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	Not For New Designs
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	37
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908gz16mfae

Chapter 19
Timer Interface Module (TIM)

Chapter 20
Development Support

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$003C	ADC Status and Control Register (ADSCR) See page 53.	Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
		Write:								
		Reset:	0	0	0	1	1	1	1	1
$003D	ADC Data High Register (ADRH) See page 55.	Read:	0	0	0	0	0	0	AD9	AD8
		Write:								
		Reset:				Unaffected by reset				
$003E	ADC Data Low Register (ADRL) See page 55.	Read:	AD7	AD6	AD5	AD4	A3	AD2	AD1	AD0
		Write:								
		Reset:				Unaffected by reset				
$003F	ADC Clock Register (ADCLK) See page 57.	Read:	ADIV2	ADIV1	ADIV0	ADICLK	MODE1	MODE0	R	0
		Write:								
		Reset:	0	0	0	0	0	1	0	0
$0500 ↓ $0508	MSCAN08 Control Registers See page 141.		MSCAN08 control registers (9 bytes) Refer to 12.13 Programmer's Model of Control Registers							
$0509 ↓ $050D	Reserved		Reserved (5 bytes)							
$050E ↓ $050F	MSCAN08 Error Counters		MSCAN08 error counters (2 bytes)							
$0510 ↓ $0517	MSCAN08 Identifier Filter See page 141.		MSCAN08 control registers (9 bytes) Refer to 12.13 Programmer's Model of Control Registers							
$0518 ↓ $053F	Reserved		Reserved (40 bytes)							
$0540 ↓ $054F	MSCAN08 Receive Buffer See page 137.		MSC08 receive buffer Refer to 12.12 Programmer's Model of Message Storage							
$0550 ↓ $055F	MSCAN08 Transmit Buffer 0 See page 137.		MSC08 transmitter buffer 0 Refer to 12.12 Programmer's Model of Message Storage							

	= Unimplemented	R = Reserved	U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 6 of 8)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE0A	Break Address Register Low (BRKL) See page 274.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0B	Break Status and Control Register (BRKSCR) See page 274.	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0C	LVI Status Register (LVISR) See page 119.	Read:	LVIOUT	0	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FF7E	FLASH Block Protect Register (FLBPR)[3] See page 44.	Read:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Write:								
		Reset:	Unaffected by reset							

3. Nonvolatile FLASH register

Addr.	Register Name		
$FFFF	COP Control Register (COPCTL) See page 85.	Read:	Low byte of reset vector
		Write:	Writing clears COP counter (any value)
		Reset:	Unaffected by reset

☐ = Unimplemented   R = Reserved   U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 8 of 8)

2.5 Random-Access Memory (RAM)

Addresses $0040 through $043F 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 (FLASH)

This subsection 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. It is recommended that the user utilize the FLASH programming routines provided in the on-chip ROM, which are described more fully in a separate application note.

2.6.1 Functional Description

The FLASH memory is an array of 15,872 bytes with an additional 44 bytes of user vectors and one byte of block protection. *An erased bit reads as logic 1 and a programmed bit reads as a logic 0*. Memory in the FLASH array is organized into two rows per page basis. For the 16-K word by 8-bit embedded FLASH memory, 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.

8.4 $\overline{\text{IRQ}}$ Pin

A logic 0 on the $\overline{\text{IRQ}}$ pin can latch an interrupt request into the IRQ latch. A vector fetch, software clear, or reset clears the IRQ latch.

If the MODE bit is set, the $\overline{\text{IRQ}}$ pin is both falling-edge-sensitive and low-level-sensitive. With MODE set, both of the following actions must occur to clear IRQ:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the latch. Software may generate the interrupt acknowledge signal by writing a logic 1 to the ACK bit in the interrupt status and control register (INTSCR). The ACK bit is useful in applications that poll the $\overline{\text{IRQ}}$ pin and require software to clear the IRQ latch. Writing to the ACK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACK does not affect subsequent transitions on the $\overline{\text{IRQ}}$ pin. A falling edge that occurs after writing to the ACK bit another interrupt request. If the IRQ mask bit, IMASK, is clear, the CPU loads the program counter with the vector address at locations $FFFA and $FFFB.

- Return of the $\overline{\text{IRQ}}$ pin to logic 1 — As long as the $\overline{\text{IRQ}}$ pin is at logic 0, IRQ remains active.

The vector fetch or software clear and the return of the $\overline{\text{IRQ}}$ pin to logic 1 may occur in any order. The interrupt request remains pending as long as the $\overline{\text{IRQ}}$ pin is at logic 0. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

If the MODE bit is clear, the $\overline{\text{IRQ}}$ pin is falling-edge-sensitive only. With MODE clear, a vector fetch or software clear immediately clears the IRQ latch.

The IRQF bit in the INTSCR register can be used to check for pending interrupts. The IRQF bit is not affected by the IMASK bit, which makes it useful in applications where polling is preferred.

Use the BIH or BIL instruction to read the logic level on the $\overline{\text{IRQ}}$ pin.

NOTE
When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine.

8.5 IRQ Module During Break Interrupts

The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear the latch during the break state. See Chapter 20 Development Support.

