



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	40MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	44
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12xs64caer

3.3 Memory Map and Registers

3.3.1 Module Memory Map

A summary of the registers associated with the MMC block is shown in Figure 3-2. Detailed descriptions of the registers and bits are given in the subsections that follow.

Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000A	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x000B	MODE	R	MODC	0	0	0	0	0	0	0
		W								
0x0010	GPAGE	R	0							
		W		GP6	GP5	GP4	GP3	GP2	GP1	GP0
0x0011	DIRECT	R								
		W	DP15	DP14	DP13	DP12	DP11	DP10	DP9	DP8
0x0012	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0013	MMCCTL1	R		0				0	0	0
		W	MGRAMON		DFIFRON	PGMIFRON				
0x0014	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0015	PPAGE	R								
		W	PIX7	PIX6	PIX5	PIX4	PIX3	PIX2	PIX1	PIX0
0x0016	RPAGE	R								
		W	RP7	RP6	RP5	RP4	RP3	RP2	RP1	RP0
0x0017	EPAGE	R								
		W	EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0

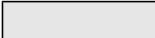
 = Unimplemented or Reserved

Figure 3-2. MMC Register Summary

3.3.2 Register Descriptions

Global Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x7FFF07	BDMCCRH	R	0	0	0	0	0	CCR10	CCR9	CCR8
		W								
0x7FFF08	BDMGPR	R	BGAE	BGP6	BGP5	BGP4	BGP3	BGP2	BGP1	BGP0
		W								
0x7FFF09	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x7FFF0A	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x7FFF0B	Reserved	R	0	0	0	0	0	0	0	0
		W								

= Unimplemented, Reserved
 = Implemented (do not alter)

X = Indeterminate
 0 = Always read zero

Figure 5-2. BDM Register Summary (continued)

5.3.2.1 BDM Status Register (BDMSTS)

Register Global Address 0x7FFF01

	7	6	5	4	3	2	1	0
R	ENBDM	BDMACT	0	SDV	TRACE	CLKSW	UNSEC	0
W								
Reset								
Special Single-Chip Mode	0 ¹	1	0	0	0	0	0 ³	0
Emulation Modes (if modes available)	1	0	0	0	0	1 ²	0	0
All Other Modes	0	0	0	0	0	0	0	0

= Unimplemented, Reserved
 = Implemented (do not alter)

0 = Always read zero

- ¹ ENBDM is read as 1 by a debugging environment in special single chip mode when the device is not secured or secured but fully erased (non-volatile memory). This is because the ENBDM bit is set by the standard firmware before a BDM command can be fully transmitted and executed.
- ² CLKSW is read as 1 by a debugging environment in emulation modes when the device is not secured and read as 0 when secured if emulation modes available.
- ³ UNSEC is read as 1 by a debugging environment in special single chip mode when the device is secured and fully erased, else it is 0 and can only be read if not secure (see also bit description).

Figure 5-3. BDM Status Register (BDMSTS)

Table 5-7. Firmware Commands

Command ¹	Opcode (hex)	Data	Description
READ_NEXT ²	62	16-bit data out	Increment X index register by 2 ($X = X + 2$), then read word X points to.
READ_PC	63	16-bit data out	Read program counter.
READ_D	64	16-bit data out	Read D accumulator.
READ_X	65	16-bit data out	Read X index register.
READ_Y	66	16-bit data out	Read Y index register.
READ_SP	67	16-bit data out	Read stack pointer.
WRITE_NEXT<f-helvetica><st-superscript>	42	16-bit data in	Increment X index register by 2 ($X = X + 2$), then write word to location pointed to by X.
WRITE_PC	43	16-bit data in	Write program counter.
WRITE_D	44	16-bit data in	Write D accumulator.
WRITE_X	45	16-bit data in	Write X index register.
WRITE_Y	46	16-bit data in	Write Y index register.
WRITE_SP	47	16-bit data in	Write stack pointer.
GO	08	none	Go to user program. If enabled, ACK will occur when leaving active background mode.
GO_UNTIL ³	0C	none	Go to user program. If enabled, ACK will occur upon returning to active background mode.
TRACE1	10	none	Execute one user instruction then return to active BDM. If enabled, ACK will occur upon returning to active background mode.
TAGGO -> GO	18	none	(Previous enable tagging and go to user program.) This command will be deprecated and should not be used anymore. Opcode will be executed as a GO command.

¹ If enabled, ACK will occur when data is ready for transmission for all BDM READ commands and will occur after the write is complete for all BDM WRITE commands.

² When the firmware command READ_NEXT or WRITE_NEXT is used to access the BDM address space the BDM resources are accessed rather than user code. Writing BDM firmware is not possible.

³ System stop disables the ACK function and ignored commands will not have an ACK-pulse (e.g., CPU in stop or wait mode). The GO_UNTIL command will not get an Acknowledge if CPU executes the wait or stop instruction before the “UNTIL” condition (BDM active again) is reached (see Section 5.4.7, “Serial Interface Hardware Handshake Protocol” last Note).

5.4.5 BDM Command Structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.

8-bit reads return 16-bits of data, of which, only one byte will contain valid data. If reading an even address, the valid data will appear in the MSB. If reading an odd address, the valid data will appear in the LSB.

after a system stop mode the handshake feature must be enabled again by sending the ACK_ENABLE command.

