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	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	36
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	42-SDIP (0.600", 15.24mm)
Supplier Device Package	42-PDIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908gt16cbe

Revision History (Sheet 2 of 2)

Date	Revision Level	Description	Page Number(s)
September, 2004	3.0 (Continued from previous page)	Chapter 15 System Integration Module (SIM) — Clarified SIM features and functionality	179, 182, 183, 184
		15.7.2 SIM Reset Status Register — Clarified SRSR operation	194
		Table 19-1. Monitor Mode Signal Requirements and Options — Reworked	247
		19.2.1 Functional Description — Corrected Break description	237, 240
		19.3 Monitor Module (MON) — Reworked	243
		Chapter 20 Electrical Specifications — Revised/added tables: 20.5 5.0-V DC Electrical Characteristics 20.6 3.0-V DC Electrical Characteristics 20.7 Supply Current Characteristics 20.8 5-V Control Timing 20.9 3-V Control Timing	 257 258 259 260 260
		20.20 Memory Characteristics — Updated memory table	273
		Chapter 20 Electrical Specifications — Added figures: Figure 20-1. RST and IRQ Timing Figure 20-2. RST and IRQ Timing	 260 260
March, 2006	4.0	Appendix A MC68HC08GT16 — Introduces the MC68HC08GT16, the ROM part equivalent to the MC68HC908GT16.	281
April, 2007	5.0	4.2 Functional Description — In the description of the COP Rate Select Bit corrected the values for COP timeout period	57
		Figure 5-1. COP Block Diagram — Replaced BUSCLKX4 with COPCLK	61
		14.9.1 ESCI Arbiter Control Register — Replaced one half with one quarter in definition for ACLK = 0	176
		14.9.3 Bit Time Measurement — Replaced one half with one quarter in definition for ACLK = 0	177
		Revised the following diagrams: Figure 19-10. Forced Monitor Mode (Low) Figure 19-11. Forced Monitor Mode (High) Figure 19-12. Standard Monitor Mode	 245 245 246

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001A	Keyboard Status and Control Register (INTKBSCR) See page 111.	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$001B	Keyboard Interrupt Enable Register (INTKBIER) See page 112.	Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001C	Timebase Module Control Register (TBCR) See page 218.	Read:	TBIF	TBR2	TBR1	TBR0	0	TBIE	TBON	R
		Write:					TACK			
		Reset:	0	0	0	0	0	0	0	0
$001D	IRQ Status and Control Register (INTSCR) See page 104.	Read:	0	0	0	0	IRQF1	0	IMASK1	MODE1
		Write:						ACK1		
		Reset:	0	0	0	0	0	0	0	0
$001E	Configuration Register 2 (CONFIG2)† See page 58.	Read:	R	0	EXT-XTALEN	EXT-SLOW	EXT-CLKEN	0	OSCENIN-STOP	R
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001F	Configuration Register 1 (CONFIG1)† See page 58.	Read:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVI5OR3(1)	SSREC	STOP	COPD
		Write:								
		Reset:	0	0	0	0	0	0	0	0

1. One-time writable register after each reset, except LVI5OR3 bit. LVI5OR3 bit is only reset via POR (power-on reset).

$0020	Timer 1 Status and Control Register (T1SC) See page 231.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0021	Timer 1 Counter Register High (T1CNTH) See page 232.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0022	Timer 1 Counter Register Low (T1CNTL) See page 232.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0023	Timer 1 Counter Modulo Register High (T1MODH) See page 233.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0024	Timer 1 Counter Modulo Register Low (T1MODL) See page 233.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0025	Timer 1 Channel 0 Status and Control Register (T1SC0) See page 233.	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 3 of 7)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0033	Timer 2 Channel 1 Status and Control Register (T2SC1) See page 234.	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0034	Timer 2 Channel 1 Register High (T2CH1H) See page 236.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0035	Timer 2 Channel 1 Register Low (T2CH1L) See page 236.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0036	ICG Control Register (ICGCR) See page 98.	Read:	CMIE	CMF	CMON	CS	ICGON	ICGS	ECGON	ECGS
		Write:		0						
		Reset:	0	0	0	0	1	0	0	0
$0037	ICG Multiplier Register (ICGMR) See page 99.	Read:		N6	$\overline{N5}$	N4	N3	N2	N1	N0
		Write:								
		Reset:	0	0	0	1	0	1	0	1
$0038	ICG Trim Register (ICGTR) See page 100.	Read:	TRIM7	TRIM6	TRIM5	TRIM4	TRIM3	TRIM2	TRIM1	TRIM0
		Write:								
		Reset:	1	0	0	0	0	0	0	0
$0039	ICG Divider Control Register (ICGDVR) See page 100.	Read:					DDIV3	DDIV2	DDIV1	DDIV0
		Write:								
		Reset:	0	0	0	0	U	U	U	U
$003A	ICG DCO Stage Control Register (ICGDSR) See page 100.	Read:	DSTG7	DSTG6	DSTG5	DSTG4	DSTG3	DSTG2	DSTG1	DSTG0
		Write:	R	R	R	R	R	R	R	R
		Reset:	Unaffected by reset							
$003B	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:	Indeterminate after reset							
$003C	ADC Status and Control Register (ADSCR) See page 54.	Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
		Write:	R							
		Reset:	0	0	0	1	1	1	1	1
$003D	ADC Data Register (ADR) See page 55.	Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$003E	ADC Clock Register (ADCLK) See page 56.	Read:	ADIV2	ADIV1	ADIV0	ADICLK	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$003F	Unimplemented	Read:								
		Write:								
		Reset:								

= Unimplemented R = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 5 of 7)

2.5 Random-Access Memory (RAM)

Addresses $0040 through $023F are RAM locations. The location of the stack RAM is programmable. The 16-bit stack pointer allows the stack to be anywhere in the 64-Kbyte memory space.

