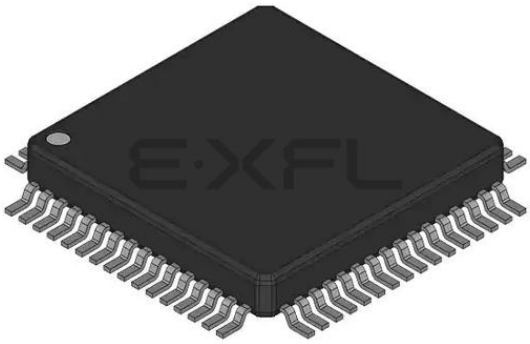


Welcome to E-XFL.COM

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	8.4MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	52
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 15x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908az60avfue

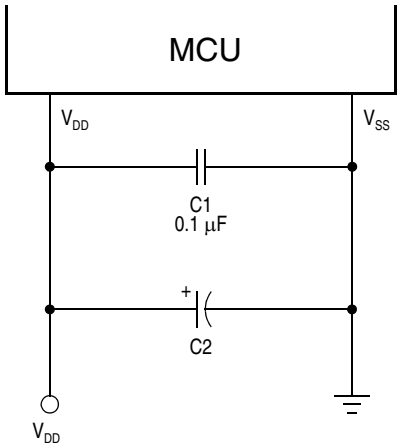
NOTE

The following pin descriptions are just a quick reference. For a more detailed representation, see Chapter 22 Input/Output Ports.

1.4.1 Power Supply Pins (V_{DD} and V_{SS})

V_{DD} and V_{SS} are the power supply and ground pins. The MCU operates from a single power supply.

Fast signal transitions on MCU pins place high, short-duration current demands on the power supply. To prevent noise problems, take special care to provide power supply bypassing at the MCU as shown in Figure 1-6. Place the C1 bypass capacitor as close to the MCU as possible. Use a high-frequency response ceramic capacitor for C1. C2 is an optional bulk current bypass capacitor for use in applications that require the port pins to source high current levels.



NOTE: Component values shown represent typical applications.

Figure 1-6. Power Supply Bypassing

V_{SS} is also the ground for the port output buffers and the ground return for the serial clock in the Serial Peripheral Interface module (SPI). See Chapter 19 Serial Peripheral Interface (SPI).

NOTE

V_{SS} must be grounded for proper MCU operation.

1.4.2 Oscillator Pins (OSC1 and OSC2)

The OSC1 and OSC2 pins are the connections for the on-chip oscillator circuit. See Chapter 10 Clock Generator Module (CGM).

1.4.3 External Reset Pin ($\overline{RST}$)

A 0 on the $\overline{RST}$ pin forces the MCU to a known startup state. $\overline{RST}$ is bidirectional, allowing a reset of the entire system. It is driven low when any internal reset source is asserted. See Chapter 9 System Integration Module (SIM) for more information.

1.4.4 External Interrupt Pin ($\overline{IRQ}$)

$\overline{IRQ}$ is an asynchronous external interrupt pin. See Chapter 17 External Interrupt Module (IRQ).

Table 1-1. External Pins Summary (Continued)

Pin Name	Function	Driver Type	Hysteresis[1]	Reset State
PTF0/TACH2	General-Purpose I/O Timer A Channel 2	Dual State	Yes	Input Hi-Z
PTG2/KBD2–PTG0/KBD0	General-Purpose I/O/ Keyboard Wakeup Pin	Dual State	Yes	Input Hi-Z
PTH1/KBD4 –PTH0/KBD3	General-Purpose I/O/ Keyboard Wakeup Pin	Dual State	Yes	Input Hi-Z
V_{DD}	Chip Power Supply	N/A	N/A	N/A
V_{SS}	Chip Ground	N/A	N/A	N/A
V_{DDA}	CGM Analog Power Supply			
V_{SSA}	CGM Analog Ground			
V_{DDAREF}	ADC Power Supply	N/A	N/A	N/A
A_{VSS}/V_{REFL}	ADC Ground/ ADC Reference Low Voltage	N/A	N/A	N/A
V_{REFH}	A/D Reference High Voltage	N/A	N/A	N/A
OSC1	External Clock In	N/A	No	Input Hi-Z
OSC2	External Clock Out	N/A	N/A	Output
CGMXFC	PLL Loop Filter Cap	N/A	N/A	N/A
$\overline{IRQ}$	External Interrupt Request	N/A	N/A	Input Hi-Z
$\overline{RST}$	Reset	N/A	N/A	Output Low
CANRx	CAN Serial Input	N/A	Yes	Input Hi-Z
CANTx	CAN Serial Output	Output	No	Output
BDRxD	BDLC Serial Input	N/A	Yes	Input Hi-Z
BDTxD	BDLC Serial Output	Output	No	Output

