



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	2MHz
Connectivity	-
Peripherals	LVD, POR, PWM
Number of I/O	5
Program Memory Size	1.5KB (1.5K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-SOIC (0.209", 5.30mm Width)
Supplier Device Package	8-SO
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mcl908qt1cdwer

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong





15.3 Mo	onitor Module (MON)
15.3.1	Functional Description
15.3.1.1	Normal Monitor Mode
15.3.1.2	Forced Monitor Mode
15.3.1.3	Monitor Vectors
15.3.1.4	Data Format
15.3.1.5	Break Signal
15.3.1.6	Baud Rate
15.3.1.7	Commands
15.3.2	Security

Chapter 16 **Electrical Specifications**

16.1	Introduction
16.2	Absolute Maximum Ratings
16.3	Functional Operating Range
16.4	Thermal Characteristics
16.5	DC Electrical Characteristics
16.6	Control Timing
16.7	Typical 3.0-V Output Drive Characteristics
16.8	Oscillator Characteristics
16.9	Supply Current Characteristics 157
16.10	Analog-to-Digital (ADC) Converter Characteristics 159
16.10.1	ADC Electrical Operating Conditions 159
16.10.2	2 ADC Performance Characteristics 159
16.11	Timer Interface Module Characteristics 160
16.12	Memory Characteristics

Chapter 17 Ordering Information and Mechanical Specifications

17.1	Introduction	163
17.2	MC Order Numbers	163
17.3	Package Dimensions	163



Chapter 2 Memory

2.1 Introduction

The central processor unit (CPU08) can address 64 Kbytes of memory space. The memory map, shown in Figure 2-1, includes:

- 4096 bytes of user FLASH for MC68HLC908QT4 and MC68HLC908QY4
- 1536 bytes of user FLASH for MC68HLC908QT2, MC68HLC908QT1, MC68HLC908QY2, and MC68HLC908QY1
- 128 bytes of random access memory (RAM)
- 48 bytes of user-defined vectors, located in FLASH
- 416 bytes of monitor read-only memory (ROM)
- 1536 bytes of FLASH program and erase routines, located in ROM

2.2 Unimplemented Memory Locations

Accessing an unimplemented location can have unpredictable effects on MCU operation. In Figure 2-1 and in register figures in this document, unimplemented locations are shaded.

2.3 Reserved Memory Locations

Accessing a reserved location can have unpredictable effects on MCU operation. In Figure 2-1 and in register figures in this document, reserved locations are marked with the word Reserved or with the letter R.



Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0			
\$0029	TIM Channel 1 Register High (TCH1H)	Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8			
	See page 133.	Reset:				Indetermina	te after reset						
\$002A	TIM Channel 1 Register Low (TCH1L)	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0			
	See page 133.	Reset:		Indeterminate after reset									
\$002B ↓ \$0035	Unimplemented												
		1			Γ				I	1			
\$0036	Oscillator Status Register (OSCSTAT)	Read: Write:	R	R	R	R	R	R	ECGON	ECGST			
	See page 95.	Reset:	0	0	0	0	0	0	0	0			
\$0037	Unimplemented	Read:											
\$0038	Oscillator Trim Register (OSCTRIM) See page 96.	Read: Write:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0			
	1.0	Reset:	1	0	0	0	0	0	0	0			
\$0039 ↓ \$003B	Unimplemented												
				T	Γ	T	T		I				
	ADC Status and Control	Read:	0000	AIEN	ADCO	CH4	СНЗ	CH2	CH1	CH0			
\$003C	Register (ADSCR)	Write:	R										
	occ page 40.	Reset:	0	0	0	1	1	1	1	1			
\$003D	Unimplemented												
		I		1		1	1		1				
\$003E	ADC Data Register (ADR)	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0			
	See page 44.	Reset:				Indetermina	te after reset						
\$003F	ADC Input Clock Register (ADICLK)	Read: Write:	ADIV2	ADIV1	ADIV0	0	0	0	0	0			
	See page 45.	Reset:	0	0	0	0	0	0	0	0			
				= Unimplem	ented	R	R = Reserved U = Unaffected						

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 6)



