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	91
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912xet256bmalr

Chapter 1
Device Overview MC9S12XE-Family

1.1 Introduction

The MC9S12XE-Family of micro controllers is a further development of the S12XD-Family including new features for enhanced system integrity and greater functionality. These new features include a Memory Protection Unit (MPU) and Error Correction Code (ECC) on the Flash memory together with enhanced EEPROM functionality (EEE), an enhanced XGATE, an Internally filtered, frequency modulated Phase Locked Loop (IPLL) and an enhanced ATD. The E-Family extends the S12X product range up to 1MB of Flash memory with increased I/O capability in the 208-pin version of the flagship MC9S12XE100.

The MC9S12XE-Family delivers 32-bit performance with all the advantages and efficiencies of a 16 bit MCU. It retains the low cost, power consumption, EMC and code-size efficiency advantages currently enjoyed by users of Freescale's existing 16-Bit MC9S12 and S12X MCU families. There is a high level of compatibility between the S12XE and S12XD families.

The MC9S12XE-Family features an enhanced version of the performance-boosting XGATE co-processor which is programmable in "C" language and runs at twice the bus frequency of the S12X with an instruction set optimized for data movement, logic and bit manipulation instructions and which can service any peripheral module on the device. The new enhanced version has improved interrupt handling capability and is fully compatible with the existing XGATE module.

The MC9S12XE-Family is composed of standard on-chip peripherals including up to 64Kbytes of RAM, eight asynchronous serial communications interfaces (SCI), three serial peripheral interfaces (SPI), an 8-channel IC/OC enhanced capture timer (ECT), two 16-channel, 12-bit analog-to-digital converters, an 8-channel pulse-width modulator (PWM), five CAN 2.0 A, B software compatible modules (MSCAN12), two inter-IC bus blocks (IIC), an 8-channel 24-bit periodic interrupt timer (PIT) and an 8-channel 16-bit standard timer module (TIM).

The MC9S12XE-Family uses 16-bit wide accesses without wait states for all peripherals and memories. The non-multiplexed expanded bus interface available on the 144/208-Pin versions allows an easy interface to external memories.

In addition to the I/O ports available in each module, up to 26 further I/O ports are available with interrupt capability allowing Wake-Up from STOP or WAIT modes. The MC9S12XE-Family is available in 208-Pin MAPBGA, 144-Pin LQFP, 112-Pin LQFP or 80-Pin QFP options.

1.1.1 Features

Features of the MC9S12XE-Family are listed here. Please see Table D-2.for memory options and Table D-2. for the peripheral features that are available on the different family members.

Table 2-68. DDR1AD0 Register Field Descriptions

Field	Description
7-0 DDR1AD0	**Port AD0 data direction**— This register controls the data direction of pins 7 through 0. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PT0AD0 registers, when changing the DDR1AD0 register.

NOTE

To use the digital input function on Port AD0 the ATD Digital Input Enable Register (ATD0DIEN1) has to be set to logic level “1”.

2.3.73 Port AD0 Reduced Drive Register 0 (RDR0AD0)

Address 0x0274 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	RDR0AD07	RDR0AD06	RDR0AD05	RDR0AD04	RDR0AD03	RDR0AD02	RDR0AD01	RDR0AD00
Reset	0	0	0	0	0	0	0	0

Figure 2-71. Port AD0 Reduced Drive Register 0 (RDR0AD0)

1. Read: Anytime.
   Write: Anytime.

Table 2-69. RDR0AD0 Register Field Descriptions

Field	Description
7-0 RDR0AD0	**Port AD0 reduced drive**—Select reduced drive for Port AD0 outputs This register configures the drive strength of Port AD0 output pins 15 through 8 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.98 Port L Polarity Select Register (PPSL)

Address 0x0375 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	PPSL7	PPSL6	PPSL5	PPSL4	PPSL3	PPSL2	PPSL1	PPSL0
Reset	0	0	0	0	0	0	0	0

Figure 2-96. Port L Polarity Select Register (PPSL)

1. Read: Anytime.
   Write: Anytime.

Table 2-93. PPSL Register Field Descriptions

Field	Description
7-0 PPSL	**Port L pull device select**—Determine pull device polarity on input pins This register selects whether a pull-down or a pull-up device is connected to the pin. 1 A pull-down device is connected to the associated pin, if enabled and if the pin is used as input. 0 A pull-up device is connected to the associated pin, if enabled and if the pin is used as input.

