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	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, EBI/EMI, I²C, SCI, SPI
Peripherals	LCD, Motor control PWM, POR, PWM, WDT
Number of I/O	85
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	6K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12hz128val

1.6 System Clock Description

The clock and reset generator (CRG) provides the internal clock signals for the core and all peripheral modules. Figure 1-12 shows the clock connections from the CRG to all modules.

Consult the CRG block description chapter for details on clock generation.

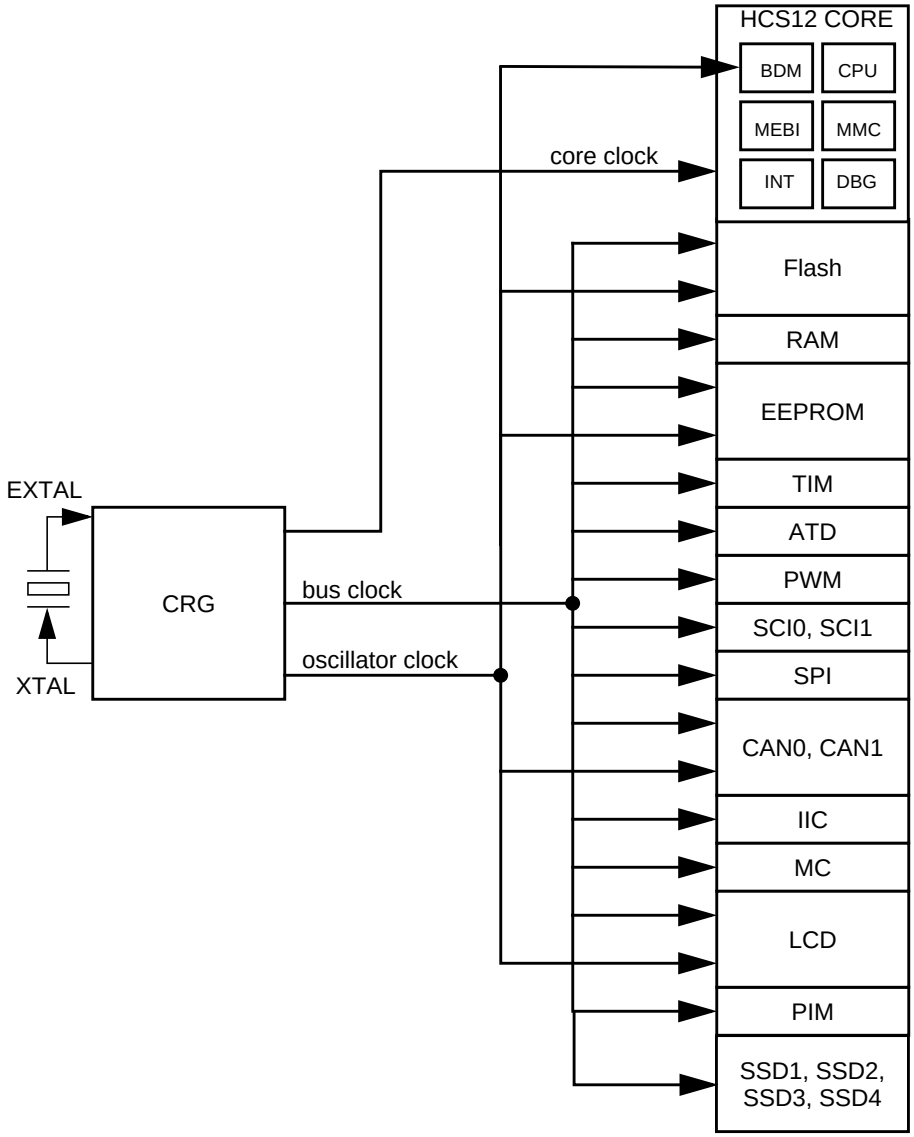

Figure 1-12. Clock Connections

1.10.2 Resets

Resets are a subset of the interrupts featured in Table 1-12. The different sources capable of generating a system reset are summarized in Table 1-12.

Table 1-12. Reset Summary

Reset	Priority	Source	Vector
Power-on Reset	1	CRG Module	0xFFFE, 0xFFFF
External Reset	1	RESET pin	0xFFFE, 0xFFFF
Low Voltage Reset	1	VREG Module	0xFFFE, 0xFFFF
Clock Monitor Reset	2	CRG Module	0xFFFC, 0xFFFD
COP Watchdog Reset	3	CRG Module	0xFFFA, 0xFFFB

1.10.3 Effects of Reset

When a reset occurs, MCU registers and control bits are changed to known start-up states. Refer to the respective module block description chapters for register reset states. MC mode dependent pin configuration of port A, B and E out of reset.