Input/Output (I/O) Section

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$FE00	Break Status Register (BSR)	Read: Write:	R	R	R	R	R	R	SBSW See note 1	R
	See page 139.	Reset:	1. Writing a () clears SBS\	N.				0	
	SIM Reset Status Register	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
\$FE01	(SRSR)	Write:								
	See page 117.	POR:	1	0	0	0	0	0	0	0
	Break Auxiliary	Read:	0	0	0	0	0	0	0	BDCOP
\$FE02	Register (BRKAR)	Write:								
	See page 139.	Reset:	0	0	0	0	0	0	0	0
\$FE03	Break Flag Control Register (BFCR)	Read: Write:	BCFE	R	R	R	R	R	R	R
	See page 139.	Reset:	0							
	Interrupt Status Register 1 (INT1) See page 77.	Read:	0	IF5	IF4	IF3	0	IF1	0	0
\$FE04		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
	Interrupt Status Register 2	Read:	IF14	0	0	0	0	0	0	0
\$FE05	(INT2)	Write:	R	R	R	R	R	R	R	R
	See page 77.	Reset:	0	0	0	0	0	0	0	0
	Interrupt Status Register 3	Read:	0	0	0	0	0	0	0	IF15
\$FE06	(INT3)	Write:	R	R	R	R	R	R	R	R
	See page 77.	Reset:	0	0	0	0	0	0	0	0
\$FE07	Reserved	ļ	R	R	R	R	R	R	R	R
		(,	
	FLASH Control Register	Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
\$FE08	(FLCR) See page 32.	Write:							<u> </u>	
	000 page 01.	Reset:	0	0	0	0	0	0	0	0
\$FE09	Break Address High Register (BRKH)	Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	See page 138.	Reset:	0	0	0	0	0	0	0	0
\$FE0A	Break Address low Register (BRKL)	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	See page 138.	Reset:	0	0	0	0	0	0	0	0
				= Unimplemented R = Reserved U = Unaffected					fected	

Figure 2-2. Control, Status, and Data Registers (Sheet 5 of 6)



Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$FE0B	Break Status and Control Register (BRKSCR) See page 138.	Read:	BBKE	BBKA	0	0	0	0	0	0
		Write:	DITIL	2.101						
		Reset:	0	0	0	0	0	0	0	0
		Read:	LVIOUT	0	0	0	0	0	0	R
\$FE0C	LVI Status Register (LVISR) See page 87.	Write:								
	000 page on	Reset:	0	0	0	0	0	0	0	0
\$FE0D ↓ \$FE0F	Reserved for FLASH Test	Í	R	R	R	R	R	R	R	R

\$FFBE	FLASH Block Protect Register (FLBPR)	Read: Write:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0		
	See page 37.	Reset:		Unaffected by reset								
\$FFBF	Reserved		R	R	R	R	R	R	R	R		
		•										
\$FFC0	Internal Oscillator Trim Value (Optional)	Read: Write:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0		
	(-F)	Reset: Unaffected by reset										
\$FFC1	Reserved		R	R	R	R	R	R	R	R		
		•		•								







2.6 FLASH Memory (FLASH)

This subsection describes the operation of the embedded FLASH memory. The FLASH memory can be read, programmed, and erased from a single external supply. The program and erase operations are enabled through the use of an internal charge pump.

The FLASH memory consists of an array of 4096 or 1536 bytes with an additional 48 bytes for user vectors. The minimum size of FLASH memory that can be erased is 64 bytes; and the maximum size of FLASH memory that can be programmed in a program cycle is 32 bytes (a row). Program and erase operations are facilitated through control bits in the FLASH control register (FLCR). Details for these operations appear later in this section. The address ranges for the user memory and vectors are:

- \$EE00 \$FDFF; user memory, 4096 bytes: MC68HLC908QY4 and MC68HLC908QT4
- \$F800 \$FDFF; user memory, 1536 bytes: MC68HLC908QY2, MC68HLC908QT2, MC68HLC908QY1 and MC68HLC908QT1
- \$FFD0 \$FFFF; user interrupt vectors, 48 bytes.

NOTE

An erased bit reads as a 1 and a programmed bit reads as a 0. A security feature prevents viewing of the FLASH contents.⁽¹⁾

2.6.1 FLASH Control Register

The FLASH control register (FLCR) controls FLASH program and erase operations.



