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, LINbus, SCI, SPI
Peripherals	LCD, Motor control PWM, POR, PWM, WDT
Number of I/O	117
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)	2.35V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12xhz256cag

1.5.2.4 System Wait Mode

This mode is entered when the CPU executes the WAI instruction. In this mode the CPU will not execute instructions. The internal CPU clock is switched off. All peripherals and the XGATE can be active in system wait mode. For further power consumption savings, the peripherals can individually turn off their local clocks. Asserting $\overline{\text{RESET}}$, $\overline{\text{XIRQ}}$, $\overline{\text{IRQ}}$ or any other interrupt that has not been masked ends system wait mode.

1.5.3 Freeze Mode

The enhanced capture timer, pulse width modulator, analog-to-digital converter, the periodic interrupt timer and the XGATE module provide a software programmable option to freeze the module status during the background debug module is active. This is useful when debugging application software. For detailed description of the behavior of the ATD, ECT, PWM, XGATE and PIT when the background debug module is active consult the corresponding module block description chapters.

1.6 Resets and Interrupts

Consult the S12XCPU block description chapter for information on exception processing.

1.6.1 Vectors

Table 1-9 lists all interrupt sources and vectors in the default order of priority. The interrupt module (S12XINT) provides an interrupt vector base register (IVBR) to relocate the vectors. Associated with each I-bit maskable service request is a configuration register. It selects if the service request is enabled, the service request priority level and whether the service request is handled either by the S12X CPU or by the XGATE module.

Table 2-2. S12XHZPIM Memory Map (continued)

Address Offset	Use	Access
0x0208	Port S I/O Register (PTS)	R/W
0x0209	Port S Input Register (PTIS)	R
0x020A	Port S Data Direction Register (DDRS)	R/W
0x020B	Port S Reduced Drive Register (RDRS)	R/W
0x020C	Port S Pull Device Enable Register (PERS)	R/W
0x020D	Port S Polarity Select Register (PPSS)	R/W
0x020E	Port S Wired-OR Mode Register (WOMS)	R/W
0x020F	Port S Slew Rate Register (SRRS)	R/W
0x0210	Port M I/O Register (PTM)	R/W
0x0211	Port M Input Register (PTIM)	R
0x0212	Port M Data Direction Register (DDRM)	R/W
0x0213	Port M Reduced Drive Register (RDRM)	R/W
0x0214	Port M Pull Device Enable Register (PERM)	R/W
0x0215	Port M Polarity Select Register (PPSM)	R/W
0x0216	Port M Wired-OR Mode Register (WOMM)	R/W
0x0217	Port M Slew Rate Register (SRRM)	R/W
0x0218	Port P I/O Register (PTP)	R/W
0x0219	Port P Input Register (PTIP)	R
0x021A	Port P Data Direction Register (DDRP)	R/W
0x021B	Port P Reduced Drive Register (RDRP)	R/W
0x021C	Port P Pull Device Enable Register (PERP)	R/W
0x021D	Port P Polarity Select Register (PPSP)	R/W
0x021E	Port P Wired-OR Mode Register (WOMP)	R/W
0x021F	Port P Slew Rate Register (SRRP)	R/W
0x0220 - 0x022F	Reserved	—
0x0230	Port L I/O Register (PTL)	R/W
0x0231	Port L Input Register (PTIL)	R
0x0232	Port L Data Direction Register (DDRL)	R/W
0x0233	Port L Reduced Drive Register (RDRL)	R/W
0x0234	Port L Pull Device Enable Register (PERL)	R/W
0x0235	Port L Polarity Select Register (PPSL)	R/W
0x0236	Reserved	—
0x0237	Port L Slew Rate Register (SRRL)	R/W
0x0238	Port U I/O Register (PTU)	R/W
0x0239	Port U Input Register (PTIU)	R
0x023A	Port U Data Direction Register (DDRU)	R/W
0x023B	Port U Slew Rate Register (SRRU)	R/W
0x023C	Port U Pull Device Enable Register (PERU)	R/W
0x023D	Port U Polarity Select Register (PPSU)	R/W
0x023E - 0x023F	Reserved	—

Table 2-3. DDRA Field Descriptions

Field	Description
7:0 DDRA[7:0]	**Data Direction Port A** 0 Associated pin is configured as input. 1 Associated pin is configured as output.

2.3.1.4 Port B Data Direction Register (DDRB)

Module Base + 0x0055

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

Figure 2-5. Port B Data Direction Register (DDRB)

Read: Anytime. Write: Anytime.