Refer to the PIM block description chapter for reset configurations of all peripheral module ports.

Refer to Table 1-1 for locations of the memories depending on the operating mode after reset.

The RAM array is not automatically initialized out of reset.

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

= Unimplemented or Reserved

Figure 2-12. Flash Command Register (FCMD - NVM User Mode)

All CMDB bits are readable and writable during a command write sequence while bit 7 reads 0 and is not writable.

Table 2-17. FCMD Field Descriptions

Field	Description
6-0 CMDB[6:0]	**Flash Command** — Valid Flash commands are shown in Table 2-18. Writing any command other than those listed in Table 2-18 sets the ACCERR flag in the FSTAT register.

Table 2-18. Valid Flash Command List

CMDB[6:0]	NVM Command
0x05	Erase Verify
0x06	Data Compress
0x20	Word Program
0x40	Sector Erase
0x41	Mass Erase
0x47	Sector Erase Abort

2.3.2.9 Flash Control Register (FCTL)

The banked FCTL register is the Flash control register.

	7	6	5	4	3	2	1	0
R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
W								
Reset	F	F	F	F	F	F	F	F

= Unimplemented or Reserved

Figure 2-13. Flash Control Register (FCTL)

All bits in the FCTL register are readable but are not writable.

The FCTL register is loaded from the Flash Configuration Field byte at $FF0E during the reset sequence, indicated by F in Figure 2-13.

Table 2-19. FCTL Field Descriptions

Field	Description
7-0 NV[7:0]	**Nonvolatile Bits** — The NV[7:0] bits are available as nonvolatile bits. Refer to the Device User Guide for proper use of the NV bits.

2.6.1 Unsecuring the MCU using Backdoor Key Access

The MCU may be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor keys (four 16-bit words programmed at addresses 0xFF00–0xFF07). If the KEYEN[1:0] bits are in the enabled state (see Section 2.3.2.2, “Flash Security Register (FSEC)”) and the KEYACC bit is set, a write to a backdoor key address in the Flash memory triggers a comparison between the written data and the backdoor key data stored in the Flash memory. If all four words of data are written to the correct addresses in the correct order and the data matches the backdoor keys stored in the Flash memory, the MCU will be unsecured. The data must be written to the backdoor keys sequentially starting with 0xFF00–0xFF01 and ending with 0xFF06–0xFF07. 0x0000 and 0xFFFF are not permitted as backdoor keys. While the KEYACC bit is set, reads of the Flash memory will return invalid data.

The user code stored in the Flash memory must have a method of receiving the backdoor key from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 2.3.2.2, “Flash Security Register (FSEC)”), the MCU can be unsecured by the backdoor access sequence described below:

1. Set the KEYACC bit in the Flash configuration register (FCNFG).
2. Write the correct four 16-bit words to Flash addresses 0xFF00–0xFF07 sequentially starting with 0xFF00.
3. Clear the KEYACC bit.
4. If all four 16-bit words match the backdoor keys stored in Flash addresses 0xFF00–0xFF07, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 1:0.

The backdoor key access sequence is monitored by an internal security state machine. An illegal operation during the backdoor key access sequence will cause the security state machine to lock, leaving the MCU in the secured state. A reset of the MCU will cause the security state machine to exit the lock state and allow a new backdoor key access sequence to be attempted. The following operations during the backdoor key access sequence will lock the security state machine:

1. If any of the four 16-bit words does not match the backdoor keys programmed in the Flash array.
2. If the four 16-bit words are written in the wrong sequence.
3. If more than four 16-bit words are written.
4. If any of the four 16-bit words written are 0x0000 or 0xFFFF.
5. If the KEYACC bit does not remain set while the four 16-bit words are written.
6. If any two of the four 16-bit words are written on successive MCU clock cycles.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the Flash security byte can be programmed to the unsecure state, if desired.

In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0xFF00–0xFF07 in the Flash configuration field.

The security as defined in the Flash security byte (0xFF0F) is not changed by using the backdoor key access sequence to unsecure. The backdoor keys stored in addresses 0xFF00–0xFF07 are unaffected by the backdoor key access sequence. After the next reset of the MCU, the security state of the Flash module

3.4.1.3 Valid EEPROM Commands

Table 3-10 summarizes the valid EEPROM commands. Also shown are the effects of the commands on the EEPROM array.