5.4.11 Serial Communication Time Out

The host initiates a host-to-target serial transmission by generating a falling edge on the BKGD pin. If BKGD is kept low for more than 128 target clock cycles, the target understands that a SYNC command was issued. In this case, the target will keep waiting for a rising edge on BKGD in order to answer the SYNC request pulse. If the rising edge is not detected, the target will keep waiting forever without any time-out limit.

Consider now the case where the host returns BKGD to logic one before 128 cycles. This is interpreted as a valid bit transmission, and not as a SYNC request. The target will keep waiting for another falling edge marking the start of a new bit. If, however, a new falling edge is not detected by the target within 512 clock cycles since the last falling edge, a time-out occurs and the current command is discarded without affecting memory or the operating mode of the MCU. This is referred to as a soft-reset.

If a read command is issued but the data is not retrieved within 512 serial clock cycles, a soft-reset will occur causing the command to be disregarded. The data is not available for retrieval after the time-out has occurred. This is the expected behavior if the handshake protocol is not enabled. However, consider the behavior where the BDM is running in a frequency much greater than the CPU frequency. In this case, the command could time out before the data is ready to be retrieved. In order to allow the data to be retrieved even with a large clock frequency mismatch (between BDM and CPU) when the hardware handshake protocol is enabled, the time out between a read command and the data retrieval is disabled. Therefore, the host could wait for more than 512 serial clock cycles and still be able to retrieve the data from an issued read command. However, once the handshake pulse (ACK pulse) is issued, the time-out feature is re-activated, meaning that the target will time out after 512 clock cycles. Therefore, the host needs to retrieve the data within a 512 serial clock cycles time frame after the ACK pulse had been issued. After that period, the read command is discarded and the data is no longer available for retrieval. Any negative edge in the BKGD pin after the time-out period is considered to be a new command or a SYNC request.

Note that whenever a partially issued command, or partially retrieved data, has occurred the time out in the serial communication is active. This means that if a time frame higher than 512 serial clock cycles is observed between two consecutive negative edges and the command being issued or data being retrieved is not complete, a soft-reset will occur causing the partially received command or data retrieved to be disregarded. The next negative edge in the BKGD pin, after a soft-reset has occurred, is considered by the target as the start of a new BDM command, or the start of a SYNC request pulse.

```

SUB_1    BRN      *                      ; JMP Destination address TRACE BUFFER ENTRY 1
                                                ; RTI Destination address TRACE BUFFER ENTRY 3
        NOP                      ;
ADDR1    DBNE     A, PART5              ; Source address TRACE BUFFER ENTRY 4

IRQ_ISR  LDAB     #$F0                  ; IRQ Vector $FFF2 = TRACE BUFFER ENTRY 2
        STAB     VAR_C1
        RTI                      ;

```

The execution flow taking into account the IRQ is as follows

```

        LDX      #SUB_1
MARK1    JMP      0, X                  ;
IRQ_ISR  LDAB     #$F0                  ;
        STAB     VAR_C1
        RTI                      ;
SUB_1    BRN      *                      ;
        NOP                      ;
ADDR1    DBNE     A, PART5              ;

```

6.4.5.2.2 Loop1 Mode

Loop1 Mode, similarly to Normal Mode also stores only COF address information to the trace buffer, it however allows the filtering out of redundant information.

The intent of Loop1 Mode is to prevent the Trace Buffer from being filled entirely with duplicate information from a looping construct such as delays using the DBNE instruction or polling loops using BRSET/BRCLR instructions. Immediately after address information is placed in the Trace Buffer, the S12XDBG module writes this value into a background register. This prevents consecutive duplicate address entries in the Trace Buffer resulting from repeated branches.

Loop1 Mode only inhibits consecutive duplicate source address entries that would typically be stored in most tight looping constructs. It does not inhibit repeated entries of destination addresses or vector addresses, since repeated entries of these would most likely indicate a bug in the user's code that the S12XDBG module is designed to help find.

6.4.5.2.3 Detail Mode

In Detail Mode, address and data for all memory and register accesses is stored in the trace buffer. This mode also features information byte entries to the trace buffer, for each address byte entry. The information byte indicates the size of access (word or byte) and the type of access (read or write).

When tracing CPU12X activity in Detail Mode, all cycles are traced except those when the CPU12X is either in a free or opcode fetch cycle, the address range can be limited to a range specified by the TRANGE bits in DBGTCR. This function uses comparators C and D to define an address range inside which CPU12X activity should be traced (see Table 6-40). Thus the traced CPU12X activity can be restricted to particular register range accesses.

6.4.5.2.4 Pure PC Mode

In Pure PC Mode, tracing from the CPU the PC addresses of all executed opcodes, including illegal opcodes, are stored.

6.4.5.3 Trace Buffer Organization

Referring to Table 6-40. ADRH, ADRM, ADRL denote address high, middle and low byte respectively. INF bytes contain control information (R/W, S/D etc.). The numerical suffix indicates which tracing step. The information format for Loop1 Mode and PurePC Mode is the same as that of Normal Mode. Whilst tracing in Normal or Loop1 modes each array line contains 2 data entries, thus in this case the DBGCNT[0] is incremented after each separate entry. In Detail mode DBGCNT[0] remains cleared whilst the other DBGCNT bits are incremented on each trace buffer entry.

