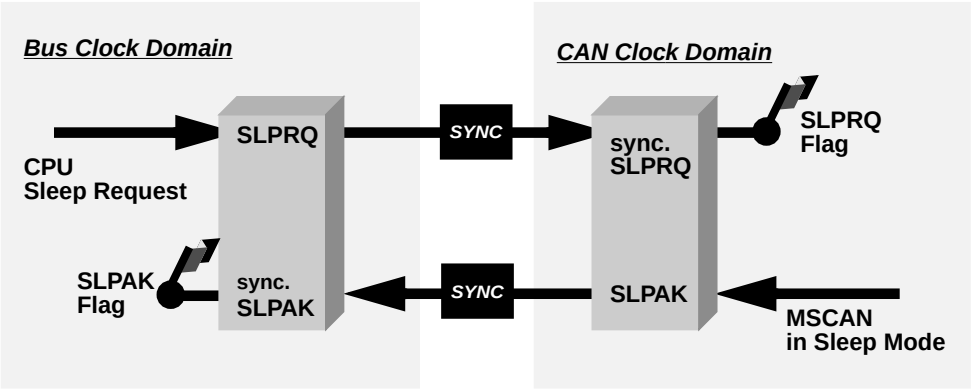


Figure 10-45. Sleep Request / Acknowledge Cycle

NOTE

The application software must avoid setting up a transmission (by clearing one or more TXEx flag(s)) and immediately request sleep mode (by setting SLPRQ). Whether the MSCAN starts transmitting or goes into sleep mode directly depends on the exact sequence of operations.

If sleep mode is active, the SLPRQ and SLPAK bits are set (Figure 10-45). The application software must use SLPAK as a handshake indication for the request (SLPRQ) to go into sleep mode.

When in sleep mode (SLPRQ = 1 and SLPAK = 1), the MSCAN stops its internal clocks. However, clocks that allow register accesses from the CPU side continue to run.

If the MSCAN is in bus-off state, it stops counting the 128 occurrences of 11 consecutive recessive bits due to the stopped clocks. The TXCAN pin remains in a recessive state. If RXF = 1, the message can be read and RXF can be cleared. Shifting a new message into the foreground buffer of the receiver FIFO (RxFG) does not take place while in sleep mode.

It is possible to access the transmit buffers and to clear the associated TXE flags. No message abort takes place while in sleep mode.

If the WUPE bit in CANCTL0 is not asserted, the MSCAN will mask any activity it detects on CAN. The RXCAN pin is therefore held internally in a recessive state. This locks the MSCAN in sleep mode (Figure 10-46). WUPE must be set before entering sleep mode to take effect.

The MSCAN is able to leave sleep mode (wake up) only when:

- CAN bus activity occurs and WUPE = 1

20.4.5.3.1 Information Byte Organization

The format of the control information byte for both S12XCPU and XGATE modules is dependent upon the active trace mode and tracing source as described below. In Normal, Loop1, or Pure PC modes tracing of XGATE activity, XINF is used to store control information. In Normal, Loop1, or Pure PC modes tracing of S12XCPU activity, CINF is used to store control information. In Detail Mode, CXINF contains the control information

XGATE Information Byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XSD	XSOT	XCOT	XDV	0	0	0	0

Figure 20-24. XGATE Information Byte XINF

Table 20-40. XINF Field Descriptions

Field	Description
7 XSD	**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 or Start of Thread or Continuation of Thread
6 XSOT	**Source Of Thread Indicator** — This bit indicates that the corresponding stored address is a start of thread address. This is only used in Normal and Loop1 mode tracing. **NOTE. This bit only has effect on devices where the XGATE module supports multiple interrupt levels**. 0 Stored address not from a start of thread 1 Stored address from a start of thread
5 XCOT	**Continuation Of Thread Indicator** — This bit indicates that the corresponding stored address is the first address following a return from a higher priority thread. This is only used in Normal and Loop1 mode tracing. **NOTE. This bit only has effect on devices where the XGATE module supports multiple interrupt levels.** 0 Stored address not from a continuation of thread 1 Stored address from a continuation of thread
4 XDV	**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 XGATE trace buffer entry is valid. 0 Trace buffer entry is invalid 1 Trace buffer entry is valid