1. Hysteresis is not 100% tested but is typically a minimum of 300 mV.

Chapter 6
EEPROM-1 Memory

6.1 Introduction

This chapter describes the 512 bytes of electrically erasable programmable read-only memory (EEPROM) residing at address range $0800 to $09FF. There are 1024 bytes of EEPROM available on the MC68HC908AS60A and MC68HC908AZ60A which are physically located in two 512 byte arrays. For information relating to the array covering address range $0600 to $07FF please see Chapter 7 EEPROM-2 Memory.

6.2 Features

Features of the EEPROM-1 include the following:

- 512 bytes Nonvolatile Memory
- Byte, Block, or Bulk Erasable
- Nonvolatile EEPROM Configuration and Block Protection Options
- On-chip Charge Pump for Programming/Erasing
- Security Option
- AUTO Bit Driven Programming/Erasing Time Feature

6.3 EEPROM-1 Register Summary

The EEPROM-1 Register Summary is shown in Figure 6-1.

6.4.5 EEPROM-1 Programming and Erasing

The unprogrammed or erase state of an EEPROM bit is a logic 1. The factory default for all bytes within the EEPROM-1 array is $FF.

The programming operation changes an EEPROM bit from logic 1 to logic 0 (programming cannot change a bit from logic 0 to a logic 1). In a single programming operation, the minimum EEPROM programming size is one bit; the maximum is eight bits (one byte).

The erase operation changes an EEPROM bit from logic 0 to logic 1. In a single erase operation, the minimum EEPROM erase size is one byte; the maximum is the entire EEPROM-1 array.

The EEPROM can be programmed such that one or multiple bits are programmed (written to a logic 0) at a time. However, the user may never program the same bit location more than once before erasing the entire byte. In other words, the user is not allowed to program a logic 0 to a bit that is already programmed (bit state is already logic 0).

For some applications it might be advantageous to track more than 10K events with a single byte of EEPROM by programming one bit at a time. For that purpose, a special selective bit programming technique is available. An example of this technique is illustrated in Table 6-2.

Table 6-2. Example Selective Bit Programming Description

Description	Program Data in Binary	Result in Binary
Original state of byte (erased)	n/a	1111:1111
First event is recorded by programming bit position 0	1111:1110	1111:1110
Second event is recorded by programming bit position 1	1111:1101	1111:1100
Third event is recorded by programming bit position 2	1111:1011	1111:1000
Fourth event is recorded by programming bit position 3	1111:0111	1111:0000
Events five through eight are recorded in a similar fashion		

Note that none of the bit locations are actually programmed more than once although the byte was programmed eight times.

When this technique is utilized, a program/erase cycle is defined as multiple program sequences (up to eight) to a unique location followed by a single erase operation.

6.4.5.1 Program/Erase Using AUTO Bit

An additional feature available for EEPROM-1 program and erase operations is the AUTO mode. When enabled, AUTO mode will activate an internal timer that will automatically terminate the program/erase cycle and clear the EEPGM bit. Please see 6.4.5.2 EEPROM-1 Programming, 6.4.5.3 EEPROM-1 Erasing, and 6.5.1 EEPROM-1 Control Register for more information.

7.4 Functional Description

The 512 bytes of EEPROM-2 are located at $0600-$07FF and can be programmed or erased without an additional external high voltage supply. The program and erase operations are enabled through the use of an internal charge pump. For each byte of EEPROM, the write/erase endurance is 10,000 cycles.