When a COF occurs a trace buffer entry is made and the corresponding CDV bit is set.

Single byte data accesses in Detail Mode are always stored to the low byte of the trace buffer (CDATAL) and the high byte is cleared. When tracing word accesses, the byte at the lower address is always stored to trace buffer byte3 and the byte at the higher address is stored to byte2.

Table 6-40. Trace Buffer Organization

Mode	8-Byte Wide Word Buffer							
	7	6	5	4	3	2	1	0
S12XCPU Detail	CXINF1	CADRH1	CADRM1	CADRL1	CDATAH1	CDATAL1		
	CXINF2	CADRH2	CADRM2	CADRL2	CDATAH2	CDATAL2		
CPU12X Other Modes	CINF1	CPCH1	CPCM1	CPCL1	CINF0	CPCH0	CPCM0	CPCL0
	CINF3	CPCH3	CPCM3	CPCL3	CINF2	CPCH2	CPCM2	CPCL2

8.3 Memory Map and Registers

This section provides a detailed description of all registers accessible in the S12XECRG.

8.3.1 Module Memory Map

Figure 8-2 gives an overview on all S12XECRG registers.

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000	SYNR	R W	VCOFRQ[1:0]		SYNDIV[5:0]					
0x0001	REFDV	R W	REFFRQ[1:0]		REFDIV[5:0]					
0x0002	POSTDIV	R W	0	0	0	POSTDIV[4:0]				
0x0003	CRGFLG	R W	RTIF	PORF	LVRF	LOCKIF	LOCK	ILAF	SCMIF	SCM
0x0004	CRGINT	R W	RTIE	0	0	LOCKIE	0	0	SCMIE	0
0x0005	CLKSEL	R W	PLLSEL	PSTP	XCLKS	0	PLLWAI	0	RTIWAI	COPWAI
0x0006	PLLCTL	R W	CME	PLLON	FM1	FM0	FSTWKP	PRE	PCE	SCME
0x0007	RTICTL	R W	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
0x0008	COPCTL	R W	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
0x0009	FORBYP ²	R W	0	0	0	0	0	0	0	0
0x000A	CTCTL ²	R W	0	0	0	0	0	0	0	0
0x000B	ARMCOP	R W	0	0	0	0	0	0	0	0
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

2. FORBYP and CTCTL are intended for factory test purposes only.


 = Unimplemented or Reserved

Figure 8-2. CRG Register Summary

NOTE

Register Address = Base Address + Address Offset, where the Base Address is defined at the MCU level and the Address Offset is defined at the module level.

Table 8-6. CLKSEL Field Descriptions

Field	Description
7 PLLSEL	PLL Select Bit Write: Anytime. Writing a one when LOCK=0 has no effect. This prevents the selection of an unstable PLLCLK as SYSCCLK. PLLSEL bit is cleared when the MCU enters Self Clock Mode, Stop Mode or Wait Mode with PLLWAI bit set. It is recommended to read back the PLLSEL bit to make sure PLLCLK has really been selected as SYSCCLK, as LOCK status bit could theoretically change at the very moment writing the PLLSEL bit. 0 System clocks are derived from OSCCLK ($f_{BUS} = f_{OSC} / 2$). 1 System clocks are derived from PLLCLK ($f_{BUS} = f_{PLL} / 2$).
6 PSTP	Pseudo Stop Bit Write: Anytime This bit controls the functionality of the oscillator during Stop Mode. 0 Oscillator is disabled in Stop Mode. 1 Oscillator continues to run in Stop Mode (Pseudo Stop). Note: Pseudo Stop Mode allows for faster STOP recovery and reduces the mechanical stress and aging of the resonator in case of frequent STOP conditions at the expense of a slightly increased power consumption.
5 XCLKS	Oscillator Configuration Status Bit — This read-only bit shows the oscillator configuration status. 0 Loop controlled Pierce Oscillator is selected. 1 External clock / full swing Pierce Oscillator is selected.
3 PLLWAI	PLL Stops in Wait Mode Bit Write: Anytime If PLLWAI is set, the S12XECRG will clear the PLLSEL bit before entering Wait Mode. The PLLON bit remains set during Wait Mode but the IPLL is powered down. Upon exiting Wait Mode, the PLLSEL bit has to be set manually if PLL clock is required. 0 IPLL keeps running in Wait Mode. 1 IPLL stops in Wait Mode.
1 RTIWAI	RTI Stops in Wait Mode Bit Write: Anytime 0 RTI keeps running in Wait Mode. 1 RTI stops and initializes the RTI dividers whenever the part goes into Wait Mode.
0 COPWAI	COP Stops in Wait Mode Bit Normal modes: Write once Special modes: Write anytime 0 COP keeps running in Wait Mode. 1 COP stops and initializes the COP counter whenever the part goes into Wait Mode.

8.3.2.7 S12XECRG IPLL Control Register (PLLCTL)

This register controls the IPLL functionality.

Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	CME	PLLON	FM1	FM0	FSTWKP	PRE	PCE	SCME
W								
Reset	1	1	0	0	0	0	0	1

Figure 8-9. S12XECRG IPLL Control Register (PLLCTL)

Table 8-8. FM Amplitude selection

FM1	FM0	FM Amplitude / f_{VCO} Variation
0	0	FM off
0	1	$\pm 1\%$
1	0	$\pm 2\%$
1	1	$\pm 4\%$

8.3.2.8 S12XECRG RTI Control Register (RTICTL)

This register selects the timeout period for the Real Time Interrupt.

Module Base + 0x0007

	7	6	5	4	3	2	1	0
R	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
W								
Reset	0	0	0	0	0	0	0	0

Figure 8-10. S12XECRG RTI Control Register (RTICTL)

Read: Anytime

Write: Anytime

NOTE

A write to this register initializes the RTI counter.

Table 8-9. RTICTL Field Descriptions

Field	Description
7 RTDEC	Decimal or Binary Divider Select Bit — RTDEC selects decimal or binary based prescaler values. 0 Binary based divider value. See Table 8-10 1 Decimal based divider value. See Table 8-11
6–4 RTR[6:4]	Real Time Interrupt Prescale Rate Select Bits — These bits select the prescale rate for the RTI. See Table 8-10 and Table 8-11.
3–0 RTR[3:0]	Real Time Interrupt Modulus Counter Select Bits — These bits select the modulus counter target value to provide additional granularity. Table 8-10 and Table 8-11 show all possible divide values selectable by the RTICTL register. The source clock for the RTI is OSCCLK.

Table 8-10. RTI Frequency Divide Rates for RTDEC = 0

RTR[3:0]	RTR[6:4] =							
	000 (OFF)	001 (2^{10})	010 (2^{11})	011 (2^{12})	100 (2^{13})	101 (2^{14})	110 (2^{15})	111 (2^{16})
0000 ($\div 1$)	OFF ⁽¹⁾	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}

Table 11-9. Time Segment 2 Values

TSEG22	TSEG21	TSEG20	Time Segment 2
0	0	0	1 Tq clock cycle ⁽¹⁾
0	0	1	2 Tq clock cycles
:	:	:	:
1	1	0	7 Tq clock cycles
1	1	1	8 Tq clock cycles

1. This setting is not valid. Please refer to Table 11-37 for valid settings.

Table 11-10. Time Segment 1 Values

TSEG13	TSEG12	TSEG11	TSEG10	Time segment 1
0	0	0	0	1 Tq clock cycle ⁽¹⁾
0	0	0	1	2 Tq clock cycles ¹
0	0	1	0	3 Tq clock cycles ¹
0	0	1	1	4 Tq clock cycles
:	:	:	:	:
1	1	1	0	15 Tq clock cycles
1	1	1	1	16 Tq clock cycles

1. This setting is not valid. Please refer to Table 11-37 for valid settings.

The bit time is determined by the oscillator frequency, the baud rate prescaler, and the number of time quanta (Tq) clock cycles per bit (as shown in Table 11-9 and Table 11-10).

Eqn. 11-1

$$\text{Bit Time} = \frac{(\text{Prescaler value})}{f_{\text{CANCLK}}} \cdot (1 + \text{TimeSegment1} + \text{TimeSegment2})$$

11.3.2.5 MSCAN Receiver Flag Register (CANRFLG)

A flag can be cleared only by software (writing a 1 to the corresponding bit position) when the condition which caused the setting is no longer valid. Every flag has an associated interrupt enable bit in the CANIER register.

Module Base + 0x0004

Access: User read/write⁽¹⁾

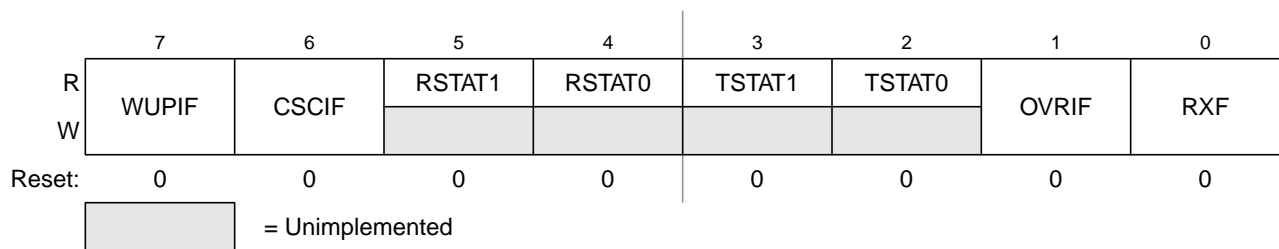


Figure 11-8. MSCAN Receiver Flag Register (CANRFLG)

- The transmission buffer with the lowest local priority field wins the prioritization.

In cases of more than one buffer having the same lowest priority, the message buffer with the lower index number wins.