X12X_CPU 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 20-25. S12XCPU Information Byte CINF

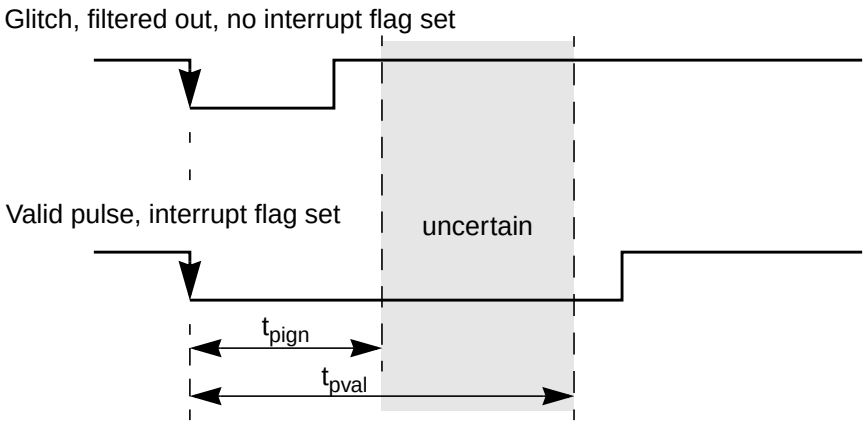


Figure 23-77. Interrupt Glitch Filter on Port P, H, and J (PPS = 0)

Table 23-69. Pulse Detection Criteria

Pulse	Mode		
	$\overline{STOP}$	Unit	STOP[1]
Ignored	$t_{pulse} \leq 3$	Bus clocks	$t_{pulse} \leq t_{pign}$
Uncertain	$3 < t_{pulse} < 4$	Bus clocks	$t_{pign} < t_{pulse} < t_{pval}$
Valid	$t_{pulse} \geq 4$	Bus clocks	$t_{pulse} \geq t_{pval}$

1. These values include the spread of the oscillator frequency over temperature, voltage and process.

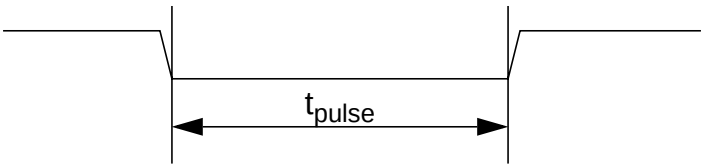


Figure 23-78. Pulse Illustration

A valid edge on an input is detected if 4 consecutive samples of a passive level are followed by 4 consecutive samples of an active level directly or indirectly.

The filters are continuously clocked by the bus clock in run and wait mode. In stop mode, the clock is generated by an RC-oscillator in the port integration module. To maximize current saving the RC oscillator runs only if the following condition is true on any pin individually:

Sample count <= 4 and interrupt enabled (PIE = 1) and interrupt flag not set (PIF = 0).

23.0.9 Expanded Bus Pin Functions

All peripheral ports T, S, M, P, H, J, AD0, and AD1 start up as general purpose inputs after reset.

Depending on the external mode pin condition, the external bus interface related ports A, B, C, D, E, and K start up as general purpose inputs on reset or are configured for their alternate functions.

Write: Anytime.

This register configures whether a pull-up or a pull-down device is activated, if the port is used as input or as output in wired-OR (open drain) mode. This bit has no effect if the port is used as push-pull output. Out of reset a pull-up device is enabled.

Table 24-25. PERS Field Descriptions

Field	Description
7–0 PERS[7:0]	**Pull Device Enable Port S** 0 Pull-up or pull-down device is disabled. 1 Either a pull-up or pull-down device is enabled.