Table 3-10. Valid EEPROM Commands

ECMD	Meaning	Function on EEPROM Array
0x05	Erase Verify	Verify all memory bytes of the EEPROM array are erased. If the array is erased, the BLANK bit will set in the ESTAT register upon command completion.
0x20	Program	Program a word (two bytes).
0x40	Sector Erase	Erase two words (four bytes) of EEPROM array.
0x41	Mass Erase	Erase all of the EEPROM array. A mass erase of the full array is only possible when EPDIS and EPOPEN are set.
0x60	Sector Modify	Erase two words of EEPROM, re-program one word.

An EEPROM word must be in an erased state before being programmed.
Cumulative programming of bits within a word is not allowed.

The sector modify command (0x60) is a two-step command which first erases a sector (2 words) of the EEPROM array and then re-programs one of the words in that sector. The EEPROM sector which is erased by the sector modify command is the sector containing the address of the aligned array write which starts the valid command sequence. That same address is re-programmed with the data which is written. By launching a sector modify command and then pipelining a program command it is possible to completely replace the contents of an EEPROM sector.

3.4.1.4 Illegal EEPROM Operations

The ACCERR flag will be set during the command write sequence if any of the following illegal operations are performed causing the command write sequence to immediately abort:

1. Writing to the EEPROM address space before initializing ECLKDIV.
2. Writing a misaligned word or a byte to the valid EEPROM address space.
3. Writing to the EEPROM address space while CBEIF is not set.
4. Writing a second word to the EEPROM address space before executing a program or erase command on the previously written word.
5. Writing to any EEPROM register other than ECMD after writing a word to the EEPROM address space.
6. Writing a second command to the ECMD register before executing the previously written command.
7. Writing an invalid command to the ECMD register in normal mode.
8. Writing to any EEPROM register other than ESTAT (to clear CBEIF) after writing to the command register (ECMD).

4.3.4.4 Port P Reduced Drive Register (RDRP)

Figure 4-26. Port P Reduced Drive Register (RDRP)

Read:Anytime. Write:Anytime.

This register configures the drive strength of configured output pins as either full or reduced. If a pin is configured as input, the corresponding Reduced Drive Register bit has no effect.

Table 4-19. RDRP Field Descriptions

Field	Description
5:0 RDRP[5:0]	**Reduced Drive Port P** 0 Full drive strength at output. 1 Associated pin drives at about 1/3 of the full drive strength.

4.4 Functional Description

Each pin associated with ports AD, L, P, S, T, U and V can act as general-purpose I/O. In addition the pin can act as an output from a peripheral module or an input to a peripheral module.

A set of configuration registers is common to all ports. All registers can be written at any time, however a specific configuration might not become active.

> Example: Selecting a pull-up resistor. This resistor does not become active while the port is used as a push-pull output.

4.4.1 I/O Register

The I/O Register holds the value driven out to the pin if the port is used as a general-purpose I/O. Writing to the I/O Register only has an effect on the pin if the port is used as general-purpose output.

When reading the I/O Register, the value of each pin is returned if the corresponding Data Direction Register bit is set to 0 (pin configured as input). If the data direction register bits is set to 1, the content of the I/O Register bit is returned. This is independent of any other configuration (Figure 4-55).

Due to internal synchronization circuits, it can take up to 2 bus cycles until the correct value is read on the I/O Register when changing the data direction register.

4.4.2 Input Register

The Input Register is a read-only register and generally returns the value of the pin (Figure 4-55). It can be used to detect overload or short circuit conditions.

Due to internal synchronization circuits, it can take up to 2 bus cycles until the correct value is read on the Input Register when changing the Data Direction Register.

4.4.3 Data Direction Register

The Data Direction Register defines whether the pin is used as an input or an output. A Data Direction Register bit set to 0 configures the pin as an input. A Data Direction Register bit set to 0 configures the pin as an output. If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 4-55).

4.4.6 Polarity Select Register

The Polarity Select Register selects either a pull-up or pull-down device if enabled. The pull device becomes active only if the pin is used as an input or as a wired-or output.

4.4.7 Pin Configuration Summary

The following table summarizes the effect of various configuration in the Data Direction (DDR), Input/Output (I/O), reduced drive (RDR), Pull Enable (PE), Pull Select (PS) and Interrupt Enable (IE) register bits. The PS configuration bit is used for two purposes:

1. Configure the sensitive interrupt edge (rising or falling), if interrupt is enabled.
2. Select either a pull-up or pull-down device if PE is set to “1”.