This register configures port pins PB[7:0] as either input or output.If a LCD frontplane driver is enabled (and LCD module is enabled), it outputs an analog signal to the corresponding pin and the associated Data Direction Register bit has no effect. If a LCD frontplane driver is disabled (or LCD module is disabled), the corresponding Data Direction Register bit reverts to control the I/O direction of the associated pin.

Table 2-4. DDRB Field Descriptions

Field	Description
7:0 DDRB[7:0]	**Data Direction Port B** 0 Associated pin is configured as input. 1 Associated pin is configured as output.

2.3.4 Port K

Port K pins can be used for either general-purpose I/O, or the liquid crystal display (LCD) driver, or the external address bus outputs ADDR22-ADDR16 muxed with master access output ACC2-ACC0 and instruction pipe signals IQSTAT3-IQSTAT0, or inputs $\overline{\text{EWAIT}}$ and ROMCTL. Refer to the LCD block description chapter for information on enabling and disabling the LCD and its frontplane drivers. Refer to the S12X_EBI block description chapter for information on external bus.

2.3.4.1 Port K I/O Register (PTK)

Module Base + 0x0051

	7	6	5	4	3	2	1	0
R W	PTK7	PTK6	PTK5	PTK4	PTK3	PTK2	PTK1	PTK0
XEBI:	ROMCTL[1] or $\overline{\text{EWAIT}}$	ADDR22 or ACC2	ADDR21 or ACC1	ADDR20 or ACC0	ADDR19 or IQSTAT3	ADDR18 or IQSTAT2	ADDR17 or IQSTAT1	ADDR16 or IQSTAT0
LCD:	FP23				BP3	BP2	BP1	BP0
Reset	0	0	0	0	0	0	0	0

Figure 2-12. Port K I/O Register (PTK)

1 Function active when $\overline{\text{RESET}}$ asserted.

Read: Anytime. Write: Anytime.

If the associated data direction bit (DDRKx) is set to 1 (output), a read returns the value of the I/O register bit.

If the associated data direction bit (DDRKx) is set to 0 (input) and the LCD frontplane driver is enabled (and LCD module is enabled), the associated I/O register bit (PTKx) reads “1”.

If the associated data direction bit (DDRKx) is set to 0 (input) and the LCD frontplane driver is disabled (or LCD module is disabled), a read returns the value of the pin.

2.3.4.2 Port K Data Direction Register (DDRK)

Module Base + 0x0055

	7	6	5	4	3	2	1	0
R W	DDRK7	DDRK6	DDRK5	DDRK4	DDRK3	DDRK2	DDRK1	DDRK0
Reset	0	0	0	0	0	0	0	0

Figure 2-13. Port K Data Direction Register (DDRK)

Read: Anytime. Write: Anytime.

3.4.2.1 Erase Verify Command

The erase verify operation will verify that a Flash block is erased.

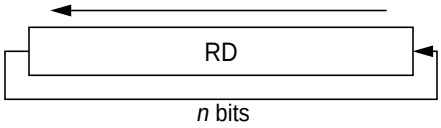
An example flow to execute the erase verify operation is shown in Figure 3-25. The erase verify command write sequence is as follows:

1. Write an aligned word to a valid address in the Flash array memory to start the command write sequence for the erase verify command. The address and data written will be ignored. Multiple Flash blocks can be simultaneously erase verified by writing to the same relative address in each Flash block.
2. Write the erase verify command, 0x05, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the erase verify command.

After launching the erase verify command, the CCIF flag in the FSTAT register will set after the operation has completed unless a new command write sequence has been buffered. The number of bus cycles required to execute the erase verify operation is equal to the number of addresses in a Flash block plus 14 bus cycles as measured from the time the CBEIF flag is cleared until the CCIF flag is set. Upon completion of the erase verify operation, the BLANK flag in the FSTAT register will be set if all addresses in the selected Flash blocks are verified to be erased. If any address in a selected Flash block is not erased, the erase verify operation will terminate and the BLANK flag in the FSTAT register will remain clear. The MRDS bits in the FTSTMOD register will determine the sense-amp margin setting during the erase verify operation.

ROL — Rotate Left — ROL

Operation



n = RS or IMM4

Rotates the bits in register RD *n* positions to the left. The lower *n* bits of the register RD are filled with the upper *n* bits. Two source forms are available. In the first form, the parameter *n* is contained in the instruction code as an immediate operand. In the second form, the parameter is contained in the lower bits of the source register RS[3:0]. All other bits in RS are ignored. If *n* is zero, no shift will take place and the register RD will be unaffected; however, the condition code flags will be updated.