To allow software to clear the IRQ latch during a break interrupt, write a logic 1 to the BCFE bit. If a latch is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect CPU interrupt flags during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), writing to the ACK bit in the IRQ status and control register during the break state has no effect on the IRQ interrupt flags.
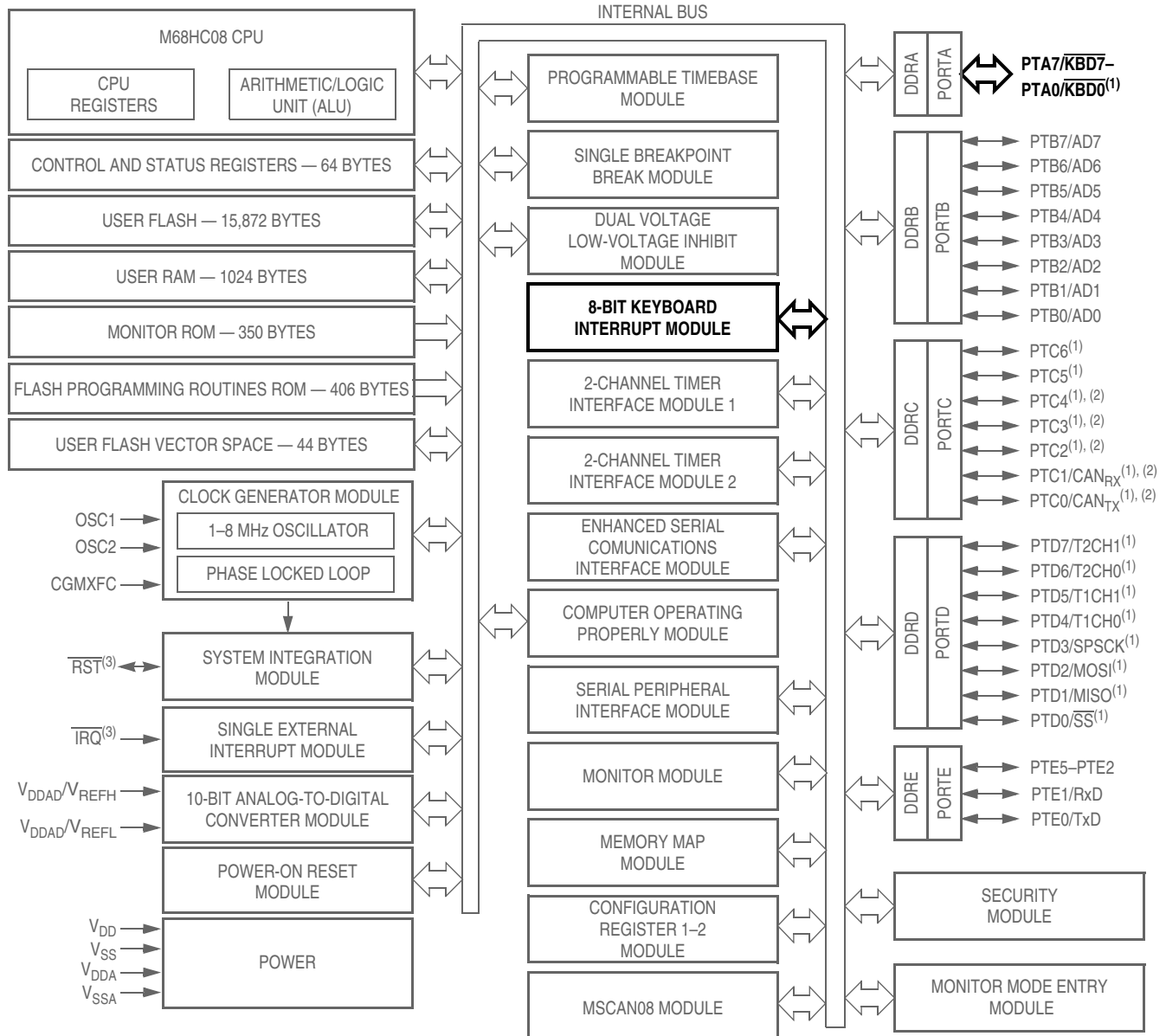
Figure 9-1. Block Diagram Highlighting KBI Block and Pins

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

Chapter 11
Low-Voltage Inhibit (LVI)

11.1 Introduction

This section describes the low-voltage inhibit (LVI) module, which monitors the voltage on the V_{DD} pin and can force a reset when the V_{DD} voltage falls below the LVI trip falling voltage, V_{TRIPF}.

11.2 Features

Features of the LVI module include:
- Programmable LVI reset
- Selectable LVI trip voltage
- Programmable stop mode operation

11.3 Functional Description

Figure 11-1 shows the structure of the LVI module. The LVI is enabled out of reset. The LVI module contains a bandgap reference circuit and comparator. Clearing the LVI power disable bit, LVIPWRD, enables the LVI to monitor V_{DD} voltage. Clearing the LVI reset disable bit, LVIRSTD, enables the LVI module to generate a reset when V_{DD} falls below a voltage, V_{TRIPF}. Setting the LVI enable in stop mode bit, LVISTOP, enables the LVI to operate in stop mode. Setting the LVI 5-V or 3-V trip point bit, LVI5OR3, enables the trip point voltage, V_{TRIPF}, to be configured for 5-V operation. Clearing the LVI5OR3 bit enables the trip point voltage, V_{TRIPF}, to be configured for 3-V operation. The actual trip points are shown in Chapter 21 Electrical Specifications.

> ### NOTE
> *After a power-on reset (POR) the LVI's default mode of operation is 3 V. If a 5-V system is used, the user must set the LVI5OR3 bit to raise the trip point to 5-V operation. Note that this must be done after every power-on reset since the default will revert back to 3-V mode after each power-on reset. If the V_{DD} supply is below the 5-V mode trip voltage but above the 3-V mode trip voltage when POR is released, the part will operate because V_{TRIPF} defaults to 3-V mode after a POR. So, in a 5-V system care must be taken to ensure that V_{DD} is above the 5-V mode trip voltage after POR is released.*
>
> *If the user requires 5-V mode and sets the LVI5OR3 bit after a power-on reset while the V_{DD} supply is not above the V_{TRIPR} for 5-V mode, the microcontroller unit (MCU) will immediately go into reset. The LVI in this case will hold the part in reset until either V_{DD} goes above the rising 5-V trip point, V_{TRIPR}, which will release reset or V_{DD} decreases to approximately 0 V which will re-trigger the power-on reset and reset the trip point to 3-V operation.*

SLPAK — Sleep Mode Acknowledge

This flag indicates whether the MSCAN08 is in module internal sleep mode. It shall be used as a handshake for the sleep mode request (see 12.8.1 MSCAN08 Sleep Mode). If the MSCAN08 detects bus activity while in sleep mode, it clears the flag.

    1 = Sleep – MSCAN08 in internal sleep mode

    0 = Wakeup – MSCAN08 is not in sleep mode

SLPRQ — Sleep Request, Go to Internal Sleep Mode

This flag requests the MSCAN08 to go into an internal power-saving mode (see 12.8.1 MSCAN08 Sleep Mode).