Table 4-40. Pin Configuration Summary

DDR	IO	RDR	PE	PS	IE[1]	Function[2]	Pull Device	Interrupt
0	X	X	0	X	0	Input	Disabled	Disabled
0	X	X	1	0	0	Input	Pull Up	Disabled
0	X	X	1	1	0	Input	Pull Down	Disabled
0	X	X	0	0	1	Input	Disabled	Falling Edge
0	X	X	0	1	1	Input	Disabled	Rising Edge
0	X	X	1	0	1	Input	Pull Up	Falling Edge
0	X	X	1	1	1	Input	Pull Down	Rising Edge
1	0	0	X	X	0	Output to 0, Full Drive	Disabled	Disabled
1	1	0	X	X	0	Output to 1, Full Drive	Disabled	Disabled
1	0	1	X	X	0	Output to 0, Reduced Drive	Disabled	Disabled
1	1	1	X	X	0	Output to 1, Reduced Drive	Disabled	Disabled
1	0	0	X	0	1	Output to 0, Full Drive	Disabled	Falling Edge
1	1	0	X	1	1	Output to 1, Full Drive	Disabled	Rising Edge
1	0	1	X	0	1	Output to 0, Reduced Drive	Disabled	Falling Edge
1	1	1	X	1	1	Output to 1, Reduced Drive	Disabled	Rising Edge

1 Applicable only on Port AD.

2 Digital outputs are disabled and digital input logic is forced to “1” when an analog module associated with the port is enabled.

writes (0x0055 or 0x00AA) to the ARMCOP register must occur in the last 25% of the selected time-out period. A premature write the CRG will immediately generate a reset.

As soon as the reset sequence is completed the reset generator checks the reset condition. If no clock monitor failure is indicated and the latched state of the COP timeout is true, processing begins by fetching the COP vector.

5.5.3 Power-On Reset, Low Voltage Reset

The on-chip voltage regulator detects when V_{DD} to the MCU has reached a certain level and asserts power-on reset or low voltage reset or both. As soon as a power-on reset or low voltage reset is triggered the CRG performs a quality check on the incoming clock signal. As soon as clock quality check indicates a valid oscillator clock signal the reset sequence starts using the oscillator clock. If after 50 check windows the clock quality check indicated a non-valid oscillator clock the reset sequence starts using self-clock mode.

Figure 5-26 and Figure 5-27 show the power-up sequence for cases when the $\overline{\text{RESET}}$ pin is tied to V_{DD} and when the $\overline{\text{RESET}}$ pin is held low.

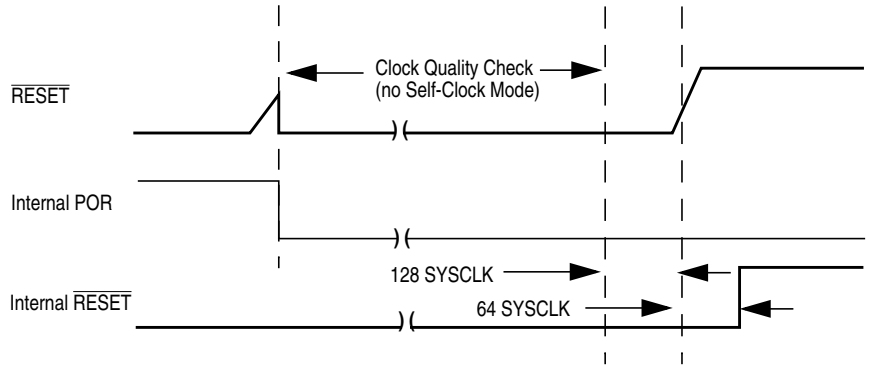

Figure 5-26. $\overline{\text{RESET}}$ Pin Tied to V_{DD} (by a Pull-Up Resistor)

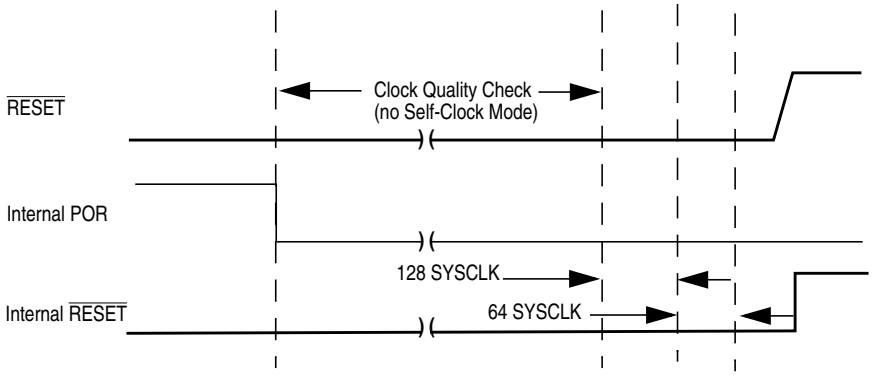

Figure 5-27. $\overline{\text{RESET}}$ Pin Held Low Externally

8.4 Functional Description

This section provides a complete functional description of the LCD32F4B block, detailing the operation of the design from the end user perspective in a number of subsections.