2.3.99 Port L Wired-Or Mode Register (WOML)

Address 0x0376 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	WOML7	WOML6	WOML5	WOML4	WOML3	WOML2	WOML1	WOML0
Reset	0	0	0	0	0	0	0	0

Figure 2-97. Port L Wired-Or Mode Register (WOML)

1. Read: Anytime.
   Write: Anytime.

Table 2-94. WOML Register Field Descriptions

Field	Description
7-0 WOML	**Port L wired-or mode**—Enable wired-or functionality This register configures the output pins as wired-or independent of the function used on the pins. If enabled the output is driven active low only (open-drain). A logic level of “1” is not driven.This allows a multipoint connection of several serial modules. These bits have no influence on pins used as inputs. 1 Output buffers operate as open-drain outputs. 0 Output buffers operate as push-pull outputs.

in the memory map. This is to make sure that the correct PPAGE value will be present on stack at the time of the RTC instruction execution.

3.5.2 Port Replacement Registers (PRRs)

Registers used for emulation purposes must be rebuilt by the in-circuit emulator hardware to achieve full emulation of single chip mode operation. These registers are called port replacement registers (PRRs) (see Table 1-25). PRRs are accessible from CPU, BDM and XGATE using different access types (word aligned, word-misaligned and byte).

Each access to PRRs will be extended to 2 bus cycles for write or read accesses independent of the operating mode. In emulation modes all write operations result in simultaneous writing to the internal registers (peripheral access) and to the emulated registers (external access) located in the PRU in the emulator. All read operations are performed from external registers (external access) in emulation modes. In all other modes the read operations are performed from the internal registers (peripheral access).

Due to internal visibility of CPU accesses the CPU will be halted during XGATE or BDM access to any PRR. This rule applies also in normal modes to ensure that operation of the device is the same as in emulation modes.

A summary of PRR accesses:

- An aligned word access to a PRR will take 2 bus cycles.
- A misaligned word access to a PRRs will take 4 cycles. If one of the two bytes accessed by the misaligned word access is not a PRR, the access will take only 3 cycles.
- A byte access to a PRR will take 2 cycles.

7.4.1 Security

If the user resets into special single chip mode with the system secured, a secured mode BDM firmware lookup table is brought into the map overlapping a portion of the standard BDM firmware lookup table. The secure BDM firmware verifies that the on-chip non-volatile memory (e.g. EEPROM and Flash EEPROM) is erased. This being the case, the UNSEC and ENBDM bit will get set. The BDM program jumps to the start of the standard BDM firmware and the secured mode BDM firmware is turned off and all BDM commands are allowed. If the non-volatile memory does not verify as erased, the BDM firmware sets the ENBDM bit, without asserting UNSEC, and the firmware enters a loop. This causes the BDM hardware commands to become enabled, but does not enable the firmware commands. This allows the BDM hardware to be used to erase the non-volatile memory.

BDM operation is not possible in any other mode than special single chip mode when the device is secured. The device can be unsecured via BDM serial interface in special single chip mode only. More information regarding security is provided in the security section of the device documentation.

7.4.2 Enabling and Activating BDM

The system must be in active BDM to execute standard BDM firmware commands. BDM can be activated only after being enabled. BDM is enabled by setting the ENBDM bit in the BDM status (BDMSTS) register. The ENBDM bit is set by writing to the BDM status (BDMSTS) register, via the single-wire interface, using a hardware command such as WRITE_BD_BYTE.

After being enabled, BDM is activated by one of the following[1]:

- Hardware BACKGROUND command
- CPU BGND instruction
- External instruction tagging mechanism[2]
- Breakpoint force or tag mechanism[2]

When BDM is activated, the CPU finishes executing the current instruction and then begins executing the firmware in the standard BDM firmware lookup table. When BDM is activated by a breakpoint, the type of breakpoint used determines if BDM becomes active before or after execution of the next instruction.

NOTE

If an attempt is made to activate BDM before being enabled, the CPU resumes normal instruction execution after a brief delay. If BDM is not enabled, any hardware BACKGROUND commands issued are ignored by the BDM and the CPU is not delayed.