    1 = Sleep — The MSCAN08 will go into internal sleep mode.

    0 = Wakeup — The MSCAN08 will function normally.

SFTRES — Soft Reset

When this bit is set by the CPU, the MSCAN08 immediately enters the soft reset state. Any ongoing transmission or reception is aborted and synchronization to the bus is lost.

The following registers enter and stay in their hard reset state:

        CMCR0, CRFLG, CRIER, CTFLG, and CTCR.

The registers CMCR1, CBTR0, CBTR1, CIDAC, CIDAR0–CIDAR3, and CIDMR0–CIDMR3 can only be written by the CPU when the MSCAN08 is in soft reset state. The values of the error counters are not affected by soft reset.

When this bit is cleared by the CPU, the MSCAN08 tries to synchronize to the CAN bus. If the MSCAN08 is not in bus-off state, it will be synchronized after 11 recessive bits on the bus; if the MSCAN08 is in bus-off state, it continues to wait for 128 occurrences of 11 recessive bits.

Clearing SFTRES and writing to other bits in CMCR0 must be in separate instructions.

    1 = MSCAN08 in soft reset state

    0 = Normal operation

12.13.2 MSCAN08 Module Control Register 1

Address: $0501

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	LOOPB	WUPM	CLKSRC
Write:								
Reset:	0	0	0	0	0	0	0	0

    = Unimplemented

Figure 12-17. Module Control Register (CMCR1)

LOOPB — Loop Back Self-Test Mode

When this bit is set, the MSCAN08 performs an internal loop back which can be used for self-test operation: the bit stream output of the transmitter is fed back to the receiver internally. The CAN_{RX} input pin is ignored and the CAN_{TX} output goes to the recessive state (logic 1). The MSCAN08 behaves as it does normally when transmitting and treats its own transmitted message as a message received from a remote node. In this state the MSCAN08 ignores the bit sent during the ACK slot of the CAN frame Acknowledge field to insure proper reception of its own message. Both transmit and receive interrupts are generated.

    1 = Activate loop back self-test mode

    0 = Normal operation

Table 12-8. Time Segment Values

TSEG13	TSEG12	TSEG11	TSEG10	Time Segment 1		TSEG22	TSEG21	TSEG20	Time Segment 2
0	0	0	0	1 T_q Cycle[1]		0	0	0	1 T_q Cycle[1]
0	0	0	1	2 T_q Cycles[1]		0	0	1	2 T_q Cycles
0	0	1	0	3T_q Cycles[1]		.	.	.	.
0	0	1	1	4 T_q Cycles		.	.	.	.
.	.	.	.	.		1	1	1	8T_q Cycles
.	.	.	.	.					
1	1	1	1	16 T_q Cycles					

1. This setting is not valid. Please refer to Table 12-4 for valid settings.

12.13.5 MSCAN08 Receiver Flag Register (CRFLG)

All bits of this register are read and clear only. A flag can be cleared by writing a 1 to the corresponding bit position. A flag can be cleared only when the condition which caused the setting is valid no more. Writing a 0 has no effect on the flag setting. Every flag has an associated interrupt enable flag in the CRIER register. A hard or soft reset will clear the register.

	Bit 7	6	5	4	3	2	1	Bit 0
Address: $0504								
Read: Write:	WUPIF	RWRNIF	TWRNIF	RERRIF	TERRIF	BOFFIF	OVRIF	RXF
Reset:	0	0	0	0	0	0	0	0

Figure 12-20. Receiver Flag Register (CRFLG)

WUPIF — Wakeup Interrupt Flag
   If the MSCAN08 detects bus activity while in sleep mode, it sets the WUPIF flag. If not masked, a wakeup interrupt is pending while this flag is set.
      1 = MSCAN08 has detected activity on the bus and requested wakeup.
      0 = No wakeup interrupt has occurred.

RWRNIF — Receiver Warning Interrupt Flag
   This flag is set when the MSCAN08 goes into warning status due to the receive error counter (REC) exceeding 96 and neither one of the error interrupt flags or the bus-off interrupt flag is set[1]. If not masked, an error interrupt is pending while this flag is set.
      1 = MSCAN08 has gone into receiver warning status.
      0 = No receiver warning status has been reached.

---

1. Condition to set the flag: RWRNIF = (96 → REC) & $\overline{RERRIF}$ & $\overline{TERRIF}$ & $\overline{BOFFIF}$

13.5.3 Port C Input Pullup Enable Register

The port C input pullup enable register (PTCPUE) contains a software configurable pullup device for each of the seven port C pins. Each bit is individually configurable and requires that the data direction register, DDRC, bit be configured as an input. Each pullup is automatically and dynamically disabled when a port bit's DDRC is configured for output mode.

Address:	$000E							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PTCPUE6	PTCPUE5	PTCPUE4	PTCPUE3	PTCPUE2	PTCPUE1	PTCPUE0
Write:		PTCPUE6	PTCPUE5	PTCPUE4	PTCPUE3	PTCPUE2	PTCPUE1	PTCPUE0
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 13-12. Port C Input Pullup Enable Register (PTCPUE)

PTCPUE6–PTCPUE0 — Port C Input Pullup Enable Bits
These writable bits are software programmable to enable pullup devices on an input port bit.
1 = Corresponding port C pin configured to have internal pullup
0 = Corresponding port C pin internal pullup disconnected

13.6 Port D

Port D is an 8-bit special-function port that shares four of its pins with the serial peripheral interface (SPI) module and four of its pins with the two timer interface (TIM1 and TIM2) modules. Port D also has software configurable pullup devices if configured as an input port.

13.6.1 Port D Data Register

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

Address:	$0003							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
Write:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
Reset:				Unaffected by reset				
Alternative Function:	T2CH1	T2CH0	$\overline{\text{T1CH1}}$	T1CH0	SPSCK	$\overline{\text{MOSI}}$	MISO	$\overline{\text{SS}}$

Figure 13-13. Port D Data Register (PTD)

PTD7–PTD0 — Port D Data Bits
These read/write bits are software-programmable. Data direction of each port D pin is under the control of the corresponding bit in data direction register D. Reset has no effect on port D data.