24.0.5.24 Port S Polarity Select Register (PPSS)

	7	6	5	4	3	2	1	0
R W	PPSS7	PPSS6	PPSS5	PPSS4	PPSS3	PPSS2	PPSS1	PPSS0
Reset	0	0	0	0	0	0	0	0

Figure 24-26. Port S Polarity Select Register (PPSS)

Read: Anytime.

Write: Anytime.

This register selects whether a pull-down or a pull-up device is connected to the pin.

Table 24-26. PPSS Field Descriptions

Field	Description
7–0 PPSS[7:0]	**Pull Select Port S** 0 A pull-up device is connected to the associated port S pin, if enabled by the associated bit in register PERS and if the port is used as input or as wired-OR output. 1 A pull-down device is connected to the associated port S pin, if enabled by the associated bit in register PERS and if the port is used as input.

24.0.5.25 Port S Wired-OR Mode Register (WOMS)

	7	6	5	4	3	2	1	0
R W	WOMS7	WOMS6	WOMS5	WOMS4	WOMS3	WOMS2	WOMS1	WOMS0
Reset	0	0	0	0	0	0	0	0

Figure 24-27. Port S Wired-OR Mode Register (WOMS)

Read: Anytime.

Write: Anytime.

28.4.2.6 Sector Erase Abort Command

The sector erase abort operation will terminate the active sector erase operation so that other sectors in a Flash block are available for read and program operations without waiting for the sector erase operation to complete.

An example flow to execute the sector erase abort operation is shown in Figure 28-31. The sector erase abort command write sequence is as follows:

1. Write to any Flash block address to start the command write sequence for the sector erase abort command. The address and data written are ignored.
2. Write the sector erase abort command, 0x47, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the sector erase abort command.

If the sector erase abort command is launched resulting in the early termination of an active sector erase operation, the ACCERR flag will set once the operation completes as indicated by the CCIF flag being set. The ACCERR flag sets to inform the user that the Flash sector may not be fully erased and a new sector erase command must be launched before programming any location in that specific sector. If the sector erase abort command is launched but the active sector erase operation completes normally, the ACCERR flag will not set upon completion of the operation as indicated by the CCIF flag being set. Therefore, if the ACCERR flag is not set after the sector erase abort command has completed, a Flash sector being erased when the abort command was launched will be fully erased. The maximum number of cycles required to abort a sector erase operation is equal to four FCLK periods (see Section 28.4.1.1, “Writing the FCLKDIV Register”) plus five bus cycles as measured from the time the CBEIF flag is cleared until the CCIF flag is set. If sectors in multiple Flash blocks are being simultaneously erased, the sector erase abort operation will be applied to all active Flash blocks without writing to each Flash block in the sector erase abort command write sequence.

NOTE

Since the ACCERR bit in the FSTAT register may be set at the completion of the sector erase abort operation, a command write sequence is not allowed to be buffered behind a sector erase abort command write sequence. The CBEIF flag will not set after launching the sector erase abort command to indicate that a command should not be buffered behind it. If an attempt is made to start a new command write sequence with a sector erase abort operation active, the ACCERR flag in the FSTAT register will be set. A new command write sequence may be started after clearing the ACCERR flag, if set.

NOTE

The sector erase abort command should be used sparingly since a sector erase operation that is aborted counts as a complete program/erase cycle.

E.4 MC9S12XD/A/B -Family SRAM & EEPROM Configuration

Figure E-2. Available RAM Pages on S12XD-Family[1]