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
ROL RD, #IMM4	IMM4	0	0	0	0	1	RD	IMM4		1	1	1	0	P
ROL RD, RS	DYA	0	0	0	0	1	RD	RS	1	0	1	1	0	P

7.3.2 Register Descriptions

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

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x_00 SYNR	R	0	0	SYN5	SYN4	SYN3	SYN2	SYN1	SYN0
	W								
0x_01 REFDV	R	0	0	REFDV5	REFDV4	REFDV3	REFDV2	REFDV1	REFDV0
	W								
0x_02 CTFLG	R	0	0	0	0	0	0	0	0
	W								
0x_03 CRGFLG	R	RTIF	PORF	LVRF	LOCKIF	LOCK	TRACK	SCMIF	SCM
	W								
0x_04 CRGINT	R	RTIE	ILAF	0	LOCKIE	0	0	SCMIE	0
	W								
0x_05 CLKSEL	R	PLLSEL	PSTP	0	0	PLLWAI	0	RTIWAI	COPWAI
	W								
0x_06 PLLCTL	R	CME	PLLON	AUTO	ACQ	FSTWKP	PRE	PCE	SCME
	W								
0x_07 RTICTL	R	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
	W								
0x_08 COPCTL	R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
	W			WRTMASK					
0x_09 FORBYP	R	0	0	0	0	0	0	0	0
	W								
0x_0A CTCTL	R	1	0	0	0	0	0	0	0
	W								
0x_0B ARMCOP	R	0	0	0	0	0	0	0	0
	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

= Unimplemented or Reserved

Figure 7-3. S12CRGV6 Register Summary

Table 7-8. RTI Frequency Divide Rates for RTDEC = 1

RTR[3:0]	RTR[6:4] =							
	000 ($1x10^3$)	**001 ($2x10^3$)**	**010 ($5x10^3$)**	**011 ($10x10^3$)**	**100 ($20x10^3$)**	**101 ($50x10^3$)**	**110 ($100x10^3$)**	**111 ($200x10^3$)**
0000 (÷1)	$1x10^3$	$2x10^3$	$5x10^3$	$10x10^3$	$20x10^3$	$50x10^3$	$100x10^3$	$200x10^3$
0001 (÷2)	$2x10^3$	$4x10^3$	$10x10^3$	$20x10^3$	$40x10^3$	$100x10^3$	$200x10^3$	$400x10^3$
0010 (÷3)	$3x10^3$	$6x10^3$	$15x10^3$	$30x10^3$	$60x10^3$	$150x10^3$	$300x10^3$	$600x10^3$
0011 (÷4)	$4x10^3$	$8x10^3$	$20x10^3$	$40x10^3$	$80x10^3$	$200x10^3$	$400x10^3$	$800x10^3$
0100 (÷5)	$5x10^3$	$10x10^3$	$25x10^3$	$50x10^3$	$100x10^3$	$250x10^3$	$500x10^3$	$1x10^6$
0101 (÷6)	$6x10^3$	$12x10^3$	$30x10^3$	$60x10^3$	$120x10^3$	$300x10^3$	$600x10^3$	$1.2x10^6$
0110 (÷7)	$7x10^3$	$14x10^3$	$35x10^3$	$70x10^3$	$140x10^3$	$350x10^3$	$700x10^3$	$1.4x10^6$
0111 (÷8)	$8x10^3$	$16x10^3$	$40x10^3$	$80x10^3$	$160x10^3$	$400x10^3$	$800x10^3$	$1.6x10^6$
1000 (÷9)	$9x10^3$	$18x10^3$	$45x10^3$	$90x10^3$	$180x10^3$	$450x10^3$	$900x10^3$	$1.8x10^6$
1001 (÷10)	$10\ x10^3$	$20x10^3$	$50x10^3$	$100x10^3$	$200x10^3$	$500x10^3$	$1x10^6$	$2x10^6$
1010 (÷11)	$11\ x10^3$	$22x10^3$	$55x10^3$	$110x10^3$	$220x10^3$	$550x10^3$	$1.1x10^6$	$2.2x10^6$
1011 (÷12)	$12x10^3$	$24x10^3$	$60x10^3$	$120x10^3$	$240x10^3$	$600x10^3$	$1.2x10^6$	$2.4x10^6$
1100 (÷13)	$13x10^3$	$26x10^3$	$65x10^3$	$130x10^3$	$260x10^3$	$650x10^3$	$1.3x10^6$	$2.6x10^6$
1101 (÷14)	$14x10^3$	$28x10^3$	$70x10^3$	$140x10^3$	$280x10^3$	$700x10^3$	$1.4x10^6$	$2.8x10^6$
1110 (÷15)	$15x10^3$	$30x10^3$	$75x10^3$	$150x10^3$	$300x10^3$	$750x10^3$	$1.5x10^6$	$3x10^6$
1111 (÷16)	$16x10^3$	$32x10^3$	$80x10^3$	$160x10^3$	$320x10^3$	$800x10^3$	$1.6x10^6$	$3.2x10^6$