T2CH1 and T2CH0 — Timer 2 Channel I/O Bits
The PTD7/T2CH1–PTD6/T2CH0 pins are the TIM2 input capture/output compare pins. The edge/level select bits, ELSxB:ELSxA, determine whether the PTD7/T2CH1–PTD6/T2CH0 pins are timer channel I/O pins or general-purpose I/O pins. See Chapter 19 Timer Interface Module (TIM).

T1CH1 and T1CH0 — Timer 1 Channel I/O Bits

The PTD7/T1CH1–PTD6/T1CH0 pins are the TIM1 input capture/output compare pins. The edge/level select bits, ELSxB and ELSxA, determine whether the PTD7/T1CH1–PTD6/T1CH0 pins are timer channel I/O pins or general-purpose I/O pins. See Chapter 19 Timer Interface Module (TIM).

SPSCK — SPI Serial Clock

The PTD3/SPSCK pin is the serial clock input of the SPI module. When the SPE bit is clear, the PTD3/SPSCK pin is available for general-purpose I/O.

MOSI — Master Out/Slave In

The PTD2/MOSI pin is the master out/slave in terminal of the SPI module. When the SPE bit is clear, the PTD2/MOSI pin is available for general-purpose I/O.

MISO — Master In/Slave Out

The PTD1/MISO pin is the master in/slave out terminal of the SPI module. When the SPI enable bit, SPE, is clear, the SPI module is disabled, and the PTD0/$\overline{SS}$ pin is available for general-purpose I/O.

Data direction register D (DDRD) does not affect the data direction of port D pins that are being used by the SPI module. However, the DDRD bits always determine whether reading port D returns the states of the latches or the states of the pins. See Table 13-5.

$\overline{SS}$ — Slave Select

The PTD0/$\overline{SS}$ pin is the slave select input of the SPI module. When the SPE bit is clear, or when the SPI master bit, SPMSTR, is set, the PTD0/$\overline{SS}$ pin is available for general-purpose I/O. When the SPI is enabled, the DDRB0 bit in data direction register B (DDRB) has no effect on the PTD0/$\overline{SS}$ pin.

13.6.2 Data Direction Register D

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

Address:	$0007							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Reset:	0	0	0	0	0	0	0	0

Figure 13-14. Data Direction Register D (DDRD)

DDRD7–DDRD0 — Data Direction Register D Bits

These read/write bits control port D data direction. Reset clears DDRD7–DDRD0, configuring all port D pins as inputs.
1 = Corresponding port D pin configured as output
0 = Corresponding port D pin configured as input

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

Figure 13-15 shows the port D I/O logic.

15.3 Pin Name Conventions

The generic names of the ESCI input/output (I/O) pins are:
- RxD (receive data)
- TxD (transmit data)

ESCI I/O lines are implemented by sharing parallel I/O port pins. The full name of an ESCI input or output reflects the name of the shared port pin. Table 15-1 shows the full names and the generic names of the ESCI I/O pins. The generic pin names appear in the text of this section.

Table 15-1. Pin Name Conventions

Generic Pin Names	RxD	TxD
Full Pin Names	PTE1/RxD	PTE0/TxD

15.4 Functional Description

Figure 15-2 shows the structure of the ESCI module. The ESCI allows full-duplex, asynchronous, NRZ serial communication between the MCU and remote devices, including other MCUs. The transmitter and receiver of the ESCI operate independently, although they use the same baud rate generator. During normal operation, the CPU monitors the status of the ESCI, writes the data to be transmitted, and processes received data.

The baud rate clock source for the ESCI can be selected via the configuration bit, ESCIBDSRC, of the CONFIG2 register ($001E).

For reference, a summary of the ESCI module input/output registers is provided in Figure 15-3.

SL = 1 -> SCI_CLK = BUSCLK
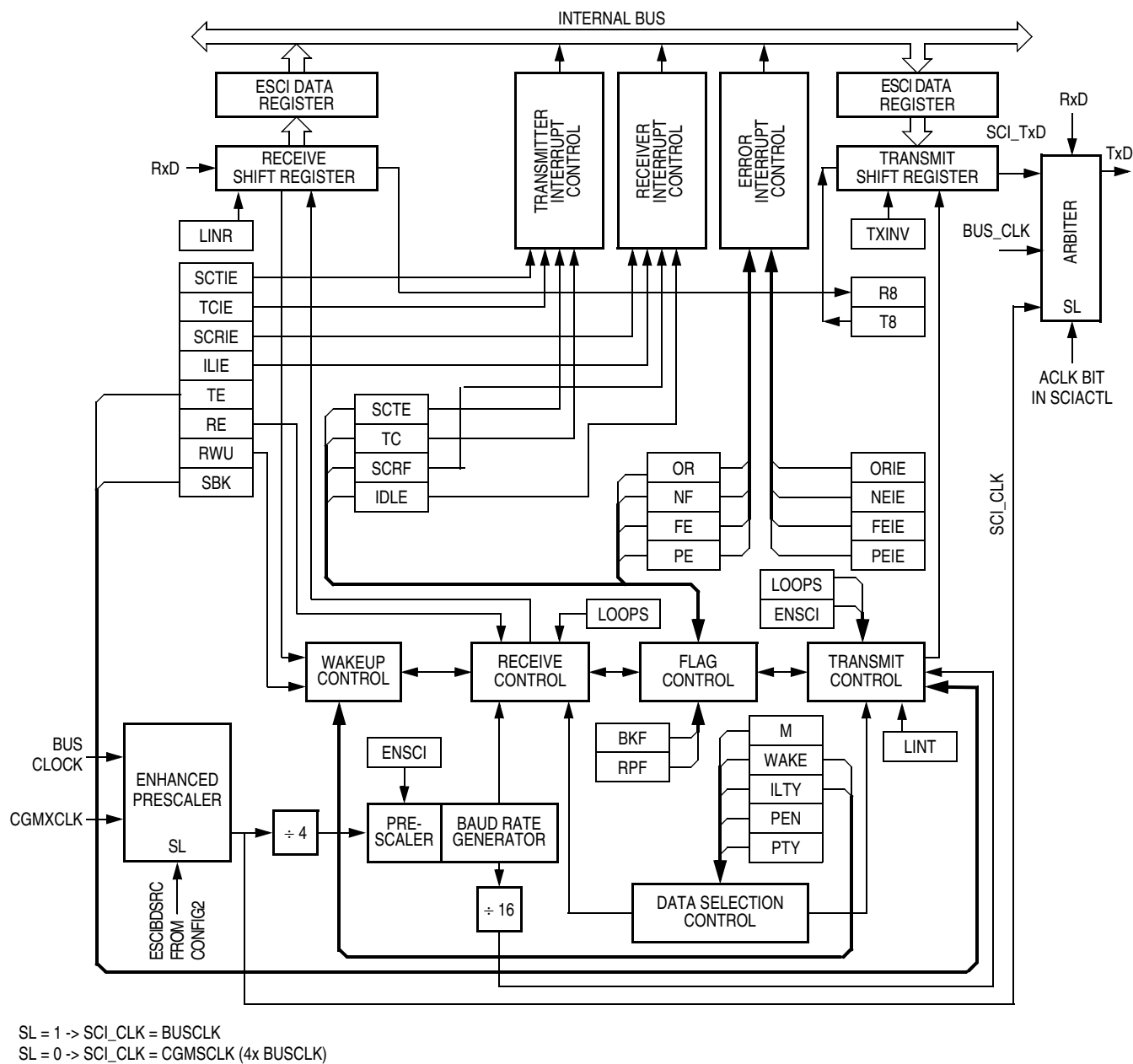SL = 0 -> SCI_CLK = CGMSCLK (4x BUSCLK)