<table>
<tr><th>RAM Page
RP[7:0]</th><th>DP512
A512</th><th>DT512
DT384</th><th>DQ256
A256</th><th>DG128
A128</th><th>D128</th><th>D64</th></tr>
<tr><td>0xF6</td><td rowspan="2"></td><td rowspan="5"></td><td rowspan="8"></td><td rowspan="10"></td><td rowspan="9"></td><td rowspan="8"></td></tr>
<tr><td>0xF7</td></tr>
<tr><td>0xF8</td><td rowspan="8">32K Byte</td></tr>
<tr><td>0xF9</td></tr>
<tr><td>0xFA</td></tr>
<tr><td>0xFB</td><td rowspan="6">20K Byte</td></tr>
<tr><td>0xFC</td><td rowspan="5">16K Byte</td></tr>
<tr><td>0xFD</td></tr>
<tr><td>0xFE</td><td rowspan="2">12K Byte</td><td rowspan="2">8K Byte</td></tr>
<tr><td>0xFF</td><td>4K Byte</td></tr>
</table>

[1] On 9S12XD256 14K byte RAM available pages FF,FE,FD and upper half of page FC
On 9S12XB256 10K byte RAM available pages FF, FE upper half of page FD
On 9S12XB218 6K byte RAM available pages FF and upper half of page FE

Table E-4. Available EEPROM Pages on MC9S12XD-Family

<table>
<tr><th>EEPROM
Page
EP[7:0]</th><th>DP512
DT512
DT384
DQ256
A256
A512</th><th>DG128
D128
A128
B256</th><th>D64
B128</th></tr>
<tr><td>0xFA</td><td rowspan="2"></td><td rowspan="4"></td><td rowspan="5"></td></tr>
<tr><td>0xFB</td></tr>
<tr><td>0xFC</td><td rowspan="4">4K Byte</td></tr>
<tr><td>0xFD</td></tr>
<tr><td>0xFE</td><td rowspan="2">2K Byte</td></tr>
<tr><td>0xFF</td><td>1K Byte</td></tr>
</table>

0x0180–0x01BF Freescale Scalable CAN — MSCAN (CAN1) Map (Sheet 3 of 3)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x019F	CAN1IDMR7	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		W								
0x01A0–0x01AF	CAN1RXFG	R	FOREGROUND RECEIVE BUFFER (See Detailed MSCAN Foreground Receive and Transmit Buffer Layout)							
		W								
0x01B0–0x01BF	CAN1TXFG	R	FOREGROUND TRANSMIT BUFFER (See Detailed MSCAN Foreground Receive and Transmit Buffer Layout)							
		W								

0x01C0–0x01FF Freescale Scalable CAN — MSCAN (CAN2) Map

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x01C0	CAN2CTL0	R	RXFRM	RXACT	CSWAI	SYNCH	TIME	WUPE	SLPRQ	INITRQ
		W								
0x01C1	CAN2CTL1	R	CANE	CLKSRC	LOOPB	LISTEN	BORM	WUPM	SLPAK	INITAK
		W								
0x01C2	CAN2BTR0	R	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
		W								
0x01C3	CAN2BTR1	R	SAMP	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
		W								
0x01C4	CAN2RFLG	R	WUPIF	CSCIF	RSTAT1	RSTAT0	TSTAT1	TSTAT0	OVRIF	RXF
		W								
0x01C5	CAN2RIER	R	WUPIE	CSCIE	RSTATE1	RSTATE0	TSTATE1	TSTATE0	OVRIE	RXFIE
		W								
0x01C6	CAN2TFLG	R	0	0	0	0	0	TXE2	TXE1	TXE0
		W								
0x01C7	CAN2TIER	R	0	0	0	0	0	TXEIE2	TXEIE1	TXEIE0
		W								
0x01C8	CAN2TARQ	R	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
		W								
0x01C9	CAN2TAAK	R	0	0	0	0	0	ABTAK2	ABTAK1	ABTAK0
		W								
0x01CA	CAN2TBSEL	R	0	0	0	0	0	TX2	TX1	TX0
		W								
0x01CB	CAN2IDAC	R	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
		W								
0x01CC	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x01CD	CAN2MISC	R	0	0	0	0	0	0	0	BOHOLD
		W								
0x01CE	CAN2RXERR	R	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
		W								
0x01CF	CAN2TXERR	R	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
		W								
0x01D0	CAN2IDAR0	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		W								