In active BDM, the BDM registers and standard BDM firmware lookup table are mapped to addresses 0x7FFF00 to 0x7FFFFF. BDM registers are mapped to addresses 0x7FFF00 to 0x7FFF0B. The BDM uses these registers which are readable anytime by the BDM. However, these registers are not readable by user programs.

1. BDM is enabled and active immediately out of special single-chip reset.
2. This method is provided by the S12X_DBG module.

Chapter 8
S12X Debug (S12XDBGV3) Module

Table 8-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V03.20	14 Sep 2007	8.3.2.7/8-317	- Clarified reserved State Sequencer encodings.
V03.21	23 Oct 2007	8.4.2.2/8-329 8.4.2.4/8-330	- Added single databyte comparison limitation information - Added statement about interrupt vector fetches whilst tagging.
V03.22	12 Nov 2007	8.4.5.2/8-334 8.4.5.5/8-341	- Removed LOOP1 tracing restriction NOTE. - Added pin reset effect NOTE.
V03.23	13 Nov 2007	General	- Text readability improved, typo removed.
V03.24	04 Jan 2008	8.4.5.3/8-336	- Corrected bit name.
V03.25	14 May 2008	General	- Updated Revision History Table format. Corrected other paragraph formats.
V03.26	12 Sep 2012	General	- Added missing full stops. Removed redundant quotation marks.

8.1 Introduction

The S12XDBG module provides an on-chip trace buffer with flexible triggering capability to allow non-intrusive debug of application software. The S12XDBG module is optimized for the S12X 16-bit architecture and allows debugging of CPU12Xand XGATE module operations.

Typically the S12XDBG module is used in conjunction with the S12XBDM module, whereby the user configures the S12XDBG module for a debugging session over the BDM interface. Once configured the S12XDBG module is armed and the device leaves BDM Mode returning control to the user program, which is then monitored by the S12XDBG module. Alternatively the S12XDBG module can be configured over a serial interface using SWI routines.

8.1.1 Glossary

Table 8-2. Glossary Of Terms

Term	Definition
COF	Change Of Flow. Change in the program flow due to a conditional branch, indexed jump or interrupt
BDM	Background Debug Mode
DUG	Device User Guide, describing the features of the device into which the DBG is integrated
WORD	16-bit data entity

SEX — Sign Extend Byte to Word — SEX

Operation

The result in RD is the 16 bit sign extended representation of the original two's complement number in the low byte of RD.L.

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
SEX RD	MON	0	0	0	0	0	RD	1	1	1	1	0	1	0	0	P

```
                        ;###########################################
                        ;#           INITIALIZE S12XE CORE         #
                        ;###########################################
                        SEI
                        MOVB  #(RAM_START_GLOB>>12), RPAGE    ;set RAM page

                        ;###########################################
                        ;#              INITIALIZE SCI             #
                        ;###########################################
INIT_SCI                MOVW  #(BUS_FREQ_HZ/(16*9600)), SCIBDH;set baud rate
                        MOVB  #(TIE|TE), SCICR2;enable tx buffer empty interrupt

                        ;###########################################
                        ;#           INITIALIZE S12X_INT           #
                        ;###########################################
INIT_INT                MOVB  #(SCI_VEC&$F0), INT_CFADDR      ;switch SCI interrupts to XGATE
                        MOVB  #RQST|$01, INT_CFDATA+((SCI_VEC&$0F)>>1)

                        ;###########################################
                        ;#             INITIALIZE XGATE            #
                        ;###########################################
INIT_XGATE              MOVW  #XGMCTL_CLEAR, XGMCTL           ;clear all XGMCTL bits

INIT_XGATE_BUSY_LOOP    TST   XGCHID          ;wait until current thread is finished
                        BNE   INIT_XGATE_BUSY_LOOP

                        LDX   #XGIF           ;clear all channel interrupt flags
                        LDD   #$FFFF
                        STD   2,X+
                        STD   2,X+
                        STD   2,X+
                        STD   2,X+
                        STD   2,X+
                        STD   2,X+
                        STD   2,X+
                        STD   2,X+

                        CLR   XGISPSEL        ;set vector base register
                        MOVW  #XGATE_VECTORS_XG, XGVBR
                        MOVW  #$FF00, XGSWT   ;clear all software triggers

                        ;###########################################
                        ;#      INITIALIZE XGATE VECTOR TABLE      #
                        ;###########################################
                        LDAA  #128            ;build XGATE vector table
                        LDY   #XGATE_VECTORS
INIT_XGATE_VECTAB_LOOP  MOVW  #XGATE_DUMMY_ISR_XG, 4,Y+
                        DBNE  A, INIT_XGATE_VECTAB_LOOP

                        MOVW  #XGATE_CODE_XG, RAM_START+(2*SCI_VEC)
                        MOVW  #XGATE_DATA_XG, RAM_START+(2*SCI_VEC)+2

                        ;###########################################
                        ;#              COPY XGATE CODE            #
                        ;###########################################
COPY_XGATE_CODE         LDX   #XGATE_DATA_FLASH
COPY_XGATE_CODE_LOOP    MOVW  2,X+, 2,Y+
```