Figure 15-2. ESCI Module Block Diagram

PTY — Parity Bit

This read/write bit determines whether the ESCI generates and checks for odd parity or even parity (see Table 15-5). Reset clears the PTY bit.

   1 = Odd parity
   0 = Even parity

> ***NOTE***
>
> *Changing the PTY bit in the middle of a transmission or reception can generate a parity error.*

15.8.2 ESCI Control Register 2

ESCI control register 2 (SCC2):
- Enables these CPU interrupt requests:
  – SCTE bit to generate transmitter CPU interrupt requests
  – TC bit to generate transmitter CPU interrupt requests
  – SCRF bit to generate receiver CPU interrupt requests
  – IDLE bit to generate receiver CPU interrupt requests
- Enables the transmitter
- Enables the receiver
- Enables ESCI wakeup
- Transmits ESCI break characters

Address: $0014

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0

Figure 15-11. ESCI Control Register 2 (SCC2)

SCTIE — ESCI Transmit Interrupt Enable Bit

This read/write bit enables the SCTE bit to generate ESCI transmitter CPU interrupt requests. Setting the SCTIE bit in SCC2 enables the SCTE bit to generate CPU interrupt requests. Reset clears the SCTIE bit.

   1 = SCTE enabled to generate CPU interrupt
   0 = SCTE not enabled to generate CPU interrupt

TCIE — Transmission Complete Interrupt Enable Bit

This read/write bit enables the TC bit to generate ESCI transmitter CPU interrupt requests. Reset clears the TCIE bit.

   1 = TC enabled to generate CPU interrupt requests
   0 = TC not enabled to generate CPU interrupt requests

SCRIE — ESCI Receive Interrupt Enable Bit

This read/write bit enables the SCRF bit to generate ESCI receiver CPU interrupt requests. Setting the SCRIE bit in SCC2 enables the SCRF bit to generate CPU interrupt requests. Reset clears the SCRIE bit.

   1 = SCRF enabled to generate CPU interrupt
   0 = SCRF not enabled to generate CPU interrupt
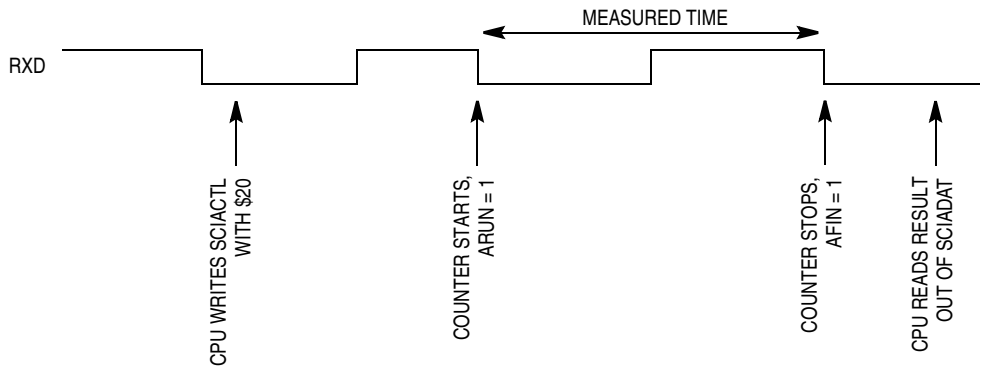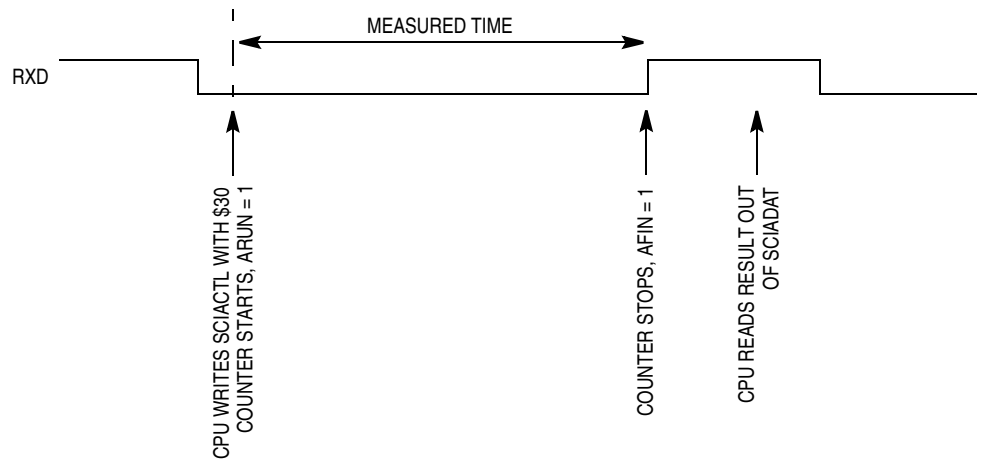
Figure 15-21. Bit Time Measurement with ACLK = 0