10.4.5.3 1/2 Duty Multiplexed with 1/3 Bias Mode

Duty = 1/2:DUTY1 = 1, DUTY0 = 0

Bias = 1/3:BIAS = 1

V_0 = VSSX, V_1 = VLCD * 1/3, V_2 = VLCD * 2/3, V_3 = VLCD

- BP2 and BP3 are not used, a maximum of 64 segments are displayed.

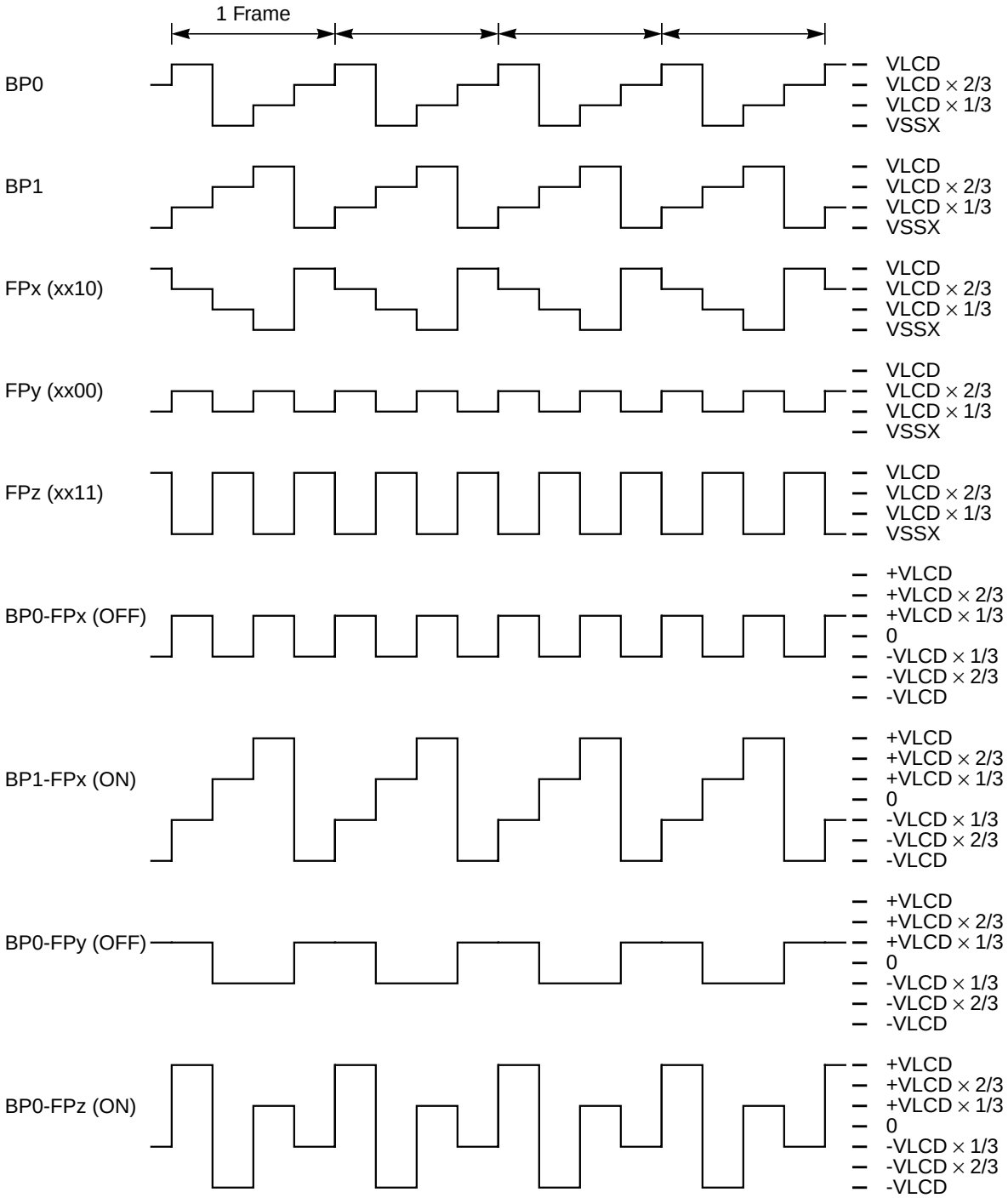


Figure 10-11. 1/2 Duty and 1/3 Bias

11.4.2 PWM Duty Cycle

The PWM duty cycle for the motor controller channel x can be determined by dividing the decimal representation of bits D[10:0] in MCDCx by the decimal representation of the bits P[10:0] in MCPER and multiplying the result by 100% as shown in the equation below:

$$\text{Effective PWM Channel X \% Duty Cycle} = \frac{\text{DUTY}}{\text{MCPER}} \cdot 100\%$$