NOTE

For correct operation, the stack pointer must point only to RAM locations.

Within page zero are 192 bytes of RAM. Because the location of the stack RAM is programmable, all page zero RAM locations can be used for I/O control and user data or code. When the stack pointer is moved from its reset location at $00FF out of page zero, direct addressing mode instructions can efficiently access all page zero RAM locations. Page zero RAM, therefore, provides ideal locations for frequently accessed global variables.

Before processing an interrupt, the CPU uses five bytes of the stack to save the contents of the CPU registers.

NOTE

For M6805 compatibility, the H register is not stacked.

During a subroutine call, the CPU uses two bytes of the stack to store the return address. The stack pointer decrements during pushes and increments during pulls.

NOTE

Be careful when using nested subroutines. The CPU may overwrite data in the RAM during a subroutine or during the interrupt stacking operation.

2.6 Flash Memory

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

2.6.1 Functional Description

The Flash memory is an array of 15,872 bytes (7,680 bytes on MC68HC908GT8) with an additional 36 bytes of user vectors, one byte of block protection and two bytes of ICG user trim storage. *An erased bit reads as 1 and a programmed bit reads as a 0.* Memory in the Flash array is organized into two rows per page basis. The page size is 64 bytes per page and the row size is 32 bytes per row. Hence the minimum erase page size is 64 bytes and the minimum program row size is 32 bytes. Program and erase operation operations are facilitated through control bits in Flash control register (FLCR). Details for these operations appear later in this section.

The address ranges for the user memory and vectors are:

- $C000–$FDFF; user memory ($E000–$FDFF on MC68HC908GT8)
- $FE08; Flash control register
- $FF7E; Flash block protect register
- $FF80; ICG user trim register (ICGTR5)
- $FF81; ICG user trim register (ICGTR3)
- $FFDC–$FFFF; these locations are reserved for user-defined interrupt and reset vectors

Settling time depends primarily on how many corrections it takes to change the clock period and the period of each correction. Since the corrections require four periods of the low-frequency base clock ($4*\tau_{IBASE}$), and since ICLK is N (the ICG multiply factor for the desired frequency) times faster than IBASE, each correction takes $4*N*\tau_{ICLK}$. The period of ICLK, however, will vary as the corrections occur.

7.4.6.1 Settling to Within 15 Percent

When the error is greater than 15 percent, the filter takes eight corrections to double or halve the clock period. Due to how the DCO increases or decreases the clock period, the total period of these eight corrections is approximately 11 times the period of the fastest correction. (If the corrections were perfectly linear, the total period would be 11.5 times the minimum period; however, the ring must be slightly nonlinear.) Therefore, the total time it takes to double or halve the clock period is $44*N*\tau_{ICLKFAST}$.

If the clock period needs more than doubled or halved, the same relationship applies, only for each time the clock period needs doubled, the total number of cycles doubles. That is, when transitioning from fast to slow, going from the initial speed to half speed takes $44*N*\tau_{ICLKFAST}$; from half speed to quarter speed takes $88*N*\tau_{ICLKFAST}$; going from quarter speed to eighth speed takes $176*N*\tau_{ICLKFAST}$; and so on. This series can be expressed as $(2^X–1)*44*N*\tau_{ICLKFAST}$, where x is the number of times the speed needs doubled or halved. Since 2^X happens to be equal to $\tau_{ICLKSLOW}/\tau_{ICLKFAST}$, the equation reduces to $44*N*(\tau_{ICLKSLOW}–\tau_{ICLKFAST})$.

Note that increasing speed takes much longer than decreasing speed since N is higher. This can be expressed in terms of the initial clock period (τ_1) minus the final clock period (τ_2) as such:

$$\tau_{15} = \text{abs}[44N(\tau_1 - \tau_2)]$$

7.4.6.2 Settling to Within 5 Percent

Once the clock period is within 15 percent of the desired clock period, the filter starts making smaller adjustments. When between 15 percent and 5 percent error, each correction will adjust the clock period between 1.61 percent and 2.94 percent. In this mode, a maximum of eight corrections will be required to get to less than 5 percent error. Since the clock period is relatively close to desired, each correction takes approximately the same period of time, or $4*\tau_{IBASE}$. At this point, the internal clock stable bit (ICGS) will be set and the clock frequency is usable, although the error will be as high as 5 percent. The total time to this point is:

$$\tau_5 = \text{abs}[44N(\tau_1 - \tau_2)] + 32\tau_{IBASE}$$

7.4.6.3 Total Settling Time

Once the clock period is within 5 percent of the desired clock period, the filter starts making minimum adjustments. In this mode, each correction will adjust the frequency between 0.202 percent and 0.368 percent. A maximum of 24 corrections will be required to get to the minimum error. Each correction takes approximately the same period of time, or $4*\tau_{IBASE}$. Added to the corrections for 15 percent to 5 percent, this makes 32 corrections ($128*\tau_{IBASE}$) to get from 15 percent to the minimum error. The total time to the minimum error is:

$$\tau_{tot} = \text{abs}[44N(\tau_1 - \tau_2)] + 128\tau_{IBASE}$$

The equations for τ_{15}, τ_5, and τ_{tot} are dependent on the actual initial and final clock periods τ_1 and τ_2, not the nominal. This means the variability in the ICLK frequency due to process, temperature, and voltage must be considered. Additionally, other process factors and noise can affect the actual tolerances of the points at which the filter changes modes. This means a worst case adjustment of up to 35 percent (ICLK

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 pin 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.