Module Base + 0x00XD

Access: User read/write⁽¹⁾

	7	6	5	4	3	2	1	0
R	PRI07	PRI06	PRI05	PRI04	PRI03	PRI02	PRI01	PRI00
W								
Reset:	0	0	0	0	0	0	0	0

Figure 11-36. Transmit Buffer Priority Register (TBPR)

1. Read: Anytime when TXEx flag is set (see Section 11.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see Section 11.3.2.11, "MSCAN Transmit Buffer Selection Register (CANTBSEL)")
Write: Anytime when TXEx flag is set (see Section 11.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see Section 11.3.2.11, "MSCAN Transmit Buffer Selection Register (CANTBSEL)")

11.3.3.5 Time Stamp Register (TSRH–TSRL)

If the TIME bit is enabled, the MSCAN will write a time stamp to the respective registers in the active transmit or receive buffer right after the EOF of a valid message on the CAN bus (see Section 11.3.2.1, "MSCAN Control Register 0 (CANCTL0)"). In case of a transmission, the CPU can only read the time stamp after the respective transmit buffer has been flagged empty.

The timer value, which is used for stamping, is taken from a free running internal CAN bit clock. A timer overrun is not indicated by the MSCAN. The timer is reset (all bits set to 0) during initialization mode. The CPU can only read the time stamp registers.

Module Base + 0x00XE

Access: User read/write⁽¹⁾

	7	6	5	4	3	2	1	0
R	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8
W								
Reset:	x	x	x	x	x	x	x	x

Figure 11-37. Time Stamp Register — High Byte (TSRH)

1. Read: Anytime when TXEx flag is set (see Section 11.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see Section 11.3.2.11, "MSCAN Transmit Buffer Selection Register (CANTBSEL)")
Write: Unimplemented

Chapter 13

Pulse-Width Modulator (S12PWM8B8CV1)

Version Number	Revision Date	Effective Date	Author	Description of Changes
01.17		08-01-2004		Added clarification of PWMIF operation in STOP and WAIT mode. Added notes on minimum pulse width of emergency shutdown signal.

13.1 Introduction

The PWM definition is based on the HC12 PWM definitions. It contains the basic features from the HC11 with some of the enhancements incorporated on the HC12: center aligned output mode and four available clock sources. The PWM module has eight channels with independent control of left and center aligned outputs on each channel.

Each of the eight channels has a programmable period and duty cycle as well as a dedicated counter. A flexible clock select scheme allows a total of four different clock sources to be used with the counters. Each of the modulators can create independent continuous waveforms with software-selectable duty rates from 0% to 100%. The PWM outputs can be programmed as left aligned outputs or center aligned outputs.

13.1.1 Features

The PWM block includes these distinctive features:

- Eight independent PWM channels with programmable period and duty cycle
- Dedicated counter for each PWM channel
- Programmable PWM enable/disable for each channel
- Software selection of PWM duty pulse polarity for each channel
- Period and duty cycle are double buffered. Change takes effect when the end of the effective period is reached (PWM counter reaches zero) or when the channel is disabled.
- Programmable center or left aligned outputs on individual channels
- Eight 8-bit channel or four 16-bit channel PWM resolution
- Four clock sources (A, B, SA, and SB) provide for a wide range of frequencies
- Programmable clock select logic
- Emergency shutdown

On the front end of the PWM timer, the clock is enabled to the PWM circuit by the PWME_x bit being high. There is an edge-synchronizing circuit to guarantee that the clock will only be enabled or disabled at an edge. When the channel is disabled (PWME_x = 0), the counter for the channel does not count.

13.4.2.2 PWM Polarity

Each channel has a polarity bit to allow starting a waveform cycle with a high or low signal. This is shown on the block diagram as a mux select of either the Q output or the \overline{Q} output of the PWM output flip flop. When one of the bits in the PWMPOL register is set, the associated PWM channel output is high at the beginning of the waveform, then goes low when the duty count is reached. Conversely, if the polarity bit is zero, the output starts low and then goes high when the duty count is reached.

13.4.2.3 PWM Period and Duty

Dedicated period and duty registers exist for each channel and are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to \$00)
- The channel is disabled

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period and duty registers will go directly to the latches as well as the buffer.

A change in duty or period can be forced into effect “immediately” by writing the new value to the duty and/or period registers and then writing to the counter. This forces the counter to reset and the new duty and/or period values to be latched. In addition, since the counter is readable, it is possible to know where the count is with respect to the duty value and software can be used to make adjustments

NOTE

When forcing a new period or duty into effect immediately, an irregular PWM cycle can occur.

Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time.

13.4.2.4 PWM Timer Counters

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source (see Section 13.4.1, “PWM Clock Select” for the available clock sources and rates). The counter compares to two registers, a duty register and a period register as shown in Figure 13-19. When the PWM counter matches the duty register, the output flip-flop changes state, causing the PWM waveform to also change state. A match between the PWM counter and the period register behaves differently depending on what output mode is selected as shown in Figure 13-19 and described in Section 13.4.2.5, “Left Aligned Outputs” and Section 13.4.2.6, “Center Aligned Outputs”.

13.6 Interrupts

The PWM module has only one interrupt which is generated at the time of emergency shutdown, if the corresponding enable bit (PWMIE) is set. This bit is the enable for the interrupt. The interrupt flag PWMIF is set whenever the input level of the PWM7 channel changes while PWM7ENA = 1 or when PWMENA is being asserted while the level at PWM7 is active.

In stop mode or wait mode (with the PSWAI bit set), the emergency shutdown feature will drive the PWM outputs to their shutdown output levels but the PWMIF flag will not be set.