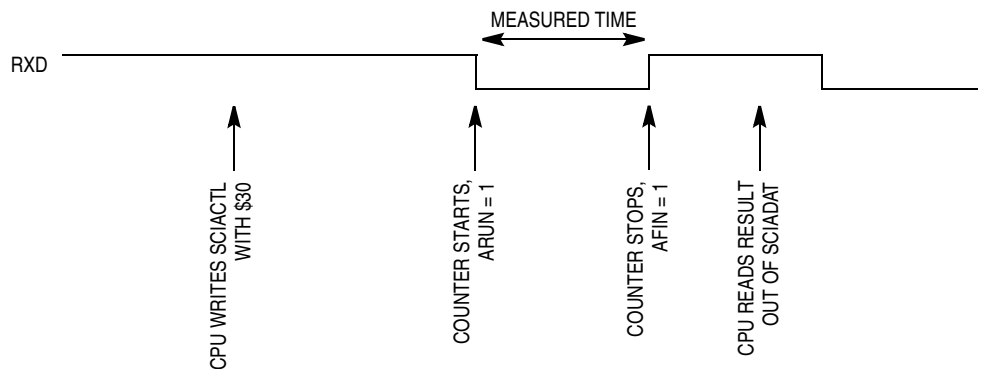**Figure 15-22. Bit Time Measurement with ACLK = 1, Scenario A**

Figure 15-23. Bit Time Measurement with ACLK = 1, Scenario B

MC68HC908GZ16 • MC68HC908GZ8 Data Sheet, Rev. 4

16.2 SIM Bus Clock Control and Generation

The bus clock generator provides system clock signals for the CPU and peripherals on the MCU. The system clocks are generated from an incoming clock, CGMOUT, as shown in Figure 16-3. This clock originates from either an external oscillator or from the on-chip PLL.

16.2.1 Bus Timing

In user mode, the internal bus frequency is either the crystal oscillator output (CGMXCLK) divided by four or the PLL output (CGMVCLK) divided by four.

16.2.2 Clock Startup from POR or LVI Reset

When the power-on reset module or the low-voltage inhibit module generates a reset, the clocks to the CPU and peripherals are inactive and held in an inactive phase until after the 4096 CGMXCLK cycle POR timeout has completed. The $\overline{RST}$ pin is driven low by the SIM during this entire period. The IBUS clocks start upon completion of the timeout.
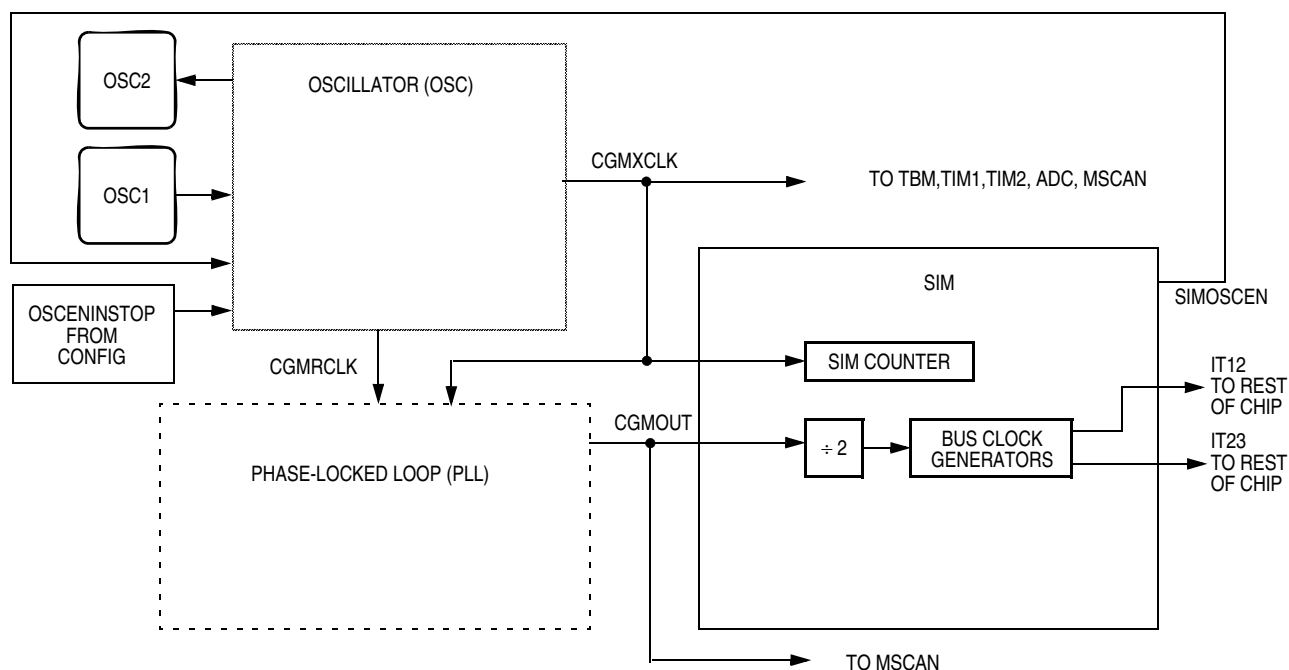

Figure 16-3. System Clock Signals

16.2.3 Clocks in Stop Mode and Wait Mode

Upon exit from stop mode by an interrupt or reset, the SIM allows CGMXCLK to clock the SIM counter. The CPU and peripheral clocks do not become active until after the stop delay timeout. This timeout is selectable as 4096 or 32 CGMXCLK cycles. See 16.6.2 Stop Mode.