Figure 2-3. FLASH Control Register (FLCR)

HVEN — High Voltage Enable Bit

This read/write bit enables high voltage from the charge pump to the memory for either program or erase operation. It can only be set if either PGM =1 or ERASE =1 and the proper sequence for program or erase is followed.

1 = High voltage enabled to array and charge pump on

0 = High voltage disabled to array and charge pump off

^{1.} No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.



2.6.3 FLASH Mass Erase Operation

Use the following procedure to erase the entire FLASH memory to read as a 1:

- 1. Set both the ERASE bit and the MASS bit in the FLASH control register.
- 2. Read the FLASH block protect register.
- 3. Write any data to any FLASH address⁽¹⁾ within the FLASH memory address range.
- 4. Wait for a time, t_{NVS} (minimum 10 μ s).
- 5. Set the HVEN bit.
- 6. Wait for a time, t_{MErase} (minimum 4 ms).
- 7. Clear the ERASE and MASS bits.

NOTE

Mass erase is disabled whenever any block is protected (FLBPR does not equal \$FF).

- 8. Wait for a time, t_{NVH} (minimum 100 μ s).
- 9. Clear the HVEN bit.
- 10. After time, t_{RCV} (typical 1 µs), the memory can be accessed in read mode again.

NOTE

Programming and erasing of FLASH locations cannot be performed by code being executed from the FLASH memory. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps.

CAUTION

A mass erase will erase the internal oscillator trim value at \$FFC0.

2.6.4 FLASH Program Operation

Programming of the FLASH memory is done on a row basis. A row consists of 32 consecutive bytes starting from addresses \$XX00, \$XX20, \$XX40, \$XX60, \$XX80, \$XXA0, \$XXC0, or \$XXE0. Use the following step-by-step procedure to program a row of FLASH memory

Figure 2-4 shows a flowchart of the programming algorithm.

NOTE

Only bytes which are currently \$FF may be programmed.

- 1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
- 2. Read the FLASH block protect register.
- 3. Write any data to any FLASH location within the address range desired.
- 4. Wait for a time, t_{NVS} (minimum 10 μ s).
- 5. Set the HVEN bit.
- 6. Wait for a time, t_{PGS} (minimum 5 μ s).
- 7. Write data to the FLASH address being $programmed^{(2)}$.

^{1.} When in monitor mode, with security sequence failed (see 15.3.2 Security), write to the FLASH block protect register instead of any FLASH address.

The time between each FLASH address change, or the time between the last FLASH address programmed to clearing PGM bit, must not exceed the maximum programming time, t_{PROG} maximum.



Central Processor Unit (CPU)