NOTE

x = PWM Channel Number = 0, 1, 2, 3 ... 11. This equation is only valid if DUTY <= MCPER and MCPER is not equal to 0.

Whenever D[10:0] >= P[10:0], a constant low level (RECIRC = 0) or high level (RECIRC = 1) will be output.

11.4.3 Motor Controller Counter Clock Source

Figure 11-22 shows how the PWM motor controller timer counter clock source is selected.

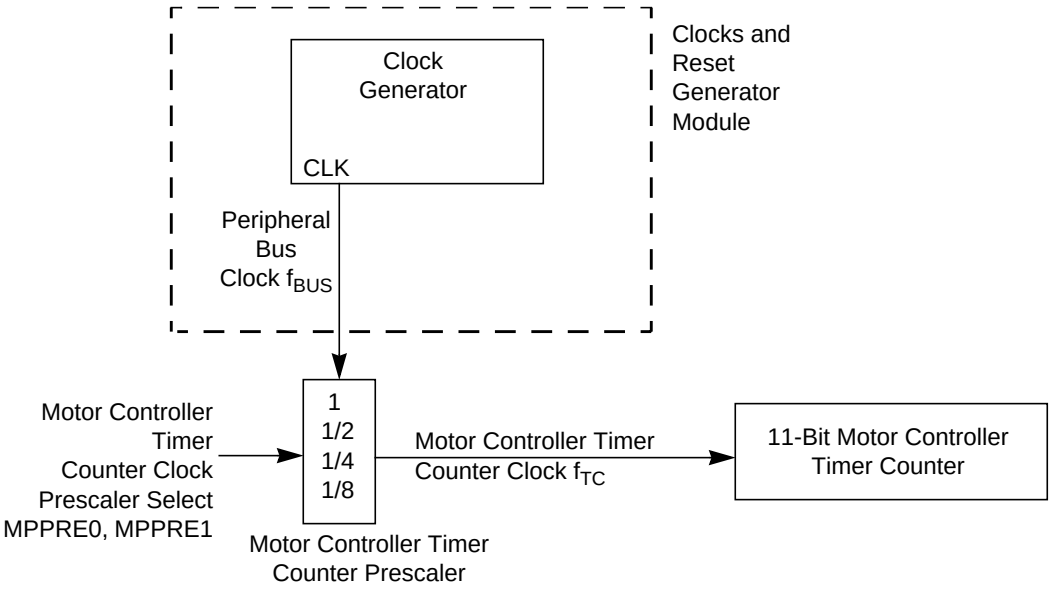


Figure 11-22. Motor Controller Counter Clock Selection

The peripheral bus clock is the source for the motor controller counter prescaler. The motor controller counter clock rate, f_{TC}, is set by selecting the appropriate prescaler value. The prescaler is selected with the MCPRE[1:0] bits in motor controller control register 0 (MCCTL0). The motor controller channel frequency of operation can be calculated using the following formula if DITH = 0:

$$\text{Motor Channel Frequency (Hz)} = \frac{f_{TC}}{\text{MCPER} \cdot \text{M}}$$

Module Base + 0x000C to Module Base + 0x000D — Access: User read/write(1)

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

= Unimplemented

Figure 14-16. MSCAN Reserved Register

1. Read: Always reads zero in normal system operation modes
   Write: Unimplemented in normal system operation modes

NOTE

Writing to this register when in special system operating modes can alter the MSCAN functionality.

14.3.2.14 MSCAN Miscellaneous Register (CANMISC)

This register provides additional features.