In wait mode, the CPU clocks are inactive. The SIM also produces two sets of clocks for other modules. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.
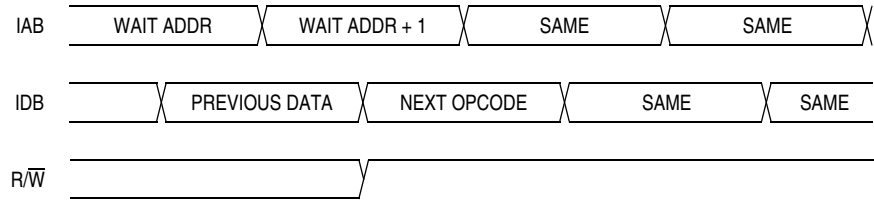
Note: Previous data can be operand data or the WAIT opcode, depending on the last instruction.

Figure 16-15. Wait Mode Entry Timing

Figure 16-16 and Figure 16-17 show the timing for WAIT recovery.
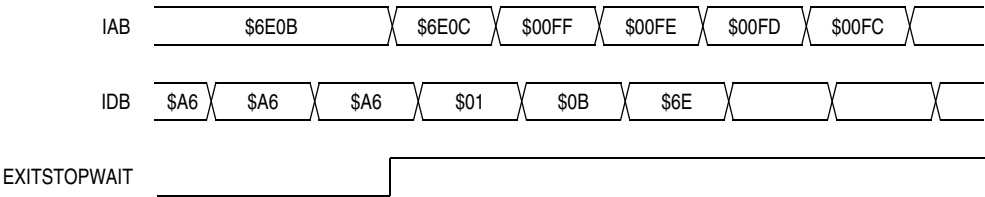
Note: EXITSTOPWAIT = $\overline{\text{RST}}$ pin or CPU interrupt
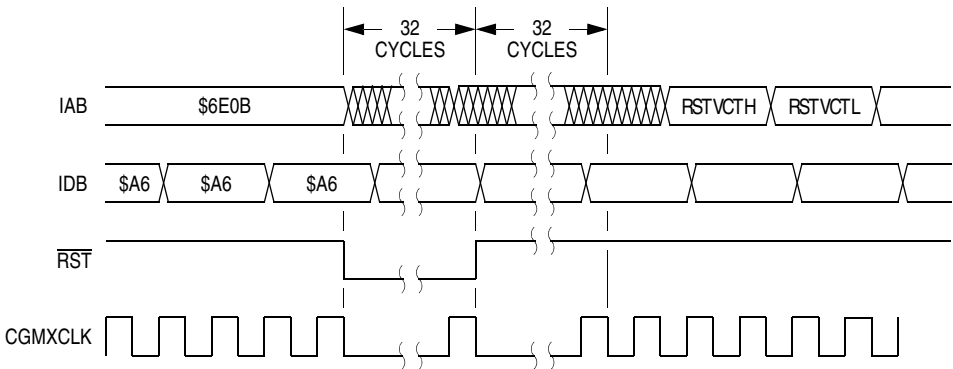
Figure 16-16. Wait Recovery from Interrupt

Figure 16-17. Wait Recovery from Internal Reset

16.6.2 Stop Mode

In stop mode, the SIM counter is reset and the system clocks are disabled. An interrupt request from a module can cause an exit from stop mode. Stacking for interrupts begins after the selected stop recovery time has elapsed. Reset causes an exit from stop mode.

The SIM disables the clock generator module outputs (CGMOUT and CGMXCLK) in stop mode, stopping the CPU and peripherals. Stop recovery time is selectable using the SSREC bit in the mask option register (MOR). If SSREC is set, stop recovery is reduced from the normal delay of 4096 CGMXCLK cycles down to 32. This is ideal for applications using canned oscillators that do not require long startup times from stop mode.

The following sources in the SPI status and control register can generate CPU interrupt requests:

- SPI receiver full bit (SPRF) — The SPRF bit becomes set every time a byte transfers from the shift register to the receive data register. If the SPI receiver interrupt enable bit, SPRIE, is also set, SPRF generates an SPI receiver/error CPU interrupt request.
- SPI transmitter empty (SPTE) — The SPTE bit becomes set every time a byte transfers from the transmit data register to the shift register. If the SPI transmit interrupt enable bit, SPTIE, is also set, SPTE generates an SPTE CPU interrupt request.

17.9 Resetting the SPI

Any system reset completely resets the SPI. Partial resets occur whenever the SPI enable bit (SPE) is low. Whenever SPE is low, the following occurs:

- The SPTE flag is set.
- Any transmission currently in progress is aborted.
- The shift register is cleared.
- The SPI state counter is cleared, making it ready for a new complete transmission.
- All the SPI port logic is defaulted back to being general-purpose I/O.

These items are reset only by a system reset:

- All control bits in the SPCR register
- All control bits in the SPSCR register (MODFEN, ERRIE, SPR1, and SPR0)
- The status flags SPRF, OVRF, and MODF

By not resetting the control bits when SPE is low, the user can clear SPE between transmissions without having to set all control bits again when SPE is set back high for the next transmission.

By not resetting the SPRF, OVRF, and MODF flags, the user can still service these interrupts after the SPI has been disabled. The user can disable the SPI by writing 0 to the SPE bit. The SPI can also be disabled by a mode fault occurring in an SPI that was configured as a master with the MODFEN bit set.

17.10 Low-Power Modes

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

17.10.1 Wait Mode

The SPI module remains active after the execution of a WAIT instruction. In wait mode the SPI module registers are not accessible by the CPU. 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.

To exit wait mode when an overflow condition occurs, enable the OVRF bit to generate CPU interrupt requests by setting the error interrupt enable bit (ERRIE). See 17.8 Interrupts.