Source				Effect on CCR					ess	de	and	ŝS
Form	Operation	Description	v	н	1	N	z	С	lode	Dpcc	bera	ycle
PULA	Pull A from Stack	$SP \leftarrow (SP + 1)$: Pull (A)	_	_	_	_	_	_	NH	86	0	2
PULH	Pull H from Stack	$SP \leftarrow (SP + 1); Pull (H)$	-	-	-	-	-	-	INH	8A		2
PULX	Pull X from Stack	$SP \leftarrow (SP + 1); Pull (X)$	-	-	-	-	-	-	INH	88		2
ROL <i>opr</i> ROLA ROLX ROL <i>opr</i> ,X ROL ,X ROL <i>opr</i> ,SP	Rotate Left through Carry	□_C	ţ	_	_	ţ	ţ	ţ	DIR INH INH IX1 IX SP1	39 49 59 69 79 9E69	dd ff ff	4 1 4 3 5
ROR <i>opr</i> RORA RORX ROR <i>opr</i> ,X ROR ,X ROR <i>opr</i> ,SP	Rotate Right through Carry	b7 b0	ţ	_	_	ţ	ţ	ţ	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	4 1 4 3 5
RSP	Reset Stack Pointer	$SP \gets \$FF$	—	-	-	I	-	-	INH	9C		1
RTI	Return from Interrupt	$\begin{array}{l} SP \leftarrow (SP) + 1; \ Pull \ (CCR) \\ SP \leftarrow (SP) + 1; \ Pull \ (A) \\ SP \leftarrow (SP) + 1; \ Pull \ (X) \\ SP \leftarrow (SP) + 1; \ Pull \ (PCH) \\ SP \leftarrow (SP) + 1; \ Pull \ (PCL) \end{array}$	ţ	ţ	ţ	ţ	ţ	ţ	INH	80		7
RTS	Return from Subroutine	$\begin{array}{l} SP \leftarrow SP + 1; Pull \ (PCH) \\ SP \leftarrow SP + 1; Pull \ (PCL) \end{array}$	-	_	-	I	-	-	INH	81		4
SBC #opr SBC opr SBC opr SBC opr,X SBC opr,X SBC ,X SBC opr,SP SBC opr,SP	Subtract with Carry	$A \gets (A) - (M) - (C)$	ţ	_	_	ţ	ţ	ţ	IMM DIR EXT IX2 IX1 IX SP1 SP2	A2 B2 C2 D2 E2 F2 9EE2 9ED2	ii dd hh II ee ff ff ee ff	23443245
SEC	Set Carry Bit	C ← 1	-	-	-	-	-	1	INH	99		1
SEI	Set Interrupt Mask	l ← 1	-	-	1	-	-	-	INH	9B		2
STA opr STA opr STA opr,X STA opr,X STA ,X STA opr,SP STA opr,SP	Store A in M	M ← (A)	0	_	_	ţ	ţ	_	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh II ee ff ff ee ff	3 4 4 3 2 4 5
STHX opr	Store H:X in M	$(M{:}M+1) \gets (H{:}X)$	0	-	-	1	\$	-	DIR	35	dd	4
STOP	Enable Interrupts, Stop Processing, Refer to MCU Documentation	$I \leftarrow 0$; Stop Processing	-	_	0	1	-	-	INH	8E		1
STX opr STX opr STX opr,X STX opr,X STX,X STX,X STX opr,SP STX opr,SP	Store X in M	$M \gets (X)$	0	_	_	ţ	ţ	_	DIR EXT IX2 IX1 IX SP1 SP2	BF CF DF EF FF 9EEF 9EDF	dd hh II ee ff ff ee ff	3443245
SUB #opr SUB opr SUB opr SUB opr,X SUB opr,X SUB opr,SP SUB opr,SP	Subtract	A ← (A) – (M)	ţ	_	_	ţ	ţ	ţ	IMM DIR EXT IX2 IX1 IX SP1 SP2	A0 B0 C0 D0 E0 F0 9EE0 9ED0	ii dd hh II ee ff ff ee ff	23443245

Table 7-1. Instr	uction Set	Summary	(Sheet 5	of 6)
		· • • • • • • • • • • • • • • • • •	(,

Keyboard Interrupt Module (KBI)



RST, IRQ: Pins have internal (about 30K Ohms) pull up PTA[0:5]: High current sink and source capability

PTA[0:5]: Pins have programmable keyboard interrupt and pull up

PTB[0:7]: Not available on 8-pin devices – MC68HLC908QT1, MC68HLC908QT2, and MC68HLC908QT4 (see note in 12.1 Introduction)

ADC: Not available on the MC68HLC908QY1 and MC68HC9L08QT1

Figure 9-1. Block Diagram Highlighting KBI Block and Pins



Functional Description



Figure 9-2. Keyboard Interrupt Block Diagram

If the MODEK bit is set, the keyboard interrupt inputs are both falling edge and low-level sensitive, and both of the following actions must occur to clear a keyboard interrupt request:

- Vector fetch or software clear A vector fetch generates an interrupt acknowledge signal to clear the interrupt request. Software may generate the interrupt acknowledge signal by writing a 1 to the ACKK bit in the keyboard status and control register (KBSCR). The ACKK bit is useful in applications that poll the keyboard interrupt inputs and require software to clear the keyboard interrupt request. Writing to the ACKK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACKK does not affect subsequent transitions on the keyboard interrupt inputs. A falling edge that occurs after writing to the ACKK bit latches another interrupt request. If the keyboard interrupt mask bit, IMASKK, is clear, the central processor unit (CPU) loads the program counter with the vector address at locations \$FFE0 and \$FFE1.
- Return of all enabled keyboard interrupt inputs to logic 1 As long as any enabled keyboard interrupt pin is at logic 0, the keyboard interrupt remains set. The auto wakeup interrupt input, AWUIREQ, will be cleared only by writing to ACKK bit in KBSCR or reset.

The vector fetch or software clear and the return of all enabled keyboard interrupt pins to logic 1 may occur in any order.