16.1.1 Glossary

Table 16-2. Terminology

ACK	Acknowledge of CAN message
CAN	Controller Area Network
CRC	Cyclic Redundancy Code
EOF	End of Frame
FIFO	First-In-First-Out Memory
IFS	Inter-Frame Sequence
SOF	Start of Frame
CPU bus	CPU related read/write data bus
CAN bus	CAN protocol related serial bus
oscillator clock	Direct clock from external oscillator
bus clock	CPU bus related clock
CAN clock	CAN protocol related clock

16.1.2 Block Diagram

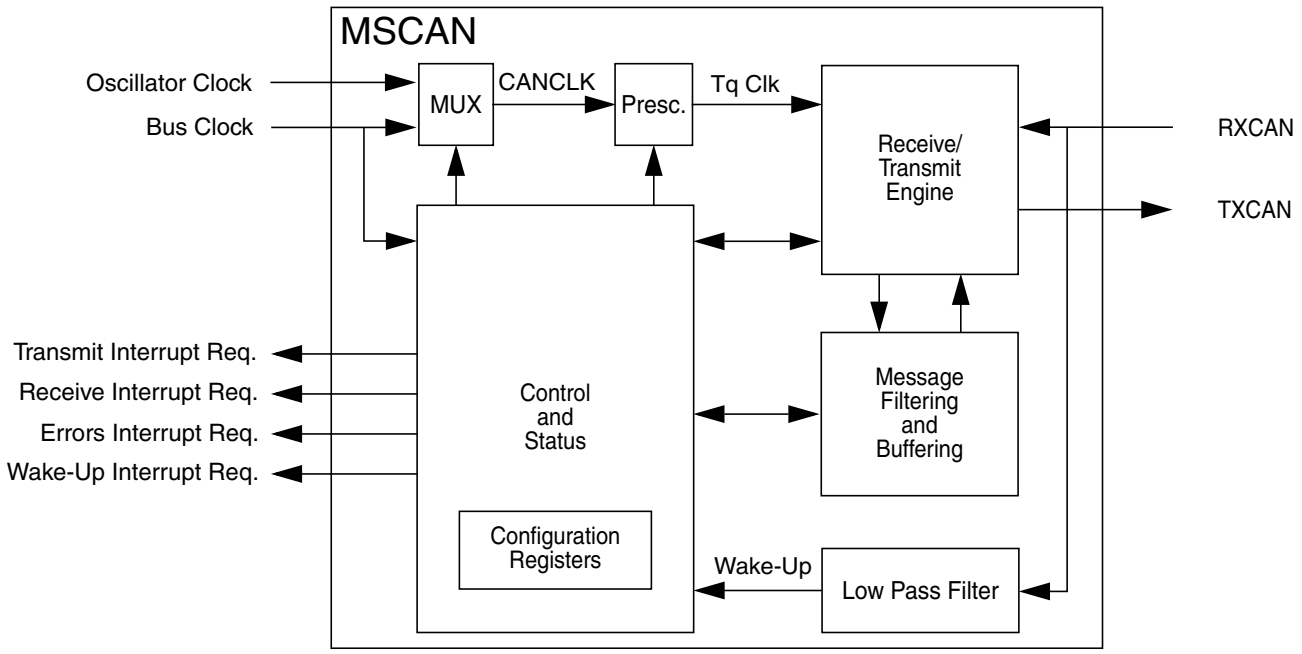

Figure 16-1. MSCAN Block Diagram

Chapter 17
Periodic Interrupt Timer (S12PIT24B8CV2)

Table 17-1. Revision History

Revision Number	RevisionDate	Sections Affected	Description of Changes
V01.00	28 Apr 2005		- Initial Release.
V01.01	05 Jul 2005	17.6/17-674	- Added application section. - Removed table 1-1.