To determine the logic level on a keyboard interrupt pin, use the data direction register to configure the pin as an input and read the data register.

NOTE

Setting a keyboard interrupt enable bit (KBIEx) forces the corresponding keyboard interrupt pin to be an input, overriding the data direction register. However, the data direction register bit must be a 0 for software to read the pin.

9.4 Keyboard Initialization

When a keyboard interrupt pin is enabled, it takes time for the internal pullup to reach a logic 1. Therefore, a false interrupt can occur as soon as the pin is enabled.

To prevent a false interrupt on keyboard initialization:

1. Mask keyboard interrupts by setting the IMASKK bit in the keyboard status and control register.
2. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.
3. Write to the ACKK bit in the keyboard status and control register to clear any false interrupts.
4. Clear the IMASKK bit.

An interrupt signal on an edge-triggered pin can be acknowledged immediately after enabling the pin. An interrupt signal on an edge- and level-triggered interrupt pin must be acknowledged after a delay that depends on the external load.

Another way to avoid a false interrupt:

1. Configure the keyboard pins as outputs by setting the appropriate DDRA bits in data direction register A.
2. Write 1s to the appropriate port A data register bits.
3. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.

9.5 Low-Power Modes

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

9.5.1 Wait Mode

The keyboard module remains active in wait mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of wait mode.

11.9.2 Stop Mode

If enabled, the LVI module remains active in stop mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of stop mode.

11.10 Enhanced Serial Communications Interface Module (SCI)

11.10.1 Wait Mode

The enhanced serial communications interface (ESCI), or SCI module for short, module remains active in wait mode. Any enabled CPU interrupt request from the SCI module can bring the MCU out of wait mode.

If SCI module functions are not required during wait mode, reduce power consumption by disabling the module before executing the WAIT instruction.

11.10.2 Stop Mode

The SCI module is inactive in stop mode. The STOP instruction does not affect SCI register states. SCI module operation resumes after the MCU exits stop mode.

Because the internal clock is inactive during stop mode, entering stop mode during an SCI transmission or reception results in invalid data.

11.11 Serial Peripheral Interface Module (SPI)

11.11.1 Wait Mode

The serial peripheral interface (SPI) module remains active in wait mode. Any enabled CPU interrupt request from the SPI module can bring the MCU out of wait mode.

If SPI module functions are not required during wait mode, reduce power consumption by disabling the SPI module before executing the WAIT instruction.

11.11.2 Stop Mode

The SPI module is inactive in stop mode. The STOP instruction does not affect SPI register states. SPI operation resumes after an external interrupt. If stop mode is exited by reset, any transfer in progress is aborted, and the SPI is reset.

11.12 Timer Interface Module (TIM1 and TIM2)

11.12.1 Wait Mode

The timer interface modules (TIM) remain active in wait mode. Any enabled CPU interrupt request from the TIM can bring the MCU out of wait mode.

If TIM functions are not required during wait mode, reduce power consumption by stopping the TIM before executing the WAIT instruction.

PTAPUE7–PTAPUE0 — Port A Input Pullup Enable Bits

These writable bits are software programmable to enable pullup devices on an input port bit.

1 = Corresponding port A pin configured to have internal pullup
0 = Corresponding port A pin has internal pullup disconnected

12.3 Port B

Port B is an 8-bit special-function port that shares all eight of its pins with the analog-to-digital converter (ADC) module.

12.3.1 Port B Data Register

The port B data register (PTB) contains a data latch for each of the eight port pins.

Address: $0001

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Reset:	Unaffected by reset							
Alternative Function:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Figure 12-6. Port B Data Register (PTB)

PTB7–PTB0 — Port B Data Bits

These read/write bits are software-programmable. Data direction of each port B pin is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.

AD7–AD0 — Analog-to-Digital Input Bits

AD7–AD0 are pins used for the input channels to the analog-to-digital converter module. The channel select bits in the ADC status and control register define which port B pin will be used as an ADC input and overrides any control from the port I/O logic by forcing that pin as the input to the analog circuitry.

NOTE

Care must be taken when reading port B while applying analog voltages to AD7–AD0 pins. If the appropriate ADC channel is not enabled, excessive current drain may occur if analog voltages are applied to the PTBx/ADx pin, while PTB is read as a digital input. Those ports not selected as analog input channels are considered digital I/O ports.

12.3.2 Data Direction Register B

Data direction register B (DDRB) determines whether each port B pin is an input or an output. Writing a 1 to a DDRB bit enables the output buffer for the corresponding port B pin; a 0 disables the output buffer.