If the MODEK bit is clear, the keyboard interrupt pin is falling-edge sensitive only. With MODEK clear, a vector fetch or software clear immediately clears the keyboard interrupt request.

Reset clears the keyboard interrupt request and the MODEK bit, clearing the interrupt request even if a keyboard interrupt input stays at logic 0.

The keyboard flag bit (KEYF) in the keyboard status and control register can be used to see if a pending interrupt exists. The KEYF bit is not affected by the keyboard interrupt mask bit (IMASKK) which makes it useful in applications where polling is preferred.



Keyboard Interrupt Module (KBI)

9.7.2 Keyboard Interrupt Enable Register

The port A keyboard interrupt enable register (KBIER) enables or disables each port A pin or auto wakeup to operate as a keyboard interrupt input.





KBIE5–KBIE0 — Port A Keyboard Interrupt Enable Bits

Each of these read/write bits enables the corresponding keyboard interrupt pin on port A to latch interrupt requests. Reset clears the keyboard interrupt enable register.

1 = KBIx pin enabled as keyboard interrupt pin

0 = KBIx pin not enabled as keyboard interrupt pin

NOTE

AWUIE bit is not used in conjunction with the keyboard interrupt feature. To see a description of this bit, see Chapter 4 Auto Wakeup Module (AWU).



Chapter 11 Oscillator Module (OSC)

11.1 Introduction

The oscillator module is used to provide a stable clock source for the microcontroller system and bus. The oscillator module generates two output clocks, BUSCLKX2 and BUSCLKX4. The BUSCLKX4 clock is used by the system integration module (SIM) and the computer operating properly module (COP). The BUSCLKX2 clock is divided by two in the SIM to be used as the bus clock for the microcontroller. Therefore the bus frequency will be one forth of the BUSCLKX4 frequency.

11.2 Features

The oscillator has these four clock source options available:

- 1. Internal oscillator: An internally generated, fixed frequency clock, trimmable to ±5%. This is the default option out of reset.
- 2. External oscillator: An external clock that can be driven directly into OSC1.
- 3. External RC: A built-in oscillator module (RC oscillator) that requires an external R connection only. The capacitor is internal to the chip.
- 4. External crystal: A built-in oscillator module (XTAL oscillator) that requires an external crystal or ceramic-resonator.

11.3 Functional Description

The oscillator contains these major subsystems:

- Internal oscillator circuit
- Internal or external clock switch control
- External clock circuit
- External crystal circuit
- External RC clock circuit

11.3.1 Internal Oscillator

The internal oscillator circuit is designed for use with no external components to provide a clock source with tolerance less than $\pm 25\%$ untrimmed. An 8-bit trimming register allows adjustment to a tolerance of less than $\pm 5\%$.

The internal oscillator will generate a clock of 4.0 MHz typical (INTCLK) resulting in a bus speed (internal clock \div 4) of 1.0 MHz.

Figure 11-3 shows how BUSCLKX4 is derived from INTCLK and, like the RC oscillator, OSC2 can output BUSCLKX4 by setting OSC2EN in PTAPUE register. See Chapter 12 Input/Output Ports (PORTS).



Oscillator Module (OSC)

11.3.3 XTAL Oscillator

The XTAL oscillator circuit is designed for use with an external low-frequency crystal or ceramic resonator to provide an accurate clock source. In this configuration, the OSC2 pin is dedicated to the external crystal circuit. The OSC2EN bit in the port A pullup enable register has no effect when this clock mode is selected.

In its typical configuration, the XTAL oscillator is connected in a Pierce oscillator configuration, as shown in Figure 11-2. This figure shows only the logical representation of the internal components and may not represent actual circuitry. The oscillator configuration uses five components:

- Crystal, X₁
- Fixed capacitor, C₁
- Tuning capacitor, C₂ (can also be a fixed capacitor)
- Feedback resistor, R_B
- Series resistor, R_s



Figure 11-2. XTAL Oscillator External Connections

11.3.4 RC Oscillator

The RC oscillator circuit is designed for use with an external resistor (R_{EXT}) to provide a clock source with a tolerance within 25% of the expected frequency. See Figure 11-3.

The capacitor (C) for the RC oscillator is internal to the MCU. The R_{EXT} value must have a tolerance of 1% or less to minimize its effect on the frequency.