17.1 Introduction

The period interrupt timer (PIT) is an array of 24-bit timers that can be used to trigger peripheral modules or raise periodic interrupts. Refer to Figure 17-1 for a simplified block diagram.

17.1.1 Glossary

Acronyms and Abbreviations	
PIT	Periodic Interrupt Timer
ISR	Interrupt Service Routine
CCR	Condition Code Register
SoC	System on Chip
micro time bases	clock periods of the 16-bit timer modulus down-counters, which are generated by the 8-bit modulus down-counters.

17.1.2 Features

The PIT includes these features:

- Eight timers implemented as modulus down-counters with independent time-out periods.
- Time-out periods selectable between 1 and 2^{24} bus clock cycles. Time-out equals m*n bus clock cycles with 1 <= m <= 256 and 1 <= n <= 65536.
- Timers that can be enabled individually.
- Eight time-out interrupts.
- Eight time-out trigger output signals available to trigger peripheral modules.
- Start of timer channels can be aligned to each other.

17.1.3 Modes of Operation

Refer to the device overview for a detailed explanation of the chip modes.

18.3.0.4 PIT Multiplex Register (PITMUX)

Module Base + 0x0003

	7	6	5	4	3	2	1	0
R	0	0	0	0	PMUX3	PMUX2	PMUX1	PMUX0
W								
Reset	0	0	0	0	0	0	0	0

Figure 18-6. PIT Multiplex Register (PITMUX)

Read: Anytime

Write: Anytime

Table 18-5. PITMUX Field Descriptions

Field	Description
3:0 PMUX[3:0]	**PIT Multiplex Bits for Timer Channel 3:0** — These bits select if the corresponding 16-bit timer is connected to micro time base 1 or 0. If PMUX is modified, the corresponding 16-bit timer is switched to the other micro time base immediately. 0 The corresponding 16-bit timer counts with micro time base 0. 1 The corresponding 16-bit timer counts with micro time base 1.

18.3.0.5 PIT Interrupt Enable Register (PITINTE)

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	0	0	0	0	PINTE3	PINTE2	PINTE1	PINTE0
W								
Reset	0	0	0	0	0	0	0	0

Figure 18-7. PIT Interrupt Enable Register (PITINTE)

Read: Anytime

Write: Anytime

Table 18-6. PITINTE Field Descriptions

Field	Description
3:0 PINTE[3:0]	**PIT Time-out Interrupt Enable Bits for Timer Channel 3:0** — These bits enable an interrupt service request whenever the time-out flag PTF of the corresponding PIT channel is set. When an interrupt is pending (PTF set) enabling the interrupt will immediately cause an interrupt. To avoid this, the corresponding PTF flag has to be cleared first. 0 Interrupt of the corresponding PIT channel is disabled. 1 Interrupt of the corresponding PIT channel is enabled.

To calculate the output frequency in center aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB) and divide it by twice the value in the period register for that channel.

- PWMx Frequency = Clock (A, B, SA, or SB) / (2*PWMPERx)
- PWMx Duty Cycle (high time as a% of period):
  - Polarity = 0 (PPOLx = 0)

    Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
  - Polarity = 1 (PPOLx = 1)

    Duty Cycle = [PWMDTYx / PWMPERx] * 100%

As an example of a center aligned output, consider the following case:

Clock Source = E, where E = 10 MHz (100 ns period)

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10 MHz/8 = 1.25 MHz

PWMx Period = 800 ns

PWMx Duty Cycle = 3/4 *100% = 75%

Shown in Figure 19-23 is the output waveform generated.