Address: $0005

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Reset:	0	0	0	0	0	0	0	0

Figure 12-7. Data Direction Register B (DDRB)

Address:	$000C							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 12-18. Data Direction Register E (DDRE)

DDRE4–DDRE0 — Data Direction Register E Bits

These read/write bits control port E data direction. Reset clears DDRE4–DDRE0, configuring all port E pins as inputs.

1 = Corresponding port E pin configured as output
0 = Corresponding port E pin configured as input

NOTE

Avoid glitches on port E pins by writing to the port E data register before changing data direction register E bits from 0 to 1.

Figure 12-19 shows the port E I/O logic.

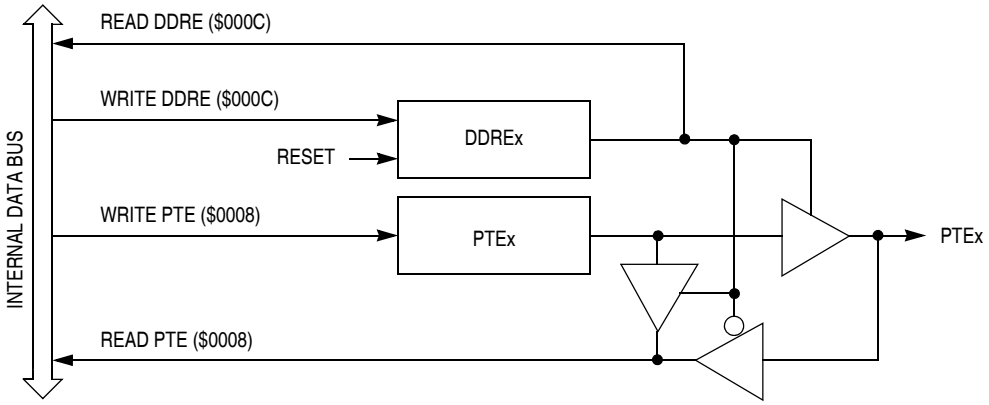


Figure 12-19. Port E I/O Circuit

When bit DDREx is a 1, reading address $0008 reads the PTEx data latch. When bit DDREx is a 0, reading address $0008 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 12-6 summarizes the operation of the port E pins.

Table 12-6. Port E Pin Functions

DDRE Bit	PTE Bit	I/O Pin Mode	Accesses to DDRE	Accesses to PTE	
			Read/Write	Read	Write
0	X[1]	Input, Hi-Z[2]	DDRE4–DDRE0	Pin	PTE4–PTE0[3]
1	X	Output	DDRE4–DDRE0	PTE4–PTE0	PTE4–PTE0

1. X = Don't care
2. Hi-Z = High impedance
3. Writing affects data register, but does not affect input.

modifies the H register or uses the indexed addressing mode, save the H register and then restore it prior to exiting the routine.

13.3.2 Sources

The sources in Table 13-1 can generate CPU interrupt requests.

13.3.2.1 Software Interrupt (SWI) Instruction

The software interrupt instruction (SWI) causes a non-maskable interrupt.

NOTE

*A software interrupt pushes PC onto the stack. An SWI does **not** push PC – 1, as a hardware interrupt does.*

Table 13-1. Interrupt Sources

Source	Flag	Mask[1]	INT Register Flag	Priority[2]	Vector Address
Reset	None	None	None	0	$FFFE—$FFFF
SWI instruction	None	None	None	0	$FFFC—$FFFD
$\overline{\text{IRQ}}$ pin	IRQF	IMASK1	IF1	1	$FFFA—$FFFB
ICG clock monitor	CMF	CMIE	IF2	2	$FFF8–$FFF9
TIM1 channel 0	CH0F	CH0IE	IF3	3	$FFF6–$FFF7
TIM1 channel 1	CH1F	CH1IE	IF4	4	$FFF4–$FFF5
TIM1 overflow	TOF	TOIE	IF5	5	$FFF2–$FFF3
TIM2 channel 0	CH0F	CH0IE	IF6	6	$FFF0–$FFF1
TIM2 channel 1	CH1F	CH1IE	IF7	7	$FFEE–$FFEF
TIM2 overflow	TOF	TOIE	IF8	8	$FFEC–$FFED
SPI receiver full	SPRF	SPRIE	IF9	9	$FFEA–$FFEB
SPI overflow	OVRF	ERRIE			
SPI mode fault	MODF	ERRIE			
SPI transmitter empty	SPTE	SPTIE	IF10	10	$FFE8–$FFE9
SCI receiver overrun	OR	ORIE	IF11	11	$FFE6–$FFE7
SCI noise fag	NF	NEIE			
SCI framing error	FE	FEIE			
SCI parity error	PE	PEIE			
SCI receiver full	SCRF	SCRIE	IF12	12	$FFE4–$FFE5
SCI input idle	IDLE	ILIE			
SCI transmitter empty	SCTE	SCTIE	IF13	13	$FFE2–$FFE3
SCI transmission complete	TC	TCIE			
Keyboard pin	KEYF	IMASKK	IF14	14	$FFE0–$FFE1
ADC conversion complete	COCO	AIEN	IF15	15	$FFDE–$FFDF
Timebase	TBIF	TBIE	IF16	16	$FFDC–$FFDD

1. The I bit in the condition code register is a global mask for all interrupt sources except the SWI instruction.
2. 0 = highest priority

14.8.6 ESCI Data Register

The ESCI data register (SCDR) is the buffer between the internal data bus and the receive and transmit shift registers. Reset has no effect on data in the ESCI data register.