7.4.1 EEPROM-2 Configuration

The 8-bit EEPROM-2 Nonvolatile Register (EE2NVR) and the 16-bit EEPROM-2 Timebase Divider Nonvolatile Register (EE2DIVNVR) contain the default settings for the following EEPROM configurations:

- EEPROM-2 Timebase Reference
- EEPROM-2 Security Option
- EEPROM-2 Block Protection

EE2NVR and EE2DIVNVR are nonvolatile EEPROM registers. They are programmed and erased in the same way as EEPROM bytes. The contents of these registers are loaded into their respective volatile registers during a MCU reset. The values in these read/write volatile registers define the EEPROM-2 configurations.

For EE2NVR, the corresponding volatile register is the EEPROM-2 Array Configuration Register (EE2ACR). For the EE2DIVNCR (two 8-bit registers: EE2DIVHNVR and EE2DIVLNVR), the corresponding volatile register is the EEPROM-2 Divider Register (EE2DIV: EE2DIVH and EE2 DIVL).

7.4.2 EEPROM-2 Timebase Requirements

A 35μs timebase is required by the EEPROM-2 control circuit for program and erase of EEPROM content. This timebase is derived from dividing the CGMXCLK or bus clock (selected by EEDIVCLK bit in CONFIG-2 Register) using a timebase divider circuit controlled by the 16-bit EEPROM-2 Timebase Divider EE2DIV Register (EE2DIVH and EE2DIVL).

As the CGMXCLK or bus clock is user selected, the EEPROM-2 Timebase Divider Register must be configured with the appropriate value to obtain the 35 μs. The timebase divider value is calculated by using the following formula:

$$EE2DIV = INT[Reference\ Frequency(Hz) \times 35 \times 10^{-6} + 0.5]$$

This value is written to the EEPROM-2 Timebase Divider Register (EE2DIVH and EE2DIVL) or programmed into the EEPROM-2 Timebase Divider Nonvolatile Register prior to any EEPROM program or erase operations (7.4.1 EEPROM-2 Configuration and 7.4.2 EEPROM-2 Timebase Requirements).

8.3.5 Condition Code Register

The 8-bit condition code register contains the interrupt mask and five flags that indicate the results of the instruction just executed. Bits 6 and 5 are set permanently to 1. The following paragraphs describe the functions of the condition code register.

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	V	1	1	H	I	N	Z	C
Reset:	X	1	1	X	1	X	X	X

X = Indeterminate

Figure 8-6. Condition Code Register (CCR)

V — Overflow Flag

The CPU sets the overflow flag when a two's complement overflow occurs. The signed branch instructions BGT, BGE, BLE, and BLT use the overflow flag.

1 = Overflow
0 = No overflow

H — Half-Carry Flag

The CPU sets the half-carry flag when a carry occurs between accumulator bits 3 and 4 during an add-without-carry (ADD) or add-with-carry (ADC) operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations. The DAA instruction uses the states of the H and C flags to determine the appropriate correction factor.

1 = Carry between bits 3 and 4
0 = No carry between bits 3 and 4

I — Interrupt Mask

When the interrupt mask is set, all maskable CPU interrupts are disabled. CPU interrupts are enabled when the interrupt mask is cleared. When a CPU interrupt occurs, the interrupt mask is set automatically after the CPU registers are saved on the stack, but before the interrupt vector is fetched.

1 = Interrupts disabled
0 = Interrupts enabled

NOTE

To maintain M6805 Family compatibility, the upper byte of the index register (H) is not stacked automatically. If the interrupt service routine modifies H, then the user must stack and unstack H using the PSHH and PULH instructions.

After the I bit is cleared, the highest-priority interrupt request is serviced first.

A return-from-interrupt (RTI) instruction pulls the CPU registers from the stack and restores the interrupt mask from the stack. After any reset, the interrupt mask is set and can be cleared only by the clear interrupt mask software instruction (CLI).

N — Negative Flag

The CPU sets the negative flag when an arithmetic operation, logic operation, or data manipulation produces a negative result, setting bit 7 of the result.

1 = Negative result
0 = Non-negative result

The LDA opcode is prefetched by both the INT1 and INT2 RTI instructions. However, in the case of the INT1 RTI prefetch, this is a redundant operation.

NOTE

To maintain compatibility with the M68HC05, M6805 and M146805 Families the H register is not pushed on the stack during interrupt entry. If the interrupt service routine modifies the H register or uses the indexed addressing mode, software should save the H register and then restore it prior to exiting the routine.