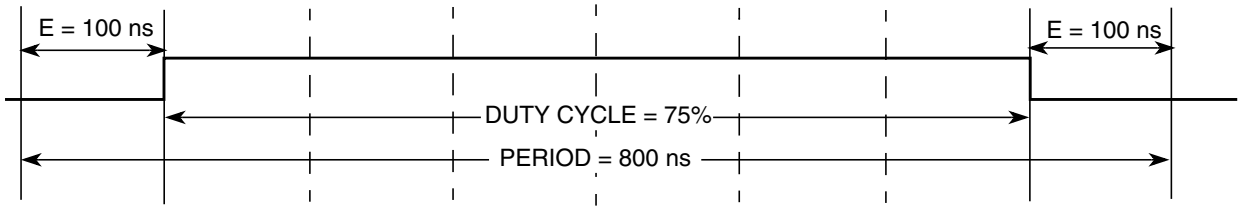

Figure 19-23. PWM Center Aligned Output Example Waveform

19.4.2.7 PWM 16-Bit Functions

The PWM timer also has the option of generating 8-channels of 8-bits or 4-channels of 16-bits for greater PWM resolution. This 16-bit channel option is achieved through the concatenation of two 8-bit channels.

The PWMCTL register contains four control bits, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. Channels 6 and 7 are concatenated with the CON67 bit, channels 4 and 5 are concatenated with the CON45 bit, channels 2 and 3 are concatenated with the CON23 bit, and channels 0 and 1 are concatenated with the CON01 bit.

NOTE

Change these bits only when both corresponding channels are disabled.

When channels 6 and 7 are concatenated, channel 6 registers become the high order bytes of the double byte channel, as shown in Figure 19-24. Similarly, when channels 4 and 5 are concatenated, channel 4 registers become the high order bytes of the double byte channel. When channels 2 and 3 are concatenated,

21.2.3 $\overline{SS}$ — Slave Select Pin

This pin is used to output the select signal from the SPI module to another peripheral with which a data transfer is to take place when it is configured as a master and it is used as an input to receive the slave select signal when the SPI is configured as slave.

21.2.4 SCK — Serial Clock Pin

In master mode, this is the synchronous output clock. In slave mode, this is the synchronous input clock.

21.3 Memory Map and Register Definition

This section provides a detailed description of address space and registers used by the SPI.

21.3.1 Module Memory Map

The memory map for the SPI is given in Figure 21-2. The address listed for each register is the sum of a base address and an address offset. The base address is defined at the SoC level and the address offset is defined at the module level. Reads from the reserved bits return zeros and writes to the reserved bits have no effect.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 SPICR1	R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
	W								
0x0001 SPICR2	R	0	XFRW	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
	W								
0x0002 SPIBR	R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
	W								
0x0003 SPISR	R	SPIF	0	SPTEF	MODF	0	0	0	0
	W								
0x0004 SPIDRH	R	R15	R14	R13	R12	R11	R10	R9	R8
	W	T15	T14	T13	T12	T11	T10	T9	T8
0x0005 SPIDRL	R	R7	R6	R5	R4	R3	R2	R1	R0
	W	T7	T6	T5	T4	T3	T2	T1	T0
0x0006 Reserved	R								
	W								
0x0007 Reserved	R								
	W								

= Unimplemented or Reserved

Figure 21-2. SPI Register Summary

Write: Has no meaning or effect in the normal mode; only writable in special modes (test_mode = 1).

The period of the first count after a write to the TCNT registers may be a different size because the write is not synchronized with the prescaler clock.

22.3.2.6 Timer System Control Register 1 (TSCR1)

Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	TEN	TSWAI	TSFRZ	TFFCA	PRNT	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 22-12. Timer System Control Register 1 (TSCR1)

Read: Anytime

Write: Anytime

Table 22-6. TSCR1 Field Descriptions

Field	Description
7 TEN	**Timer Enable** 0 Disables the main timer, including the counter. Can be used for reducing power consumption. 1 Allows the timer to function normally. If for any reason the timer is not active, there is no ÷64 clock for the pulse accumulator because the ÷64 is generated by the timer prescaler.
6 TSWAI	**Timer Module Stops While in Wait** 0 Allows the timer module to continue running during wait. 1 Disables the timer module when the MCU is in the wait mode. Timer interrupts cannot be used to get the MCU out of wait. TSWAI also affects pulse accumulator.
5 TSFRZ	**Timer Stops While in Freeze Mode** 0 Allows the timer counter to continue running while in freeze mode. 1 Disables the timer counter whenever the MCU is in freeze mode. This is useful for emulation. TSFRZ does not stop the pulse accumulator.