8.4.1 LCD Driver Description

8.4.1.1 Frontplane, Backplane, and LCD System During Reset

During a reset the following conditions exist:

- The LCD32F4B system is configured in the default mode, 1/4 duty and 1/3 bias, that means all backplanes are used.
- All frontplane enable bits, FP[31:0]EN are cleared and the ON/OFF control for the display, the LCDEN bit is cleared, thereby forcing all frontplane and backplane driver outputs to the high impedance state. The MCU pin state during reset is defined by the port integration module (PIM).

8.4.1.2 LCD Clock and Frame Frequency

The frequency of the oscillator clock (OSCCLK) and divider determine the LCD clock frequency. The divider is set by the LCD clock prescaler bits, LCLK[2:0], in the LCD control register 0 (LCDCR0). Table 8-7 shows the LCD clock and frame frequency for some multiplexed mode at OSCCLK = 16 MHz, 8 MHz, 4 MHz, 2 MHz, 1 MHz, and 0.5 MHz.

Table 8-7. LCD Clock and Frame Frequency

Oscillator Frequency in MHz	LCD Clock Prescaler			Divider	LCD Clock Frequency [Hz]	Frame Frequency [Hz]			
	LCLK2	LCLK1	LCLK0			1/1 Duty	1/2 Duty	1/3 Duty	1/4 Duty
OSCCLK = 0.5	0 0	0 0	0 1	1024 2048	488 244	488 244	244 122	163 81	122 61
OSCCLK = 1.0	0 0	0 1	1 0	2048 4096	488 244	488 244	244 122	163 81	122 61
OSCCLK = 2.0	0 0	1 1	0 1	4096 8192	488 244	488 244	244 122	163 81	122 61
OSCCLK = 4.0	0 1	1 0	1 0	8192 16384	488 244	488 244	244 122	163 81	122 61
OSCCLK = 8.0	1 1	0 0	0 1	16384 32768	488 244	488 244	244 122	163 81	122 61
OSCCLK = 16.0	1 1	1 1	0 1	65536 131072	244 122	244 122	122 61	81 40	61 31

For other combinations of OSCCLK and divider not shown in Table 8-7, the following formula may be used to calculate the LCD frame frequency for each multiplex mode:

$$\text{LCD Frame Frequency (Hz)} = \left[\frac{\text{OSCCLK (Hz)}}{\text{Divider}}\right] \cdot \text{Duty}$$

The possible divider values are shown in Table 8-7.

Chapter 10
Stepper Stall Detector (SSDV1)

10.1 Introduction

The stepper stall detector (SSD) block provides a circuit to measure and integrate the induced voltage on the non-driven coil of a stepper motor using full steps when the gauge pointer is returning to zero (RTZ). During the RTZ event, the pointer is returned to zero using full steps in either clockwise or counter clockwise direction, where only one coil is driven at any point in time. The back electromotive force (EMF) signal present on the non-driven coil is integrated after a blanking time, and its results stored in a 16-bit accumulator. The 16-bit modulus down counter can be used to monitor the blanking time and the integration time. The value in the accumulator represents the change in linked flux (magnetic flux times the number of turns in the coil) and can be compared to a stored threshold. Values above the threshold indicate a moving motor, in which case the pointer can be advanced another full step in the same direction and integration be repeated. Values below the threshold indicate a stalled motor, thereby marking the cessation of the RTZ event. The SSD is capable of multiplexing two stepper motors.

10.1.1 Modes of Operation

- Return to zero modes
  - Blanking with no drive
  - Blanking with drive
  - Conversion
  - Integration
- Low-power modes

10.1.2 Features

- Programmable full step state
- Programmable integration polarity
- Blanking (recirculation) state
- 16-bit integration accumulator register
- 16-bit modulus down counter with interrupt
- Multiplex two stepper motors

In slave transmitter routine, the received acknowledge bit (RXAK) must be tested before transmitting the next byte of data. Setting RXAK means an 'end of data' signal from the master receiver, after which it must be switched from transmitter mode to receiver mode by software. A dummy read then releases the SCL line so that the master can generate a STOP signal.