Address: $0018

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R7	R6	R5	R4	R3	R2	R1	R0
Write:	T7	T6	T5	T4	T3	T2	T1	T0
Reset:	Unaffected by reset							

Figure 14-16. ESCI Data Register (SCDR)

R7/T7:R0/T0 — Receive/Transmit Data Bits

Reading address $0018 accesses the read-only received data bits, R7:R0. Writing to address $0018 writes the data to be transmitted, T7:T0. Reset has no effect on the ESCI data register.

NOTE

Do not use read-modify-write instructions on the ESCI data register.

14.8.7 ESCI Baud Rate Register

The ESCI baud rate register (SCBR) together with the ESCI prescaler register selects the baud rate for both the receiver and the transmitter.

NOTE

There are two prescalers available to adjust the baud rate. One in the ESCI baud rate register and one in the ESCI prescaler register.

Address: $0019

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	R	LINR	SCP1	SCP0	R	SCR2	SCR1	SCR0
Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved

Figure 14-17. ESCI Baud Rate Register (SCBR)

LINR — LIN Receiver Bits

This read/write bit selects the enhanced ESCI features for the local interconnect network (LIN) protocol as shown in Table 14-6. Reset clears LINR.

Table 14-6. ESCI LIN Control Bits

LINR	M	Functionality
0	X	Normal ESCI functionality
1	0	11-bit break detect enabled for LIN receiver
1	1	12-bit break detect enabled for LIN receiver

16.3.1 Master Mode

The SPI operates in master mode when the SPI master bit, SPMSTR, is set.

NOTE

In a multi-SPI system, configure the SPI modules as master or slave before enabling them. Enable the master SPI before enabling the slave SPI. Disable the slave SPI before disabling the master SPI. See 16.12.1 SPI Control Register.

Only a master SPI module can initiate transmissions. Software begins the transmission from a master SPI module by writing to the transmit data register. If the shift register is empty, the byte immediately transfers to the shift register, setting the SPI transmitter empty bit, SPTE. The byte begins shifting out on the MOSI pin under the control of the serial clock. See Figure 16-4.