Table 22-17. TRLG2 Field Descriptions

Field	Description
7 TOF	**Timer Overflow Flag** — Set when 16-bit free-running timer overflows from 0xFFFF to 0x0000. Clearing this bit requires writing a one to bit 7 of TFLG2 register while the TEN bit of TSCR1 or PAEN bit of PACTL is set to one (See also TCRE control bit explanation.)

22.3.2.14 Timer Input Capture/Output Compare Registers High and Low 0–7 (TCxH and TCxL)

Module Base + 0x0010 = TC0H
0x0012 = TC1H
0x0014 = TC2H
0x0016 = TC3H
0x0018 = TC4H
0x001A = TC5H
0x001C = TC6H
0x001E = TC7H

	15	14	13	12	11	10	9	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 22-22. Timer Input Capture/Output Compare Register x High (TCxH)

Module Base + 0x0011 = TC0L
0x0013 = TC1L
0x0015 = TC2L
0x0017 = TC3L
0x0019 = TC4L
0x001B = TC5L
0x001D = TC6L
0x001F = TC7L

	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 22-23. Timer Input Capture/Output Compare Register x Low (TCxL)

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Read: Anytime

Write: Anytime for output compare function.Writes to these registers have no meaning or effect during input capture. All timer input capture/output compare registers are reset to 0x0000.

NOTE

Read/Write access in byte mode for high byte should takes place before low byte otherwise it will give a different result.

Table 24-12. Flash Security States

SEC[1:0]	Status of Security
00	SECURED
01	SECURED(1)
10	UNSECURED
11	SECURED

1. Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Section 24.5.

24.3.2.3 Flash CCOB Index Register (FCCOBIX)

The FCCOBIX register is used to index the FCCOB register for Flash memory operations.

Offset Module Base + 0x0002

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	CCOBIX[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 24-7. FCCOB Index Register (FCCOBIX)

CCOBIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 24-13. FCCOBIX Field Descriptions

Field	Description
2–0 CCOBIX[1:0]	**Common Command Register Index**— The CCOBIX bits are used to select which word of the FCCOB register array is being read or written to. See Section 24.3.2.11, "Flash Common Command Object Register (FCCOB)," for more details.

24.3.2.4 Flash ECCR Index Register (FECCRIX)

The FECCRIX register is used to index the FECCR register for ECC fault reporting.

Offset Module Base + 0x0003

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	ECCRIX[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 24-8. FECCR Index Register (FECCRIX)

ECCRIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 26-40. Read Once Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if a Load Data Field command sequence is currently active
		Set if command not available in current mode (see Table 26-30)
		Set if an invalid phrase index 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

26.4.2.5 Load Data Field Command

The Load Data Field command is executed to provide FCCOB parameters for multiple P-Flash blocks for a future simultaneous program operation in the P-Flash memory space.

Table 26-41. 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 27-76. EEPROM Emulation Query 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 command not available in current mode (see Table 27-30)
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

27.4.2.22 Partition D-Flash Command

The Partition D-Flash command allows the user to allocate sectors within the D-Flash block for applications and a partition within the buffer RAM for EEPROM access. The D-Flash block consists of 128 sectors with 256 bytes per sector. The Erase All Blocks command must be run prior to launching the Partition D-Flash command.

Table 27-77. Partition D-Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x20	Not required
001	Number of 256 byte sectors for the D-Flash user partition (DFPART)	
010	Number of 256 byte sectors for buffer RAM EEE partition (ERPART)	

Upon clearing CCIF to launch the Partition D-Flash command, the following actions are taken to define a partition within the D-Flash block for direct access (DFPART) and a partition within the buffer RAM for EEE use (ERPART):

- Validate the DFPART and ERPART values provided:
  - DFPART <= 128 (maximum number of 256 byte sectors in D-Flash block)
  - ERPART <= 16 (maximum number of 256 byte sectors in buffer RAM)
  - If ERPART > 0, 128 - DFPART >= 12 (minimum number of 256 byte sectors in the D-Flash block required to support EEE)
  - If ERPART > 0, ((128-DFPART)/ERPART) >= 8 (minimum ratio of D-Flash EEE space to buffer RAM EEE space to support EEE)
- Erase verify the D-Flash block and the EEE nonvolatile information register
- Program DFPART to the EEE nonvolatile information register at global address 0x12_0000 (see Table 27-7)

29.4.1.2.1 Define FCCOB Contents

The FCCOB parameter fields must be loaded with all required parameters for the Flash command being executed. Access to the FCCOB parameter fields is controlled via the CCOBIX bits in the FCCOBIX register (see Section 29.3.2.3).