A description of the registers involved and affected due to this interrupt is explained in Section 13.3.2.15, “PWM Shutdown Register (PWMSDN)”.

The PWM block only generates the interrupt and does not service it. The interrupt signal name is PWM interrupt signal.

14.3.2.5 SCI Alternative Control Register 2 (SCIACR2)

Module Base + 0x0002

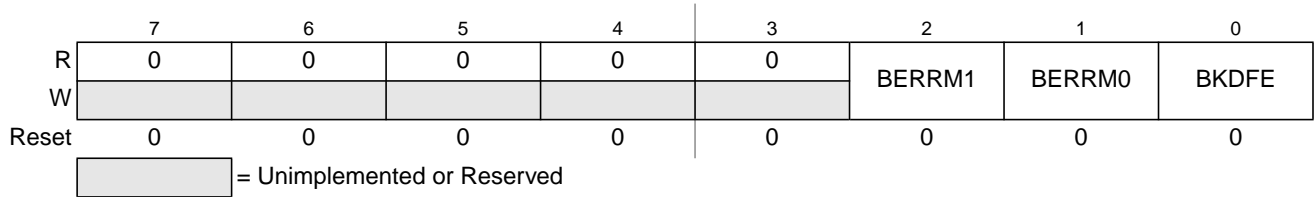


Figure 14-8. SCI Alternative Control Register 2 (SCIACR2)

Read: Anytime, if AMAP = 1

Write: Anytime, if AMAP = 1

Table 14-8. SCIACR2 Field Descriptions

Field	Description
2:1 BERRM[1:0]	Bit Error Mode — Those two bits determines the functionality of the bit error detect feature. See Table 14-9.
0 BKDFE	Break Detect Feature Enable — BKDFE enables the break detect circuitry. 0 Break detect circuit disabled 1 Break detect circuit enabled

Table 14-9. Bit Error Mode Coding

BERRM1	BERRM0	Function
0	0	Bit error detect circuit is disabled
0	1	Receive input sampling occurs during the 9th time tick of a transmitted bit (refer to Figure 14-19)
1	0	Receive input sampling occurs during the 13th time tick of a transmitted bit (refer to Figure 14-19)
1	1	Reserved