9.5.1.2 SWI Instruction

The SWI instruction is a non-maskable instruction that causes an interrupt regardless of the state of the interrupt mask (I bit) in the condition code register.

NOTE

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

9.5.2 Reset

All reset sources always have higher priority than interrupts and cannot be arbitrated.

9.5.3 Break Interrupts

The break module can stop normal program flow at a software-programmable break point by asserting its break interrupt output. See Chapter 13 Break Module (BRK). The SIM puts the CPU into the break state by forcing it to the SWI vector location. Refer to the break interrupt subsection of each module to see how each module is affected by the break state.

9.5.4 Status Flag Protection in Break Mode

The SIM controls whether status flags contained in other modules can be cleared during break mode. The user can select whether flags are protected from being cleared by properly initializing the break clear flag enable bit (BCFE) in the SIM break flag control register (SBFCR).

Protecting flags in break mode ensures that set flags will not be cleared while in break mode. This protection allows registers to be freely read and written during break mode without losing status flag information.

Setting the BCFE bit enables the clearing mechanisms. Once cleared in break mode, a flag remains cleared even when break mode is exited. Status flags with a two-step clearing mechanism — for example, a read of one register followed by the read or write of another — are protected, even when the first step is accomplished prior to entering break mode. Upon leaving break mode, execution of the second step will clear the flag as normal.

9.6 Low-Power Modes

Executing the WAIT or STOP instruction puts the MCU in a low power- consumption mode for standby situations. The SIM holds the CPU in a non-clocked state. The operation of each of these modes is described below. Both STOP and WAIT clear the interrupt mask (I) in the condition code register, allowing interrupts to occur.

7. Using the value 4.9152 MHz for f_{NOM}, calculate the VCO linear range multiplier, L. The linear range multiplier controls the frequency range of the PLL.

$$L = \text{round}\left(\frac{f_{CGMVCLK}}{f_{NOM}}\right)$$

$$\text{Example: } L = \frac{32\text{ MHz}}{4.9152\text{ MHz}} = 7$$

8. Calculate the VCO center-of-range frequency, f_{CGMVRS}. The center-of-range frequency is the midpoint between the minimum and maximum frequencies attainable by the PLL.

   $f_{CGMVRS} = L \times f_{NOM}$

   Example: $f_{CGMVRS} = 7 \times 4.9152\text{ MHz} = 34.4\text{ MHz}$

   NOTE

   For proper operation,.

$$|f_{CGMVRS} - f_{CGMVCLK}| \leq \frac{f_{NOM}}{2}$$

   Exceeding the recommended maximum bus frequency or VCO frequency can crash the MCU.

9. Program the PLL registers accordingly:
   a. In the upper four bits of the PLL programming register (PPG), program the binary equivalent of N.
   b. In the lower four bits of the PLL programming register (PPG), program the binary equivalent of L.

10.3.2.5 Special Programming Exceptions

The programming method described in 10.3.2.4 Programming the PLL, does not account for two possible exceptions. A value of 0 for N or L is meaningless when used in the equations given. To account for these exceptions:

- A 0 value for N is interpreted the same as a value of 1.
- A 0 value for L disables the PLL and prevents its selection as the source for the base clock. See 10.3.3 Base Clock Selector Circuit.

10.3.3 Base Clock Selector Circuit

This circuit is used to select either the crystal clock, CGMXCLK, or the VCO clock, CGMVCLK, as the source of the base clock, CGMOUT. The two input clocks go through a transition control circuit that waits up to three CGMXCLK cycles and three CGMVCLK cycles to change from one clock source to the other. During this time, CGMOUT is held in stasis. The output of the transition control circuit is then divided by two to correct the duty cycle. Therefore, the bus clock frequency, which is one-half of the base clock frequency, is one-fourth the frequency of the selected clock (CGMXCLK or CGMVCLK).

The BCS bit in the PLL control register (PCTL) selects which clock drives CGMOUT. The VCO clock cannot be selected as the base clock source if the PLL is not turned on. The PLL cannot be turned off if the VCO clock is selected. The PLL cannot be turned on or off simultaneously with the selection or deselection of the VCO clock. The VCO clock also cannot be selected as the base clock source if the

NOTE

Software can select the CGMVCLK divided by two as the CGMOUT source even if the PLL is not locked (LOCK = 0). Therefore, software should make sure the PLL is locked before setting the BCS bit.