The contents of the FCCOB parameter fields are transferred to the Memory Controller when the user clears the CCIF command completion flag in the FSTAT register (writing 1 clears the CCIF to 0). The CCIF flag will remain clear until the Flash command has completed. Upon completion, the Memory Controller will return CCIF to 1 and the FCCOB register will be used to communicate any results. The flow for a generic command write sequence is shown in Figure 29-26.

A.6.2 Oscillator

Table A-24. Oscillator Characteristics

Conditions are shown in Table A-4. unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1a	C	Crystal oscillator range (loop controlled Pierce)	f_{OSC}	4.0	—	16	MHz
1b	C	Crystal oscillator range (full swing Pierce) (1),(2)	f_{OSC}	2.0	—	40	MHz
2	P	Startup Current	i_{OSC}	100	—	—	µA
3a	C	Oscillator start-up time (LCP, 4MHz)(3)	t_{UPOSC}	—	2	10	ms
3b	C	Oscillator start-up time (LCP, 8MHz)3	t_{UPOSC}	—	1.6	8	ms
3c	C	Oscillator start-up time (LCP, 16MHz)3	t_{UPOSC}	—	1	5	ms
4a	C	Oscillator start-up time (full swing Pierce, 2MHz)3	t_{UPOSC}	—	8	40	ms
4b	C	Oscillator start-up time (full swing Pierce, 4MHz)3	t_{UPOSC}	—	4	20	ms
4c	C	Oscillator start-up time (full swing Pierce, 8MHz)3	t_{UPOSC}	—	2	10	ms
4d	C	Oscillator start-up time (full swing Pierce, 16MHz)3	t_{UPOSC}	—	1	5	ms
4e	C	Oscillator start-up time (full swing Pierce, 40MHz)3	t_{UPOSC}	—	0.8	4	ms
5	D	Clock Quality check time-out	t_{CQOUT}	0.45	—	2.5	s
6	P	Clock Monitor Failure Assert Frequency	f_{CMFA}	200	400	1000	KHz
7	P	External square wave input frequency	f_{EXT}	2.0	—	50	MHz
8	D	External square wave pulse width low	t_{EXTL}	9.5	—	—	ns
9	D	External square wave pulse width high	t_{EXTH}	9.5	—	—	ns
10	D	External square wave rise time	t_{EXTR}	—	—	1	ns
11	D	External square wave fall time	t_{EXTF}	—	—	1	ns
12	D	Input Capacitance (EXTAL, XTAL pins)	C_{IN}	—	7	—	pF
13	P	EXTAL Pin Input High Voltage	$V_{IH,EXTAL}$	$0.75*V_{DDPLL}$	—	—	V
	T	EXTAL Pin Input High Voltage,(4)	$V_{IH,EXTAL}$	—	—	V_{DDPLL} + 0.3	V
14	P	EXTAL Pin Input Low Voltage	$V_{IL,EXTAL}$	—	—	$0.25*V_{DDPLL}$	V
	T	EXTAL Pin Input Low Voltage,4	$V_{IL,EXTAL}$	V_{SSPLL} - 0.3	—	—	V
15	C	EXTAL Pin Input Hysteresis	$V_{HYS,EXTAL}$	—	180	—	mV
16	C	EXTAL Pin oscillation amplitude (loop controlled Pierce)	$V_{PP,EXTAL}$	—	0.9	—	V

1. Depending on the crystal a damping series resistor might be necessary
2. Only valid if full swing Pierce oscillator/external clock mode is selected
3. These values apply for carefully designed PCB layouts with capacitors that match the crystal/resonator requirements..
4. Only applies if EXTAL is externally driven