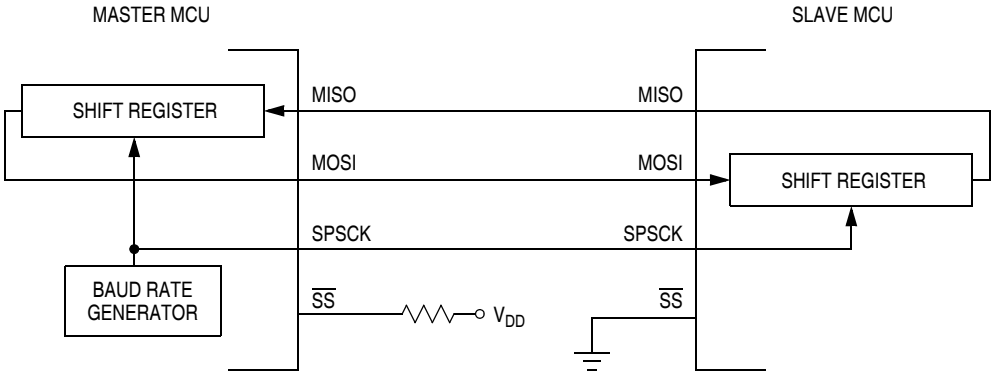


Figure 16-4. Full-Duplex Master-Slave Connections

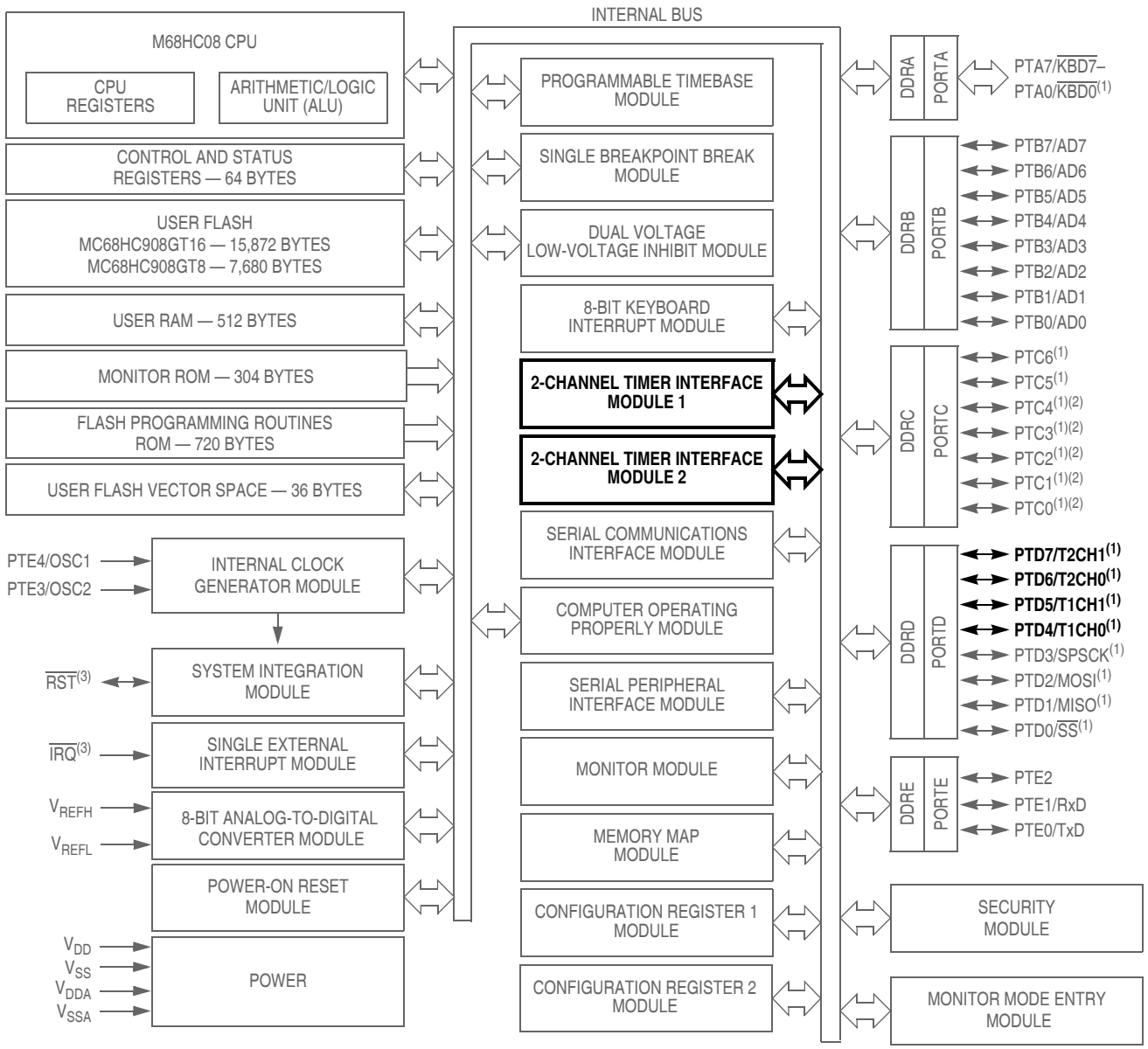
The SPR1 and SPR0 bits control the baud rate generator and determine the speed of the shift register. (See 16.12.2 SPI Status and Control Register.) Through the SPSCK pin, the baud rate generator of the master also controls the shift register of the slave peripheral.

As the byte shifts out on the MOSI pin of the master, another byte shifts in from the slave on the master's MISO pin. The transmission ends when the receiver full bit, SPRF, becomes set. At the same time that SPRF becomes set, the byte from the slave transfers to the receive data register. In normal operation, SPRF signals the end of a transmission. Software clears SPRF by reading the SPI status and control register with SPRF set and then reading the SPI data register. Writing to the SPI data register (SPDR) clears SPTE.

16.3.2 Slave Mode

The SPI operates in slave mode when SPMSTR is clear. In slave mode, the SPSCK pin is the input for the serial clock from the master MCU. Before a data transmission occurs, the $\overline{SS}$ pin of the slave SPI must be low. $\overline{SS}$ must remain low until the transmission is complete. See 16.6.2 Mode Fault Error.

In a slave SPI module, data enters the shift register under the control of the serial clock from the master SPI module. After a byte enters the shift register of a slave SPI, it transfers to the receive data register, and the SPRF bit is set. To prevent an overflow condition, slave software then must read the receive data register before another full byte enters the shift register.



1. Ports are software configurable with pullup device if input port.
2. Higher current drive port pins
3. Pin contains integrated pullup device

Figure 18-2. Block Diagram Highlighting TIM Blocks and Pins

into the TIM channel registers, TCHxH:TCHxL. The polarity of the active edge is programmable. Input captures can generate TIM CPU interrupt requests.

18.4.3 Output Compare

With the output compare function, the TIM can generate a periodic pulse with a programmable polarity, duration, and frequency. When the counter reaches the value in the registers of an output compare channel, the TIM can set, clear, or toggle the channel pin. Output compares can generate TIM CPU interrupt requests.

18.4.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in 18.4.3 Output Compare. The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change an output compare value could cause incorrect operation for up to two counter overflow periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that counter overflow period. Also, using a TIM overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIM may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the output compare value on channel x:

- When changing to a smaller value, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current output compare pulse. The interrupt routine has until the end of the counter overflow period to write the new value.
- When changing to a larger output compare value, enable TIM overflow interrupts and write the new value in the TIM overflow interrupt routine. The TIM overflow interrupt occurs at the end of the current counter overflow period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same counter overflow period.

18.4.3.2 Buffered Output Compare

Channels 0 and 1 can be linked to form a buffered output compare channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the output.

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The output compare value in the TIM channel 0 registers initially controls the output on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the output after the TIM overflows. At each subsequent overflow, the TIM channel registers (0 or 1) that control the output are the ones written to last. TSC0 controls and monitors the buffered output compare function, and TIM channel 1 status and control register (TSC1) is unused. While the MS0B bit is set, the channel 1 pin, TCH1, is available as a general-purpose I/O pin.

NOTE

In buffered output compare operation, do not write new output compare values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered output compares.