10.7 Low-Power Modes

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

10.7.1 Wait Mode

The CGM remains active in wait mode. Before entering wait mode, software can disengage and turn off the PLL by clearing the BCS and PLLON bits in the PLL control register (PCTL). Less power-sensitive applications can disengage the PLL without turning it off. Applications that require the PLL to wake the MCU from wait mode also can deselect the PLL output without turning off the PLL.

10.7.2 Stop Mode

The STOP instruction disables the CGM and holds low all CGM outputs (CGMXCLK, CGMOUT, and CGMINT).

If CGMOUT is being driven by CGMVCLK and a STOP instruction is executed; the PLL will clear the BCS bit in the PLL control register, causing CGMOUT to be driven by CGMXCLK. When the MCU recovers from STOP, the crystal clock divided by two drives CGMOUT and BCS remains clear.

10.8 CGM During Break Interrupts

The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state. See Chapter 13 Break Module (BRK).

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

To protect the PLLF bit during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), software can read and write the PLL control register during the break state without affecting the PLLF bit.

10.9 Acquisition/Lock Time Specifications

The acquisition and lock times of the PLL are, in many applications, the most critical PLL design parameters. Proper design and use of the PLL ensures the highest stability and lowest acquisition/lock times.

10.9.1 Acquisition/Lock Time Definitions

Typical control systems refer to the acquisition time or lock time as the reaction time, within specified tolerances, of the system to a step input. In a PLL, the step input occurs when the PLL is turned on or when it suffers a noise hit. The tolerance is usually specified as a percent of the step input or when the output settles to the desired value plus or minus a percent of the frequency change. Therefore, the reaction time is constant in this definition, regardless of the size of the step input. For example, consider a system with a 5% acquisition time tolerance. If a command instructs the system to change from 0 Hz to

In automatic bandwidth control mode, the acquisition and lock times are quantized into units based on the reference frequency. (See 10.3.2.3 Manual and Automatic PLL Bandwidth Modes). A certain number of clock cycles, n_{ACQ}, is required to ascertain that the PLL is within the tracking mode entry tolerance, Δ_{TRK}, before exiting acquisition mode. A certain number of clock cycles, n_{TRK}, is required to ascertain that the PLL is within the lock mode entry tolerance, Δ_{Lock}. Therefore, the acquisition time, t_{ACQ}, is an integer multiple of n_{ACQ}/f_{CGMRDV}, and the acquisition to lock time, t_{AL}, is an integer multiple of n_{TRK}/f_{CGMRDV}. Also, since the average frequency over the entire measurement period must be within the specified tolerance, the total time usually is longer than t_{Lock} as calculated above.

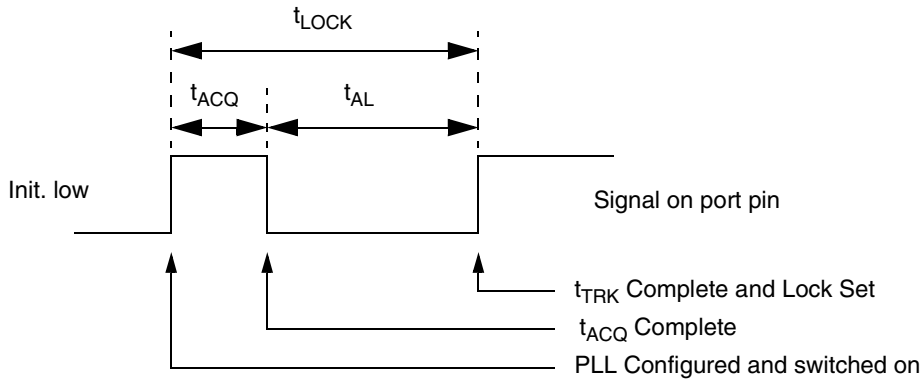
In manual mode, it is usually necessary to wait considerably longer than t_{Lock} before selecting the PLL clock (see 10.3.3 Base Clock Selector Circuit), because the factors described in 10.9.2 Parametric Influences on Reaction Time, may slow the lock time considerably.

When defining a limit in software for the maximum lock time, the value must allow for variation due to all of the factors mentioned in this chapter, especially due to the C_F capacitor and application specific influences.