In this configuration, the OSC2 pin can be left in the reset state as PTA4. Or, the OSC2EN bit in the port A pullup enable register can be set to enable the OSC2 output function on the pin. Enabling the OSC2 output slightly increases the external RC oscillator frequency, f_{RCCLK} .



Oscillator Module (OSC)

11.4.3 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal comes from the system integration module (SIM) and enables/disables either the XTAL oscillator circuit, the RC oscillator, or the internal oscillator.

11.4.4 XTAL Oscillator Clock (XTALCLK)

XTALCLK is the XTAL oscillator output signal. It runs at the full speed of the crystal (f_{XCLK}) and comes directly from the crystal oscillator circuit. Figure 11-2 shows only the logical relation of XTALCLK to OSC1 and OSC2 and may not represent the actual circuitry. The duty cycle of XTALCLK is unknown and may depend on the crystal and other external factors. Also, the frequency and amplitude of XTALCLK can be unstable at start up.

11.4.5 RC Oscillator Clock (RCCLK)

RCCLK is the RC oscillator output signal. Its frequency is directly proportional to the time constant of external R and internal C. Figure 11-3 shows only the logical relation of RCCLK to OSC1 and may not represent the actual circuitry.

11.4.6 Internal Oscillator Clock (INTCLK)

INTCLK is the internal oscillator output signal. Its nominal frequency is fixed to 4.0 MHz, but it can be also trimmed using the oscillator trimming feature of the OSCTRIM register (see 11.3.1.1 Internal Oscillator Trimming).

11.4.7 Oscillator Out 2 (BUSCLKX4)

BUSCLKX4 is the same as the input clock (XTALCLK, RCCLK, or INTCLK). This signal is driven to the SIM module and is used to determine the COP cycles.

11.4.8 Oscillator Out (BUSCLKX2)

The frequency of this signal is equal to half of the BUSCLKX4, this signal is driven to the SIM for generation of the bus clocks used by the CPU and other modules on the MCU. BUSCLKX2 will be divided again in the SIM and results in the internal bus frequency being one fourth of either the XTALCLK, RCCLK, or INTCLK frequency.

11.5 Low Power Modes

The WAIT and STOP instructions put the MCU in low-power consumption standby modes.

11.5.1 Wait Mode

The WAIT instruction has no effect on the oscillator logic. BUSCLKX2 and BUSCLKX4 continue to drive to the SIM module.

11.5.2 Stop Mode

The STOP instruction disables either the XTALCLK, the RCCLK, or INTCLK output, hence BUSCLKX2 and BUSCLKX4.



System Integration Module (SIM)



Figure 13-7. Interrupt Processing



14.4 Functional Description

Figure 14-2 shows the structure of the TIM. The central component of the TIM is the 16-bit TIM counter that can operate as a free-running counter or a modulo up-counter. The TIM counter provides the timing reference for the input capture and output compare functions. The TIM counter modulo registers, TMODH:TMODL, control the modulo value of the TIM counter. Software can read the TIM counter value at any time without affecting the counting sequence.

The two TIM channels are programmable independently as input capture or output compare channels.



Figure 14-2. TIM Block Diagram



Timer Interface Module (TIM)





15.2.2.3 Break Auxiliary Register

The break auxiliary register (BRKAR) contains a bit that enables software to disable the COP while the MCU is in a state of break interrupt with monitor mode.



Figure 15-6. Break Auxiliary Register (BRKAR)

BDCOP — Break Disable COP Bit

This read/write bit disables the COP during a break interrupt. Reset clears the BDCOP bit.

- 1 = COP disabled during break interrupt
- 0 = COP enabled during break interrupt.

15.2.2.4 Break Status Register

The break status register (BSR) contains a flag to indicate that a break caused an exit from wait mode. This register is only used in emulation mode.



Figure 15-7. Break Status Register (BSR)

SBSW — SIM Break Stop/Wait

SBSW can be read within the break state SWI routine. The user can modify the return address on the stack by subtracting one from it.

1 = Wait mode was exited by break interrupt

0 = Wait mode was not exited by break interrupt

15.2.2.5 Break Flag Control Register

The break control register (BFCR) contains a bit that enables software to clear status bits while the MCU is in a break state.