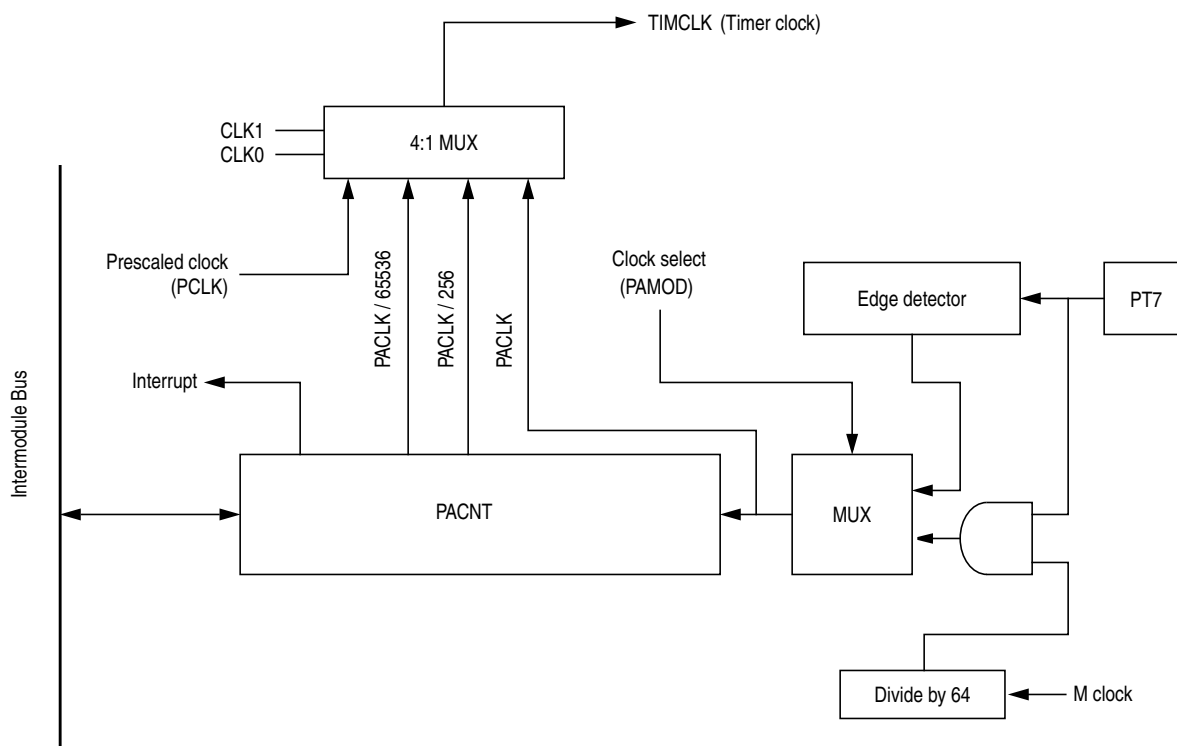


Figure 16-2. 16-Bit Pulse Accumulator Block Diagram

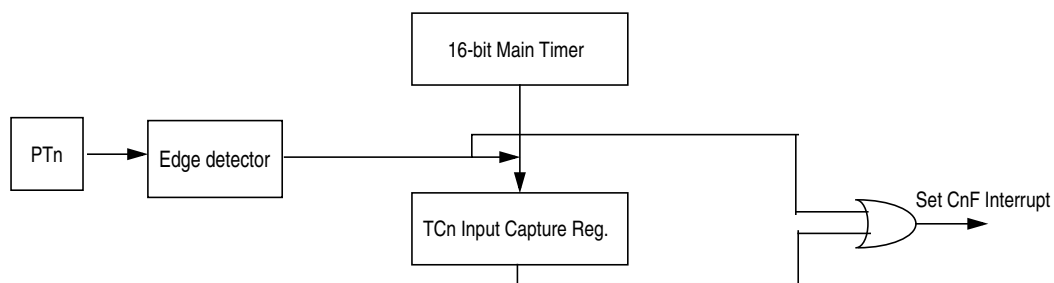


Figure 16-3. Interrupt Flag Setting

Table 16-19. Pin Action

PAMOD	PEDGE	Pin Action
0	0	Falling edge
0	1	Rising edge
1	0	Div. by 64 clock enabled with pin high level
1	1	Div. by 64 clock enabled with pin low level

NOTE

If the timer is not active (TEN = 0 in TSCR), there is no divide-by-64 because the ÷64 clock is generated by the timer prescaler.

Table 16-20. Timer Clock Selection

CLK1	CLK0	Timer Clock
0	0	Use timer prescaler clock as timer counter clock
0	1	Use PACLK as input to timer counter clock
1	0	Use PACLK/256 as timer counter clock frequency
1	1	Use PACLK/65536 as timer counter clock frequency

For the description of PACLK please refer Figure 16-30.

If the pulse accumulator is disabled (PAEN = 0), the prescaler clock from the timer is always used as an input clock to the timer counter. The change from one selected clock to the other happens immediately after these bits are written.

16.3.2.16 Pulse Accumulator Flag Register (PAFLG)

Module Base + 0x0021

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PAOVF	PAIF
W								
Reset	0	0	0	0	0	0	0	0
	Unimplemented or Reserved							

Figure 16-25. Pulse Accumulator Flag Register (PAFLG)

Read: Anytime

Write: Anytime

When the TFFCA bit in the TSCR register is set, any access to the PACNT register will clear all the flags in the PAFLG register. Timer module or Pulse Accumulator must stay enabled (TEN=1 or PAEN=1) while clearing these bits.