The calculated lock time is only an indication and it is the customer's responsibility to allow enough of a guard band for their application. Prior to finalizing any software and while determining the maximum lock time, take into account all device to device differences. Typically, applications set the maximum lock time as an order of magnitude higher than the measured value. This is considered sufficient for all such device to device variation.

Freescale recommends measuring the lock time of the application system by utilizing dedicated software, running in FLASH, EEPROM or RAM. This should toggle a port pin when the PLL is first configured and switched on, then again when it goes from acquisition to lock mode and finally again when the PLL lock bit is set. The resultant waveform can be captured on an oscilloscope and used to determine the typical lock time for the microcontroller and the associated external application circuit.

For example,



NOTE

The filter capacitor should be fully discharged prior to making any measurements.

13.5.1 Break Status and Control Register

The break status and control register contains break module enable and status bits.

Address: $FE0E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BRKE	BRKA	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 13-3. Break Status and Control Register (BSCR)

BRKE — Break Enable Bit

This read/write bit enables breaks on break address register matches. Clear BRKE by writing a 0 to bit 7. Reset clears the BRKE bit.

1 = Breaks enabled on 16-bit address match
0 = Breaks disabled on 16-bit address match

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a 1 to BRKA generates a break interrupt. Clear BRKA by writing a 0 to it before exiting the break routine. Reset clears the BRKA bit.

1 = (When read) Break address match
0 = (When read) No break address match

13.5.2 Break Address Registers

The break address registers contain the high and low bytes of the desired breakpoint address. Reset clears the break address registers.

Register:	BRKH	BRKL						
Address:	$FE0C	$FE0D						
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	14	13	12	11	10	9	Bit 8
Write:								
Reset:	0	0	0	0	0	0	0	0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 13-4. Break Address Registers (BRKH and BRKL)

Chapter 17
External Interrupt Module (IRQ)

17.1 Introduction

This chapter describes the nonmaskable external interrupt (IRQ) input.

17.2 Features

Features include:

- Dedicated External Interrupt Pin ($\overline{\text{IRQ}}$)
- Hysteresis Buffer
- Programmable Edge-Only or Edge- and Level-Interrupt Sensitivity
- Automatic Interrupt Acknowledge

17.3 Functional Description

A falling edge applied to the external interrupt pin can latch a CPU interrupt request. Figure 17-1 shows the structure of the IRQ module.

Interrupt signals on the $\overline{\text{IRQ}}$ pin are latched into the IRQ latch. An interrupt latch remains set until one of the following actions occurs:

- Vector fetch — A vector fetch automatically generates an interrupt acknowledge signal that clears the latch that caused the vector fetch.
- Software clear — Software can clear an interrupt latch by writing to the appropriate acknowledge bit in the interrupt status and control register (ISCR). Writing a logic 1 to the ACK bit clears the IRQ latch.
- Reset — A reset automatically clears both interrupt latches.

18.7 I/O Signals

Port E shares two of its pins with the SCI module. The two SCI I/O pins are:

- PTE0/SCTxD — Transmit data
- PTE1/SCRxD — Receive data

18.7.1 PTE0/SCTxD (Transmit Data)

The PTE0/SCTxD pin is the serial data output from the SCI transmitter. The SCI shares the PTE0/SCTxD pin with port E. When the SCI is enabled, the PTE0/SCTxD pin is an output regardless of the state of the DDRE2 bit in data direction register E (DDRE).

18.7.2 PTE1/SCRxD (Receive Data)

The PTE1/SCRxD pin is the serial data input to the SCI receiver. The SCI shares the PTE1/SCRxD pin with port E. When the SCI is enabled, the PTE1/SCRxD pin is an input regardless of the state of the DDRE1 bit in data direction register E (DDRE).

18.8 I/O Registers

The following I/O registers control and monitor SCI operation:

- SCI control register 1 (SCC1)
- SCI control register 2 (SCC2)
- SCI control register 3 (SCC3)
- SCI status register 1 (SCS1)
- SCI status register 2 (SCS2)
- SCI data register (SCDR)
- SCI baud rate register (SCBR)

18.8.1 SCI Control Register 1

SCI control register 1:

- Enables loop mode operation
- Enables the SCI
- Controls output polarity
- Controls character length
- Controls SCI wakeup method
- Controls idle character detection
- Enables parity function
- Controls parity type

The generic names of the SPI I/O registers are:

- SPI control register (SPCR)
- SPI status and control register (SPSCR)
- SPI data register (SPDR)

Table 19-2 shows the names and the addresses of the SPI I/O registers.