19.2.2.3 SIM Break Status Register

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

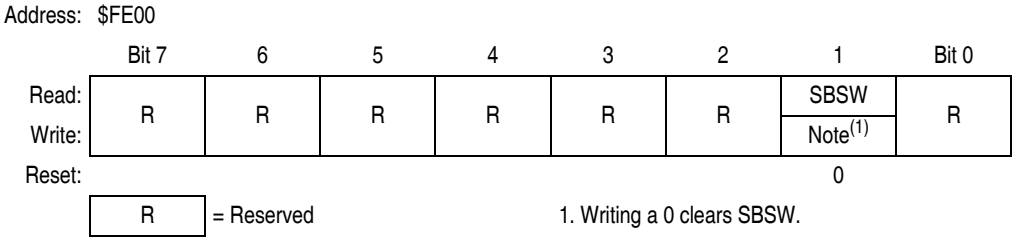


Figure 19-7. SIM Break Status Register (SBSR)

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

19.2.2.4 Break Flag Control Register

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

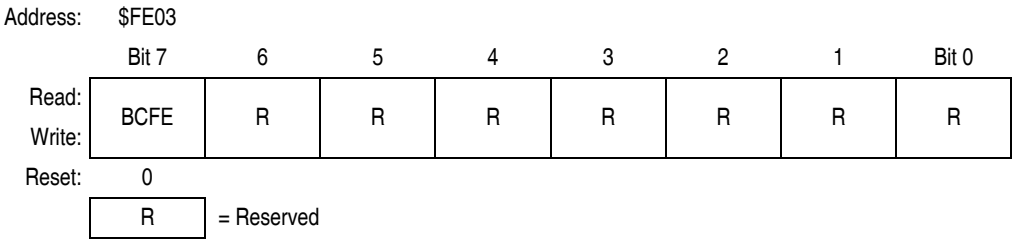


Figure 19-8. SIM Break Flag Control Register (SBFCR)

BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

1 = Status bits clearable during break
0 = Status bits not clearable during break

19.3 Monitor Module (MON)

The monitor module allows debugging and programming of the microcontroller unit (MCU) through a single-wire interface with a host computer. Monitor mode entry can be achieved without use of the higher test voltage, V_{TST}, as long as vector addresses $FFFE and $FFFF are blank, thus reducing the hardware requirements for in-circuit programming.

Features of the monitor module include:

- Normal user-mode pin functionality
- One pin dedicated to serial communication between monitor read-only memory (ROM) and host computer
- Standard mark/space non-return-to-zero (NRZ) communication with host computer
- Execution of code in random-access memory (RAM) or Flash
- Flash memory security feature(1)
- Flash memory programming interface
- External 4.92 MHz or 9.83 MHz clock used to generate internal frequency of 2.4576 MHz
- Optional ICG mode of operation (no external clock or high voltage)
- Monitor mode entry without high voltage, V_{TST}, if reset vector is blank ($FFFE and $FFFF contain $FF)
- Normal monitor mode entry if high voltage is applied to $\overline{IRQ}$

19.3.1 Functional Description

Figure 19-9 shows a simplified monitor mode entry flowchart.

The monitor ROM receives and executes commands from a host computer. Figure 19-10, Figure 19-11, and Figure 19-12 show example circuits used to enter monitor mode and communicate with a host computer via a standard RS-232 interface.

Simple monitor commands can access any memory address. In monitor mode, the MCU can execute code downloaded into RAM by a host computer while most MCU pins retain normal operating mode functions. All communication between the host computer and the MCU is through the PTA0 pin. A level-shifting and multiplexing interface is required between PTA0 and the host computer. PTA0 is used in a wired-OR configuration and requires a pullup resistor.

The monitor code has been updated from previous versions of the monitor code to allow the ICG to generate the internal clock. This option, which is selected when $\overline{IRQ}$ is held low out of reset, is intended to support serial communication/ programming at 9600 baud in monitor mode by using the ICG, and the ICG user trim value ICGTR5 (if programmed) to generate the desired internal frequency (2.4576 MHz). If ICGTR5 is not programmed (i.e., the value is $FF) then the ICG will operate at a nominal (untrimmed) 2.45 MHz and communications will be nominally at 9600 baud but the untrimmed rate may cause difficulties with hosts which cannot automatically adjust their data rates to match.

Since this feature is enabled only when $\overline{IRQ}$ is held low out of reset, it cannot be used when the reset vector is programmed (i.e., the value is not $FFFF) because entry into monitor mode in this case requires V_{TST} on $\overline{IRQ}$.

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

Chapter 20
Electrical Specifications

20.1 Introduction

This section contains electrical and timing specifications.

20.2 Absolute Maximum Ratings

Maximum ratings are the extreme limits to which the microcontroller unit (MCU) can be exposed without permanently damaging it.

NOTE

This device is not guaranteed to operate properly at the maximum ratings. Refer to 20.5 5.0-V DC Electrical Characteristics for guaranteed operating conditions.