Module Base + 0x000D — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	BOHOLD
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-17. MSCAN Miscellaneous Register (CANMISC)

1. Read: Anytime
   Write: Anytime; write of '1' clears flag; write of '0' ignored

Table 14-21. CANMISC Register Field Descriptions

Field	Description
0 BOHOLD	**Bus-off State Hold Until User Request** — If BORM is set in MSCAN Control Register 1 (CANCTL1) this bit indicates whether the module has entered the bus-off state. Clearing this bit requests the recovery from bus-off. Refer to Section 14.5.2, "Bus-Off Recovery," for details. 0 Module is not bus-off or recovery has been requested by user in bus-off state 1 Module is bus-off and holds this state until user request

14.3.2.15 MSCAN Receive Error Counter (CANRXERR)

This register reflects the status of the MSCAN receive error counter.

Module Base + 0x00X2

	7	6	5	4	3	2	1	0
R W								
Reset:	x	x	x	x	x	x	x	x

= Unused; always read 'x'

Figure 14-32. Identifier Register 2 — Standard Mapping

Module Base + 0x00X3

	7	6	5	4	3	2	1	0
R W								
Reset:	x	x	x	x	x	x	x	x

= Unused; always read 'x'

Figure 14-33. Identifier Register 3 — Standard Mapping

14.3.3.2 Data Segment Registers (DSR0-7)

The eight data segment registers, each with bits DB[7:0], contain the data to be transmitted or received. The number of bytes to be transmitted or received is determined by the data length code in the corresponding DLR register.

Module Base + 0x00X4 to Module Base + 0x00XB

	7	6	5	4	3	2	1	0
R W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Reset:	x	x	x	x	x	x	x	x

Figure 14-34. Data Segment Registers (DSR0–DSR7) — Extended Identifier Mapping

Table 14-33. DSR0–DSR7 Register Field Descriptions

Field	Description
7-0 DB[7:0]	**Data bits 7-0**

Table 16-2. $\overline{SS}$ Input / Output Selection

MODFEN	SSOE	Master Mode	Slave Mode
0	0	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
0	1	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
1	0	$\overline{SS}$ input with MODF feature	$\overline{SS}$ input
1	1	$\overline{SS}$ is slave select output	$\overline{SS}$ input

16.3.2.2 SPI Control Register 2 (SPICR2)

Module Base +0x0001

	7	6	5	4	3	2	1	0
R	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 16-4. SPI Control Register 2 (SPICR2)

Read: Anytime

Write: Anytime; writes to the reserved bits have no effect

Table 16-3. SPICR2 Field Descriptions

Field	Description
4 MODFEN	**Mode Fault Enable Bit** — This bit allows the MODF failure to be detected. If the SPI is in master mode and MODFEN is cleared, then the $\overline{SS}$ port pin is not used by the SPI. In slave mode, the $\overline{SS}$ is available only as an input regardless of the value of MODFEN. For an overview on the impact of the MODFEN bit on the $\overline{SS}$ port pin configuration, refer to Table 16-4. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 $\overline{SS}$ port pin is not used by the SPI. 1 $\overline{SS}$ port pin with MODF feature.
3 BIDIROE	**Output Enable in the Bidirectional Mode of Operation** — This bit controls the MOSI and MISO output buffer of the SPI, when in bidirectional mode of operation (SPC0 is set). In master mode, this bit controls the output buffer of the MOSI port, in slave mode it controls the output buffer of the MISO port. In master mode, with SPC0 set, a change of this bit will abort a transmission in progress and force the SPI into idle state. 0 Output buffer disabled. 1 Output buffer enabled.
1 SPISWAI	**SPI Stop in Wait Mode Bit** — This bit is used for power conservation while in wait mode. 0 SPI clock operates normally in wait mode. 1 Stop SPI clock generation when in wait mode.
0 SPC0	**Serial Pin Control Bit 0** — This bit enables bidirectional pin configurations as shown in Table 16-4. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.

16.4.3 Transmission Formats

During an SPI transmission, data is transmitted (shifted out serially) and received (shifted in serially) simultaneously. The serial clock (SCK) synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows selection of an individual slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. Optionally, on a master SPI device, the slave select line can be used to indicate multiple-master bus contention.