Table 19-2. I/O Register Addresses

Register Name	Address
SPI Control Register (SPCR)	$0010
SPI Status and Control Register (SPSCR)	$0011
SPI Data Register (SPDR)	$0012

19.4 Functional Description

Figure 19-1 summarizes the SPI I/O registers and Figure 19-2 shows the structure of the SPI module.

Addr	Register Name	R/W	Bit 7	6	5	4	3	2	1	Bit 0
$0010	SPI Control Register (SPCR)	Read:	SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
		Write:								
		Reset:	0	0	1	0	1	0	0	0
$0011	SPI Status and Control Register (SPSCR)	Read:	SPRF	ERRIE	OVRF	MODF	SPTE	MODFEN	SPR1	SPR0
		Write:								
		Reset:	0	0	0	0	1	0	0	0
$0012	SPI Data Register (SPDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by Reset							

R = Reserved = Unimplemented

Figure 19-1. SPI I/O Register Summary

The SPI module allows full-duplex, synchronous, serial communication between the MCU and peripheral devices, including other MCUs. Software can poll the SPI status flags or SPI operation can be interrupt driven. All SPI interrupts can be serviced by the CPU.

The following paragraphs describe the operation of the SPI module.

24.7.1 Keyboard Status and Control Register

The keyboard status and control register:

- Flags keyboard interrupt requests
- Acknowledges keyboard interrupt requests
- Masks keyboard interrupt requests
- Controls keyboard interrupt triggering sensitivity

Address: $001B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
Write:						ACKK		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 24-3. Keyboard Status and Control Register (KBSCR)

Bits 7–4 — Not used

These read-only bits always read as logic 0s.

KEYF — Keyboard Flag Bit

This read-only bit is set when a keyboard interrupt is pending. Reset clears the KEYF bit.

1 = Keyboard interrupt pending
0 = No keyboard interrupt pending

ACKK — Keyboard Acknowledge Bit

Writing a logic 1 to this write-only bit clears the keyboard interrupt request. ACKK always reads as logic 0. Reset clears ACKK.

IMASKK — Keyboard Interrupt Mask Bit

Writing a logic 1 to this read/write bit prevents the output of the keyboard interrupt mask from generating interrupt requests. Reset clears the IMASKK bit.

1 = Keyboard interrupt requests masked
0 = Keyboard interrupt requests not masked

MODEK — Keyboard Triggering Sensitivity Bit

This read/write bit controls the triggering sensitivity of the keyboard interrupt pins. Reset clears MODEK.

1 = Keyboard interrupt requests on falling edges and low levels
0 = Keyboard interrupt requests on falling edges only