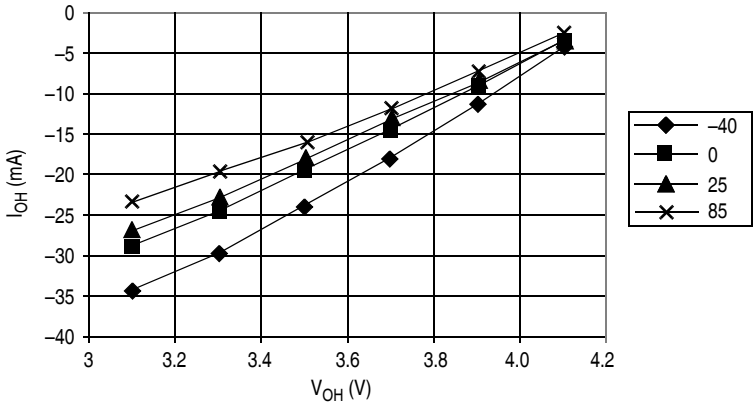
Characteristic[1]	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to + 6.0	V
Input voltage	V_{In}	V_{SS} – 0.3 to V_{DD} + 0.3	V
Maximum current per pin excluding those specified below	I	± 15	mA
Maximum current for pins PTA5-PTA7, PTD4	$I_{PTA5–PTA7}$	± 20	mA
Maximum current for pins PTC0–PTC4	$I_{PTC0–PTC4}$	± 25	mA
Maximum current into V_{DD}	I_{mvdd}	150	mA
Maximum current out of V_{SS}	I_{mvss}	150	mA
Storage temperature	T_{stg}	–55 to +150	°C

1. Voltages referenced to V_{SS}

NOTE

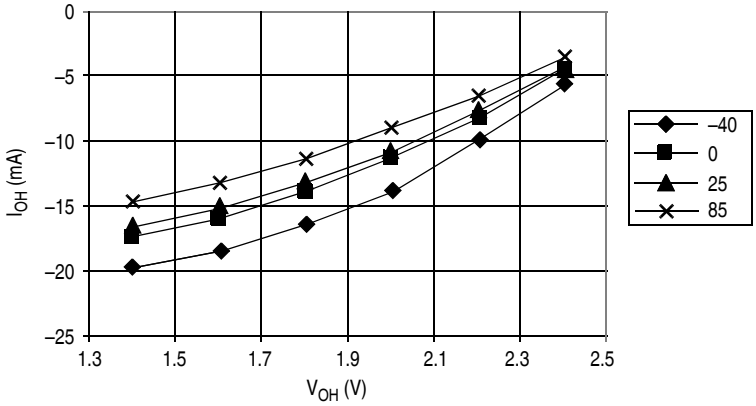
This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{In} and V_{Out} be constrained to the range $V_{SS} \le (V_{In}$ or $V_{Out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD}).

20.13 Output High-Voltage Characteristics



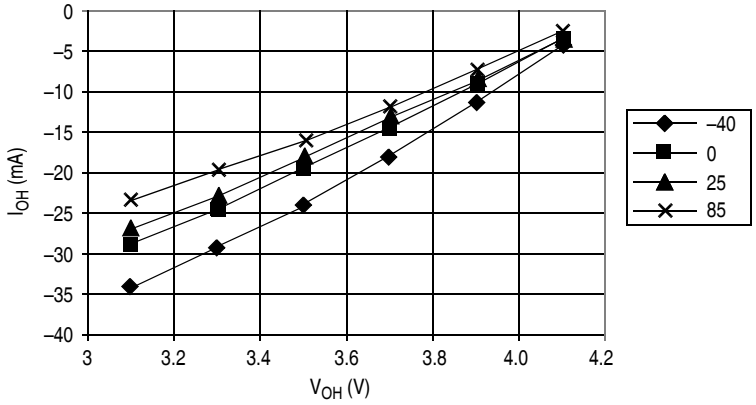
$V_{OH} > V_{DD} –0.8$ V @ $I_{OH} = –2.0$ mA
$V_{OH} > V_{DD} –1.5$ V @ $I_{OH} = –10.0$ mA

Figure 20-3. Typical High-Side Driver Characteristics – Port PTA7–PTA0 (V_{DD} = 4.5 Vdc)



$V_{OH} > V_{DD} –0.3$ V @ $I_{OH} = –0.6$ mA
$V_{OH} > V_{DD} –1.0$ V @ $I_{OH} = –10.0$ mA

Figure 20-4. Typical High-Side Driver Characteristics – Port PTA7–PTA0 (V_{DD} = 2.7 Vdc)



$V_{OH} > V_{DD} –1.5$ V @ $I_{OH} = –20.0$ mA

Figure 20-5. Typical High-Side Driver Characteristics – Port PTC4–PTC0 (V_{DD} = 4.5 Vdc)

Chapter 21
Ordering Information and Mechanical Specifications

21.1 Introduction

This section contains ordering numbers for the MC68HC908GT16 and MC68HC908GT8 and gives the dimensions for:

- 42-pin shrink dual in-line package (case 858-01)
- 44-pin plastic quad flat pack (case 824A-01)

The following figures show the latest package drawings at the time of this publication. To make sure that you have the latest package specifications, contact your local Freescale Semiconductor sales office.

21.2 MC Order Numbers

Table 21-1. MC Order Numbers

MC Order Number	Operating Temperature Range	Package
MC908GT16CB	–40°C to +85°C	42-pin SDIP
MC908GT16CFB	–40°C to +85°C	44-pin QFP
MC908GT8CB	–40°C to +85°C	42-pin SDIP
MC908GT8CFB	–40°C to +85°C	44-pin QFP

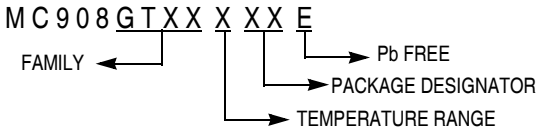


Figure 21-1. Device Numbering System

21.3 Package Dimensions

Refer to the following pages for detailed package dimensions.