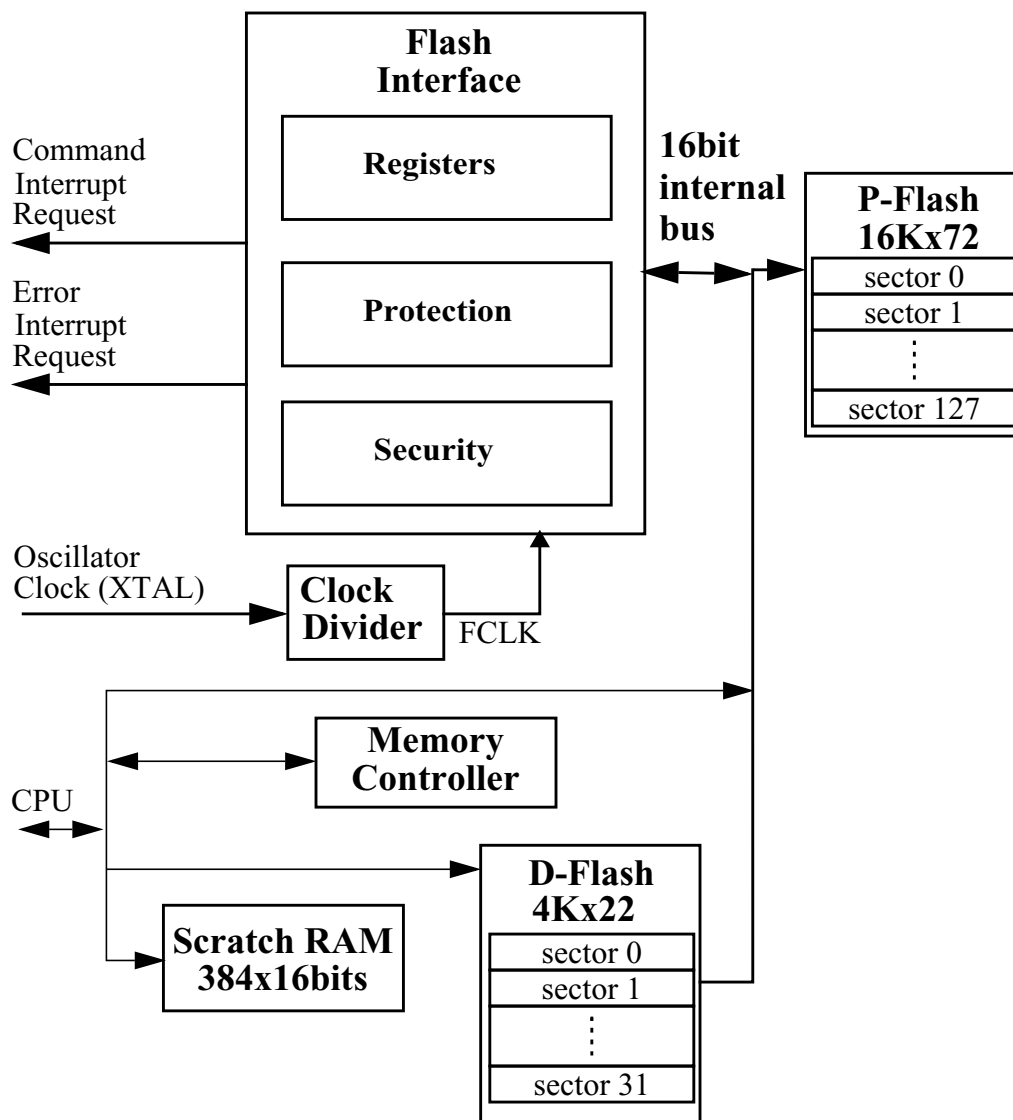


Figure 19-1. FTMR128K1 Block Diagram

19.2 External Signal Description

The Flash module contains no signals that connect off-chip.

19.3 Memory Map and Registers

This section describes the memory map and registers for the Flash module. Read data from unimplemented memory space in the Flash module is undefined. Write access to unimplemented or reserved memory space in the Flash module will be ignored by the Flash module.

Table 19-30. D-Flash Commands

FCMD	Command	Function on D-Flash Memory
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all D-Flash (and P-Flash) blocks and verifying that all D-Flash (and P-Flash) blocks are erased.
0x0D	Set User Margin Level	Specifies a user margin read level for the D-Flash block.
0x0E	Set Field Margin Level	Specifies a field margin read level for the D-Flash block (special modes only).
0x10	Erase Verify D-Flash Section	Verify that a given number of words starting at the address provided are erased.
0x11	Program D-Flash	Program up to four words in the D-Flash block.
0x12	Erase D-Flash Sector	Erase all bytes in a sector of the D-Flash block.

19.4.2 Flash Command Description

This section provides details of all available Flash commands launched by a command write sequence. The ACCERR bit in the FSTAT register will be set during the command write sequence if any of the following illegal steps are performed, causing the command not to be processed by the Memory Controller:

- Starting any command write sequence that programs or erases Flash memory before initializing the FCLKDIV register
- Writing an invalid command as part of the command write sequence
- For additional possible errors, refer to the error handling table provided for each command

If a Flash block is read during execution of an algorithm (CCIF = 0) on that same block, the read operation will return invalid data. If the SFDIF or DFDIF flags were not previously set when the invalid read operation occurred, both the SFDIF and DFDIF flags will be set and the FECCR registers will be loaded with the global address used in the invalid read operation with the data and parity fields set to all 0.

If the ACCERR or FPVIOL bits are set in the FSTAT register, the user must clear these bits before starting any command write sequence (see Section 19.3.2.7).

CAUTION

A Flash word or phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash word or phrase is not allowed.

19.4.2.1 Erase Verify All Blocks Command

The Erase Verify All Blocks command will verify that all P-Flash and D-Flash blocks have been erased.

Table 19-31. Erase Verify All Blocks Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x01	Not required

0x02C0–0x02EF Analog-to-Digital Converter 12-Bit 16-Channel (ATD0) Map (continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02E6	ATD0DR11H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02E7	ATD0DR11L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02E8	ATD0DR12H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02E9	ATD0DR12L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02EA	ATD0DR13H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02EB	ATD0DR13L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02EC	ATD0DR14H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02ED	ATD0DR14L	R	Bit7	Bit6	0	0	0	0	0	0
		W								
0x02EE	ATD0DR15H	R	Bit15	14	13	12	11	10	9	Bit8
		W								
0x02EF	ATD0DR15L	R	Bit7	Bit6	0	0	0	0	0	0
		W								

0x02F0–0x02F7 Voltage Regulator (VREG_3V3) Map

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02F0	VREGHTCL	R	0	0	VSEL	VAE	HTEN	HTDS	HTIE	HTIF
		W								
0x02F1	VREGCTRL	R	0	0	0	0	0	LVDS	LVIE	LVIF
		W								
0x02F2	VREGAPICL	R	APICLK	0	0	APIFES	APIEA	APIFE	APIE	APIF
		W								
0x02F3	VREGAPITR	R	APITR5	APITR4	APITR3	APITR2	APITR1	APITR0	0	0
		W								
0x02F4	VREGAPIRH	R	APIR15	APIR14	APIR13	APIR12	APIR11	APIR10	APIR9	APIR8
		W								
0x02F5	VREGAPIRL	R	APIR7	APIR6	APIR5	APIR4	APIR3	APIR2	APIR1	APIR0
		W								
0x02F6	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x02F7	VREGHTTR	R	HTOEN	0	0	0	HTTR3	HTTR2	HTTR1	HTTR0
		W								

0x02F8–0x02FF Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02F8– 0x02FF	Reserved	R	0	0	0	0	0	0	0	0
		W								