11.7.1.7 Arbitration Lost

If several masters try to engage the bus simultaneously, only one master wins and the others lose arbitration. The devices which lost arbitration are immediately switched to slave receive mode by the hardware. Their data output to the SDA line is stopped, but SCL continues to be generated until the end of the byte during which arbitration was lost. An interrupt occurs at the falling edge of the ninth clock of this transfer with IBAL=1 and MS/SL=0. If one master attempts to start transmission while the bus is being engaged by another master, the hardware will inhibit the transmission; switch the MS/SL bit from 1 to 0 without generating STOP condition; generate an interrupt to CPU and set the IBAL to indicate that the attempt to engage the bus is failed. When considering these cases, the slave service routine should test the IBAL first and the software should clear the IBAL bit if it is set.

Table 12-13. CANTFLG Register Field Descriptions

Field	Description
2:0 TXE[2:0]	**Transmitter Buffer Empty** — This flag indicates that the associated transmit message buffer is empty, and thus not scheduled for transmission. The CPU must clear the flag after a message is set up in the transmit buffer and is due for transmission. The MSCAN sets the flag after the message is sent successfully. The flag is also set by the MSCAN when the transmission request is successfully aborted due to a pending abort request (see Section 12.3.2.9, "MSCAN Transmitter Message Abort Request Register (CANTARQ)"). If not masked, a transmit interrupt is pending while this flag is set. Clearing a TXEx flag also clears the corresponding ABTAKx (see Section 12.3.2.10, "MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)"). When a TXEx flag is set, the corresponding ABTRQx bit is cleared (see Section 12.3.2.9, "MSCAN Transmitter Message Abort Request Register (CANTARQ)"). When listen-mode is active (see Section 12.3.2.2, "MSCAN Control Register 1 (CANCTL1)") the TXEx flags cannot be cleared and no transmission is started. Read and write accesses to the transmit buffer will be blocked, if the corresponding TXEx bit is cleared (TXEx = 0) and the buffer is scheduled for transmission. 0 The associated message buffer is full (loaded with a message due for transmission) 1 The associated message buffer is empty (not scheduled)

12.3.2.8 MSCAN Transmitter Interrupt Enable Register (CANTIER)

This register contains the interrupt enable bits for the transmit buffer empty interrupt flags.

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	TXEIE2	TXEIE1	TXEIE0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 12-9. MSCAN Transmitter Interrupt Enable Register (CANTIER)

NOTE

The CANTIER register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Read: Anytime
Write: Anytime when not in initialization mode

Table 12-14. CANTIER Register Field Descriptions

Field	Description
2:0 TXEIE[2:0]	**Transmitter Empty Interrupt Enable** 0 No interrupt request is generated from this event. 1 A transmitter empty (transmit buffer available for transmission) event causes a transmitter empty interrupt request.

14.3.2.4 SPI Status Register (SPISR)

	7	6	5	4	3	2	1	0
R	SPIF	0	SPTEF	MODF	0	0	0	0
W								
Reset	0	0	1	0	0	0	0	0

= Unimplemented or Reserved

Figure 14-6. SPI Status Register (SPISR)

Read: anytime

Write: has no effect

Table 14-8. SPISR Field Descriptions

Field	Description
7 SPIF	**SPIF Interrupt Flag** — This bit is set after a received data byte has been transferred into the SPI Data Register. This bit is cleared by reading the SPISR register (with SPIF set) followed by a read access to the SPI Data Register. 0 Transfer not yet complete 1 New data copied to SPIDR
5 SPTEF	**SPI Transmit Empty Interrupt Flag** — If set, this bit indicates that the transmit data register is empty. To clear this bit and place data into the transmit data register, SPISR has to be read with SPTEF = 1, followed by a write to SPIDR. Any write to the SPI Data Register without reading SPTEF = 1, is effectively ignored. 0 SPI Data register not empty 1 SPI Data register empty
4 MODF	**Mode Fault Flag** — This bit is set if the $\overline{SS}$ input becomes low while the SPI is configured as a master and mode fault detection is enabled, MODFEN bit of SPICR2 register is set. Refer to MODFEN bit description in Section 14.3.2.2, “SPI Control Register 2 (SPICR2).” The flag is cleared automatically by a read of the SPI Status Register (with MODF set) followed by a write to the SPI Control Register 1. 0 Mode fault has not occurred. 1 Mode fault has occurred.