17.10.2 Stop Mode

The SPI module is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions. 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.
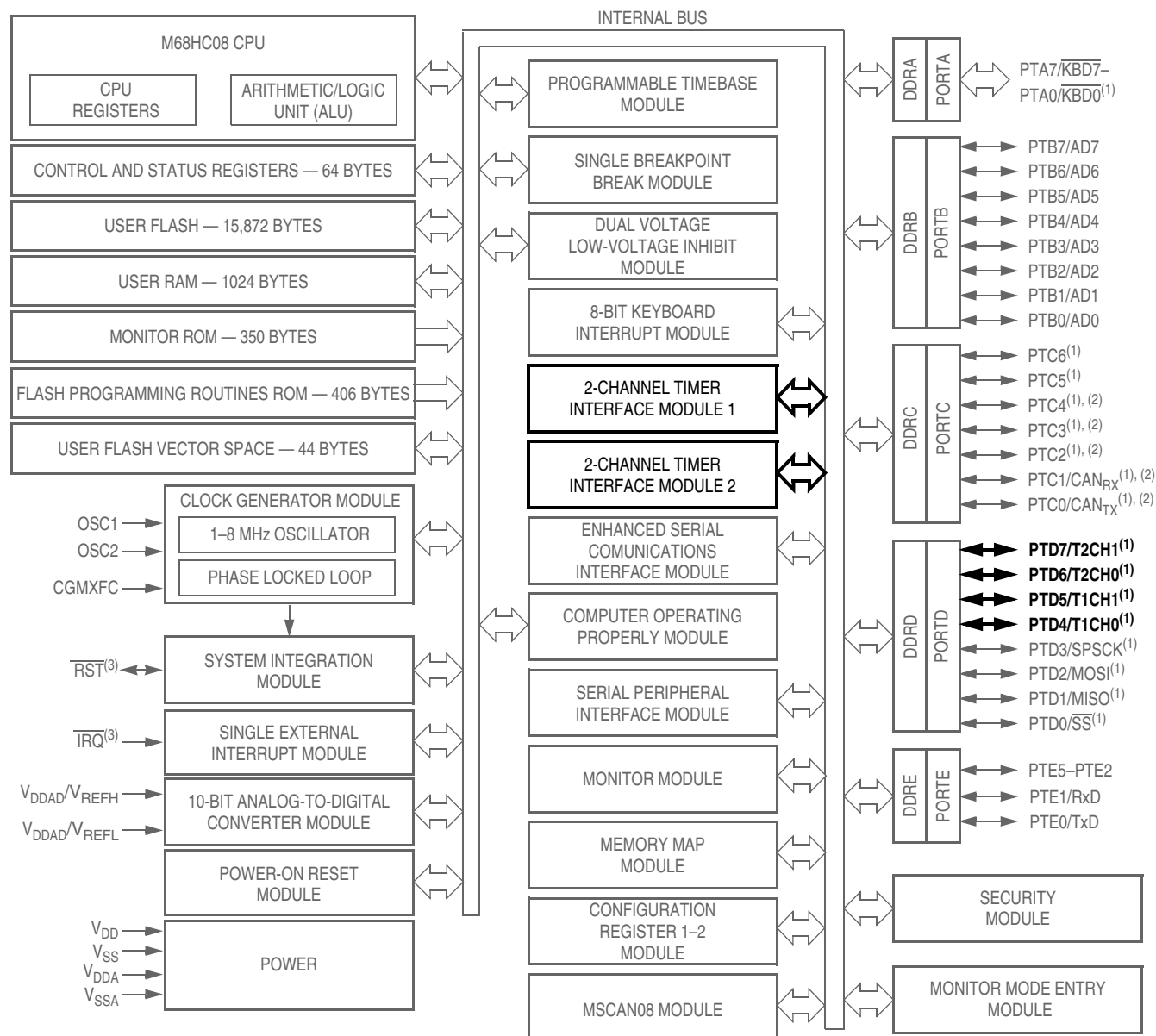
1. Ports are software configurable with pullup device if input port.
2. Higher current drive port pins
3. Pin contains integrated pullup device

Figure 19-2. Block Diagram Highlighting TIM Block and Pins

21.15 Memory Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
RAM data retention voltage	V_{RDR}	1.3	—	—	V
FLASH program bus clock frequency	—	1	—	—	MHz
FLASH read bus clock frequency	f_{Read}[1]	0	—	8 M	Hz

FLASH page erase time
   Limited endurance (<1 K cycles)
   Maximum endurance (>1 K cycles)
	t_{Erase}[2]	0.9
		3.6
			1
			4
				1.1
				5.5
					ms

FLASH mass erase time	t_{MErase}[3]	4	—	—	ms
FLASH PGM/ERASE to HVEN setup time	t_{NVS}	10	—	—	μs
FLASH high-voltage hold time	t_{NVH}	5	—	—	μs
FLASH high-voltage hold time (mass erase)	t_{NVHL}	100	—	—	μs
FLASH program hold time	t_{PGS}	5	—	—	μs
FLASH program time	t_{PROG}	30	—	40	μs
FLASH return to read time	t_{RCV}[4]	1	—	—	μs
FLASH cumulative program hv period	t_{HV}[5]	—	—	4	ms
FLASH endurance[6]	—	10 k	100 k	—	Cycles
FLASH data retention time[7]	—	15	100	—	Years

1. f_{Read} is defined as the frequency range for which the FLASH memory can be read.
2. If the page erase time is longer than t_{Erase} (min), there is no erase disturb, but it reduces the endurance of the FLASH memory.
3. If the mass erase time is longer than t_{MErase} (min), there is no erase disturb, but it reduces the endurance of the FLASH memory.
4. t_{RCV} is defined as the time it needs before the FLASH can be read after turning off the high voltage charge pump, by clearing HVEN to 0.
5. t_{HV} is defined as the cumulative high voltage programming time to the same row before next erase.
   t_{HV} must satisfy this condition: $t_{NVS} + t_{NVH} + t_{PGS} + (t_{PROG}$ x 32$) \leq t_{HV}$ maximum.
6. Typical endurance was evaluated for this product family. For additional information on how Freescale Semiconductor defines *Typical Endurance*, please refer to Engineering Bulletin EB619.
7. Typical data retention values are based on intrinsic capability of the technology measured at high temperature and de-rated to 25°C using the Arrhenius equation. For additional information on how Freescale Semiconductor defines *Typical Data Retention*, please refer to Engineering Bulletin EB618.