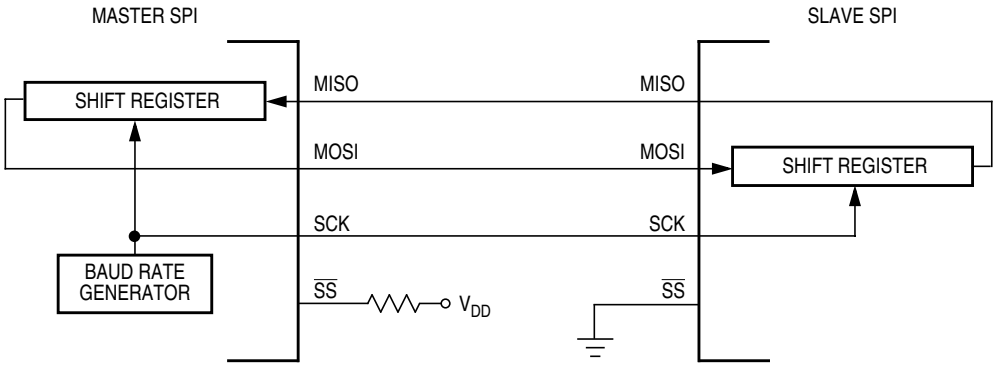


Figure 16-10. Master/Slave Transfer Block Diagram

16.4.3.1 Clock Phase and Polarity Controls

Using two bits in the SPI control register 1, software selects one of four combinations of serial clock phase and polarity.

The CPOL clock polarity control bit specifies an active high or low clock and has no significant effect on the transmission format.

The CPHA clock phase control bit selects one of two fundamentally different transmission formats.

Clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

16.4.3.2 CPHA = 0 Transfer Format

The first edge on the SCK line is used to clock the first data bit of the slave into the master and the first data bit of the master into the slave. In some peripherals, the first bit of the slave's data is available at the slave's data out pin as soon as the slave is selected. In this format, the first SCK edge is issued a half cycle after $\overline{SS}$ has become low.

A half SCK cycle later, the second edge appears on the SCK line. When this second edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the shift register, depending on LSBFE bit.

After this second edge, the next bit of the SPI master data is transmitted out of the serial data output pin of the master to the serial input pin on the slave. This process continues for a total of 16 edges on the SCK line, with data being latched on odd numbered edges and shifted on even numbered edges.

Either left aligned or center aligned output mode can be used in concatenated mode and is controlled by the low order CAEx bit. The high order CAEx bit has no effect.

Table 18-11 is used to summarize which channels are used to set the various control bits when in 16-bit mode.

Table 18-11. 16-bit Concatenation Mode Summary

CONxx	PWMEx	PPOLx	PCLKx	CAEx	PWMx Output
CON67	PWME7	PPOL7	PCLK7	CAE7	PWM7
CON45	PWME5	PPOL5	PCLK5	CAE5	PWM5
CON23	PWME3	PPOL3	PCLK3	CAE3	PWM3
CON01	PWME1	PPOL1	PCLK1	CAE1	PWM1

18.4.2.8 PWM Boundary Cases

Table 18-12 summarizes the boundary conditions for the PWM regardless of the output mode (left aligned or center aligned) and 8-bit (normal) or 16-bit (concatenation).

Table 18-12. PWM Boundary Cases

PWMDTYx	PWMPERx	PPOLx	PWMx Output
$00 (indicates no duty)	>$00	1	Always low
$00 (indicates no duty)	>$00	0	Always high
XX	$00[1] (indicates no period)	1	Always high
XX	$00[1] (indicates no period)	0	Always low
>= PWMPERx	XX	1	Always high
>= PWMPERx	XX	0	Always low

[1] Counter = $00 and does not count.

18.5 Resets

The reset state of each individual bit is listed within the Section 18.3.2, “Register Descriptions” which details the registers and their bit-fields. All special functions or modes which are initialized during or just following reset are described within this section.

- The 8-bit up/down counter is configured as an up counter out of reset.
- All the channels are disabled and all the counters do not count.

19.4.3 Interrupts

This section describes interrupts originated by the ECT block. The MCU must service the interrupt requests. Table 19-39 lists the interrupts generated by the ECT to communicate with the MCU.

Table 19-39. ECT Interrupts

Interrupt Source	Description
Timer channel 7–0	Active high timer channel interrupts 7–0
Modulus counter underflow	Active high modulus counter interrupt
Pulse accumulator B overflow	Active high pulse accumulator B interrupt
Pulse accumulator A input	Active high pulse accumulator A input interrupt
Pulse accumulator A overflow	Pulse accumulator overflow interrupt
Timer overflow	Timer 0verflow interrupt

The ECT only originates interrupt requests. The following is a description of how the module makes a request and how the MCU should acknowledge that request. The interrupt vector offset and interrupt number are chip dependent.

19.4.3.1 Channel [7:0] Interrupt

This active high output will be asserted by the module to request a timer channel 7–0 interrupt to be serviced by the system controller.

19.4.3.2 Modulus Counter Interrupt

This active high output will be asserted by the module to request a modulus counter underflow interrupt to be serviced by the system controller.

19.4.3.3 Pulse Accumulator B Overflow Interrupt

This active high output will be asserted by the module to request a timer pulse accumulator B overflow interrupt to be serviced by the system controller.