14.3.2.5 SPI Data Register (SPIDR)

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

= Unimplemented or Reserved

Figure 14-7. SPI Data Register (SPIDR)

Read: anytime; normally read only after SPIF is set

Write: anytime

15.3.2 Register Descriptions

The following paragraphs describe in detail all the registers and register bits in the PWM8B6C module.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
PWME	R	0	0	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0
	W								
PWMPOL	R	0	0	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0
	W								
PWMCLK	R	0	0	PCLK5	PCLK4	PCLK3	PCLK2	PCLK1	PCLK0
	W								
PWMPRCLK	R	0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0
	W								
PWMCAE	R	0	0	CAE5	CAE4	CAE2	CAE2	CAE1	CAE0
	W								
PWMCTL	R	0	CON45	CON23	CON01	PSWAI	PFRZ	0	0
	W								
PWMTST	R	0	0	0	0	0	0	0	0
	W								
PWMPRSC	R	0	0	0	0	0	0	0	0
	W								
PWMSCLA	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
PWMSCLB	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
PWMSCNTA	R	0	0	0	0	0	0	0	0
	W								
PWMSCNTB	R	0	0	0	0	0	0	0	0
	W								
PWMCNT0	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT1	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT2	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 15-2. PWM Register Summary

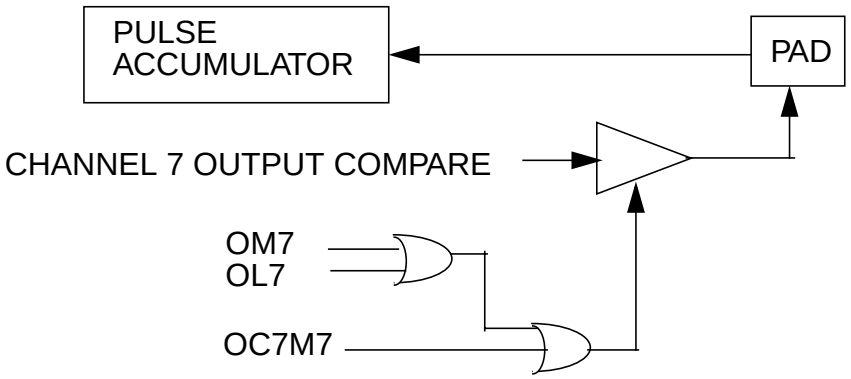

Figure 16-4. Channel 7 Output Compare/Pulse Accumulator Logic

NOTE

For more information see the respective functional descriptions in Section 16.4, “Functional Description,” of this document.

16.2 External Signal Description

The TIM16B8C module has a total of eight external pins.

16.2.1 IOC7 — Input Capture and Output Compare Channel 7 Pin

This pin serves as input capture or output compare for channel 7. This can also be configured as pulse accumulator input.

16.2.2 IOC6 — Input Capture and Output Compare Channel 6 Pin

This pin serves as input capture or output compare for channel 6.

16.2.3 IOC5 — Input Capture and Output Compare Channel 5 Pin

This pin serves as input capture or output compare for channel 5.

16.2.4 IOC4 — Input Capture and Output Compare Channel 4 Pin

This pin serves as input capture or output compare for channel 4. Pin

16.2.5 IOC3 — Input Capture and Output Compare Channel 3 Pin

This pin serves as input capture or output compare for channel 3.

[4] Write has no effect; return 0 on read

16.3.2 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 TIOS	R	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0
	W								
0x0001 CFORC	R	0	0	0	0	0	0	0	0
	W	FOC7	FOC6	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0
0x0002 OC7M	R	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0
	W								
0x0003 OC7D	R	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0
	W								
0x0004 TCNTH	R	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8
	W								
0x0005 TCNTL	R	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
	W								
0x0006 TSCR2	R	TEN	TSWAI	TSFRZ	TFFCA	0	0	0	0
	W								
0x0007 TTOV	R	TOV7	TOV6	TOV5	TOV4	TOV3	TOV2	TOV1	TOV0
	W								
0x0008 TCTL1	R	OM7	OL7	OM6	OL6	OM5	OL5	OM4	OL4
	W								
0x0009 TCTL2	R	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
	W								
0x000A TCTL3	R	EDG7B	EDG7A	EDG6B	EDG6A	EDG5B	EDG5A	EDG4B	EDG4A
	W								
0x000B TCTL4	R	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
	W								

= Unimplemented or Reserved

Figure 16-5. TIM16B8C Register Summary

19.3.2.10 Debug Comparator A Register (DBGCA)

	15	14	13	12	11	10	9	8
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 19-17. Debug Comparator A Register High (DBGCAH)