19.4.3.4 Pulse Accumulator A Input Interrupt

This active high output will be asserted by the module to request a timer pulse accumulator A input interrupt to be serviced by the system controller.

19.4.3.5 Pulse Accumulator A Overflow Interrupt

This active high output will be asserted by the module to request a timer pulse accumulator A overflow interrupt to be serviced by the system controller.

- Destination address of RTI, RTS, and RTC instructions.
- Vector address of interrupts, except for SWI and BDM vectors

LBRA, BRA, BSR, BGND as well as non-indexed JMP, JSR, and CALL instructions are not classified as change of flow and are not stored in the trace buffer.

COF addresses are defined as follows for the XGATE:

- Source address of taken conditional branches
- Destination address of indexed JAL instructions.
- First XGATE code address in a thread

Change-of-flow addresses stored include the full 23-bit address bus of CPU12X, the 16-bit address bus for the XGATE module and an information byte, which contains a source/destination bit to indicate whether the stored address was a source address or destination address.

NOTE

When an CPU12X COF instruction with destination address is executed, the destination address is stored to the trace buffer on instruction completion, indicating the COF has taken place. If an interrupt occurs simultaneously then the next instruction carried out is actually from the interrupt service routine. The instruction at the destination address of the original program flow gets exectuted after the interrupt service routine.

In the following example an IRQ interrupt occurs during execution of the indexed JMP at address MARK1. The BRN at the destination (SUB_1) is not executed until after the IRQ service routine but the destination address is entered into the trace buffer to indicate that the indexed JMP COF has taken place.

```
        LDX     #SUB_1
MARK1   JMP     0,X                     ; IRQ interrupt occurs during execution of this
MARK2   NOP                             ;

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                 ;
```

Chapter 25
Memory Mapping Control (S12XMMCV3)

Version Number	Revision Date	Effective Date	Author	Description of Changes
v03.00	25 May 2005	05/25/2005		Generic S12XMMC BlockGuide is meant for S12X derivatives. - Added FLEXRAY IP like a Master Block. - Major Cleanup,. - Added condional texts to different configurations. - Added Internal section.
v03.01	21 July 2005	07/21/2005		Clarify in details External Spaces accesses and firmware in single chip modes Update reviewed wording

25.1 Introduction

This section describes the functionality of the module mapping control (MMC) sub-block of the S12X platform. The block diagram of the MMC is shown in Figure 25-1.

The MMC module controls the multi-master priority accesses, the selection of internal resources and external space. Internal buses, including internal memories and peripherals, are controlled in this module. The local address space for each master is translated to a global memory space.

Table 25-2. External Input Signals Associated with the MMC

Signal	I/O	Description	Availability
MODC	I	Mode input	Latched after RESET (active low)
MODB	I	Mode input	Latched after RESET (active low)
MODA	I	Mode input	Latched after RESET (active low)
EROMCTL	I	EROM control input	Latched after RESET (active low)
ROMCTL	I	ROM control input	Latched after RESET (active low)

Table 25-3. External Output Signals Associated with the MMC

Signal	I/O	Description	Available in Modes					
			NS	SS	NX	ES	EX	ST
CS0	O	Chip select line 0	(see Table 25-4)					
CS1	O	Chip select line 1						
CS2	O	Chip select line 2						
CS3	O	Chip select line 3						

Appendix E
Detailed Register Map

The following tables show the detailed register map of the MC9S12XHZ512.

0x0000–0x0009 Port Integration Module (PIM) Map 1 of 5

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0000	PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA 0
		W								
0x0001	PORTB	R	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
		W								
0x0002	DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
0x0003	DDRB	R	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		W								
0x0004	PORTC	R	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		W								
0x0005	PORTD	R	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
		W								
0x0006	DDRC	R	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		W								
0x0007	DDRD	R	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		W								
0x0008	PORTE	R	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
		W								
0x0009	DDRE	R	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	0	0
		W								

0x000A–0x000B Module Mapping Control (S12XMMC) Map 1 of 4

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x000A	MMCCTL0	R	0	0	0	0	0	CS2E	CS1E	CS0E
		W								
0x000B	MODE	R	MODC	MODB	MODA	0	0	0	0	0
		W								

0x000C–0x000D Port Integration Module (PIM) Map 2 of 5

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x000C	PUCR	R	PUPKE	BKPUE	0	PUPEE	PUPDE	PUPCE	PUPBE	PUPAE
		W								
0x000D	RDRIV	R	RDPK	0	0	RDPE	RDPD	RDPC	RDPB	RDPA
		W								