	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 19-18. Debug Comparator A Register Low (DBGCAL)

Table 19-21. DBGCA Field Descriptions

Field	Description
15:0 15:0	**Comparator A Compare Bits** — The comparator A compare bits control whether comparator A compares the address bus bits [15:0] to a logic 1 or logic 0. See Table 19-20. 0 Compare corresponding address bit to a logic 0 1 Compare corresponding address bit to a logic 1

19.3.2.11 Debug Comparator B Extended Register (DBGCBX)

	7	6	5	4	3	2	1	0
R W	PAGSEL		EXTCMP					
Reset	0	0	0	0	0	0	0	0

Figure 19-19. Debug Comparator B Extended Register (DBGCBX)

Table 19-22. DBGCBX Field Descriptions

Field	Description
7:6 PAGSEL	**Page Selector Field** — If DBGEN is set in DBGC1, then PAGSEL selects the type of paging as shown in Table 19-11. DPAGE and EPAGE are not yet implemented so the value in bit 7 will be ignored (i.e., PAGSEL values of 10 and 11 will be interpreted as values of 00 and 01, respectively.) In BKP mode, PAGSEL has no meaning and EXTCMP[5:0] are compared to address bits [19:14] if the address is in the FLASH/ROM memory space.
5:0 EXTCMP	**Comparator B Extended Compare Bits** — The EXTCMP bits are used as comparison address bits as shown in Table 19-11 along with the appropriate PPAGE, DPAGE, or EPAGE signal from the core. Also see Table 19-20.

Appendix F
Ordering Information

Figure F-1 provides an ordering example for the MC9S12HZ256.

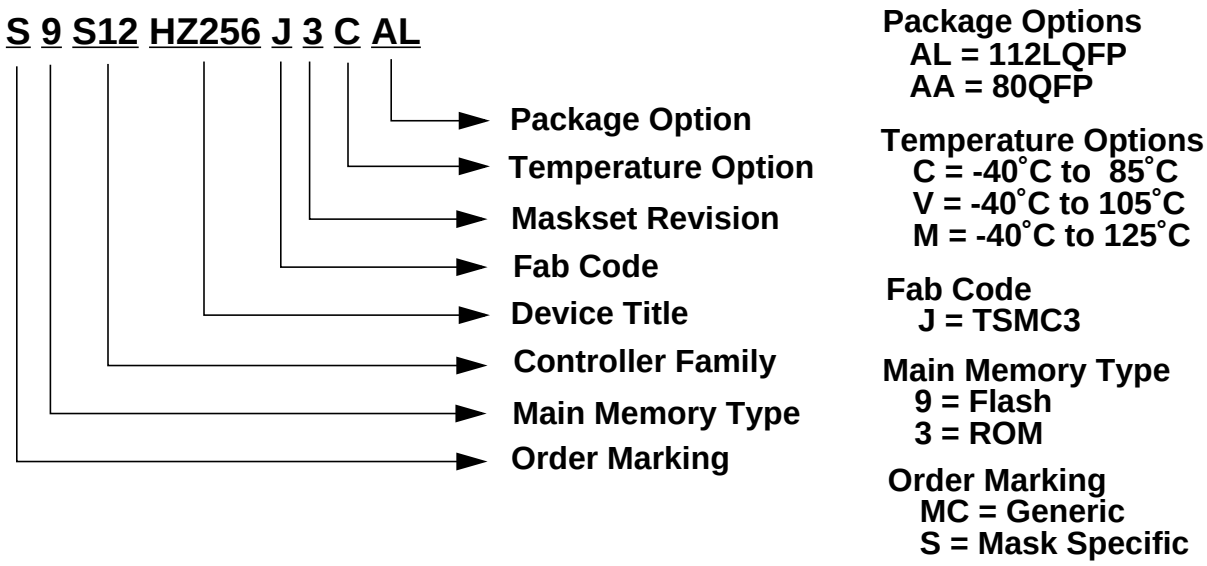

Figure F-1. Order Part Number Coding

Customers who place orders using the generic MC part numbers which are constructed using the above rules will automatically receive the preferred maskset. If the product is updated in the future and a newer maskset is put into production, then the newer maskset may automatically ship against these generic MC part numbers.

If required, a customer can specify a particular maskset when ordering product. Orders placed against a "S" part number will only receive one specific maskset. If a new maskset is made available, customers will be notified by PCN (Process Change Notification) but will have to order against a different "S" part number in order to receive the new maskset.