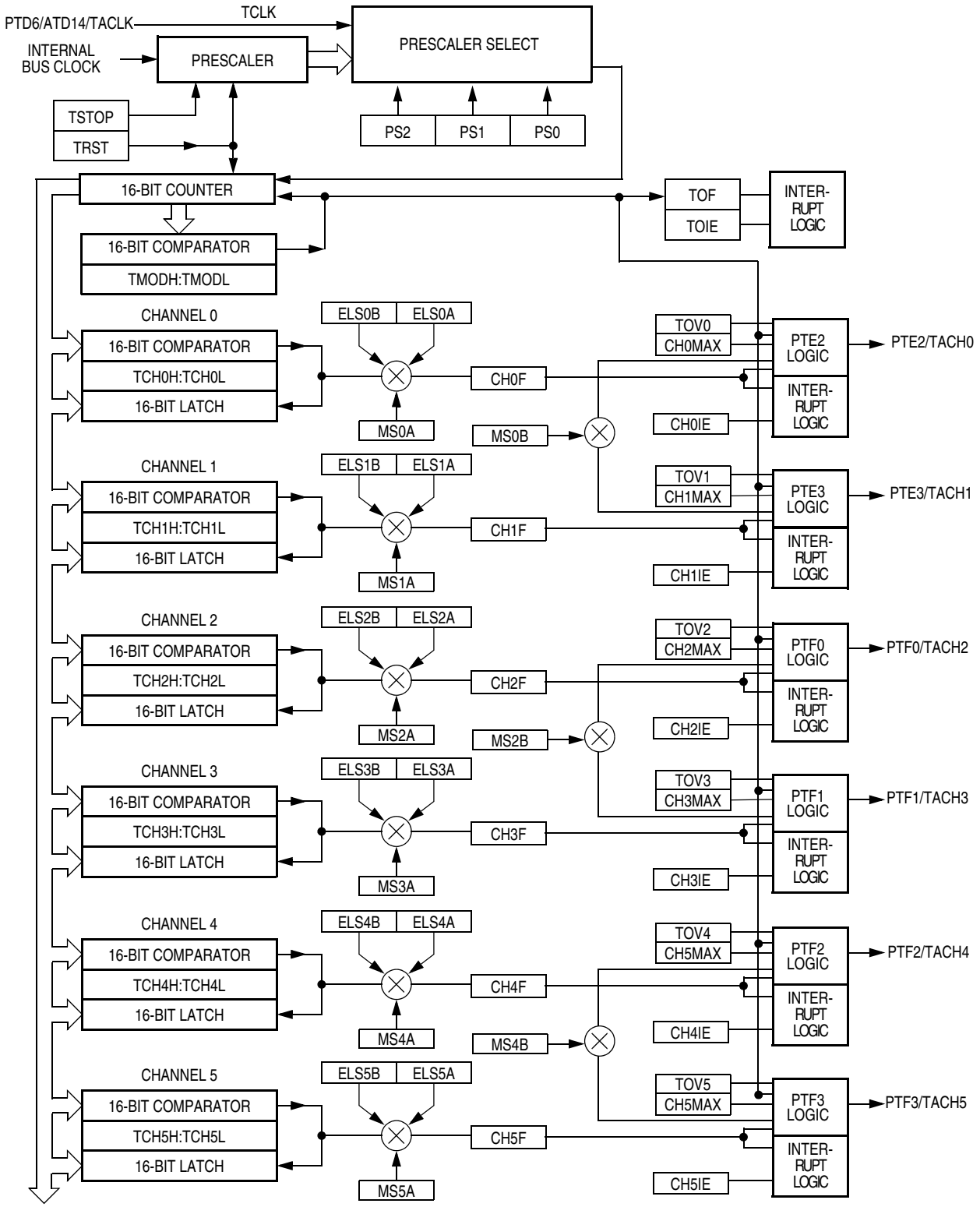


Figure 25-1. TIMA Block Diagram

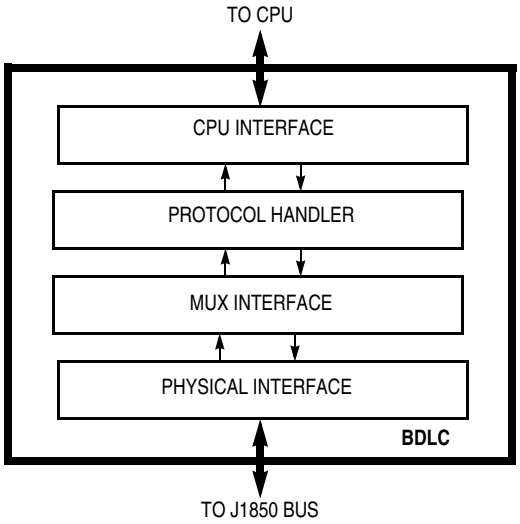


Figure 27-1. BDLC Block Diagram

Addr.	Name		Bit 7	6	5	4	3	2	1	Bit 0
$003B	BDLC Analog and Rou5ndtrip Delay Register (BARD)	Read:	ATE	RXPOL	0	0	BO3	BO2	BO1	BO0
		Write:			R	R				
$003C	BDLC Control Register 1 (BCR1)	Read:	IMSG	CLKS	R1	R0	0	0	IE	WCM
		Write:					R	R		
$003D	BDLC Control Register 2 (BCR2)	Read:	ALOOP	DLOOP	RX4XE	NBFS	TEOD	TSIFR	TMIFR1	TMIFR0
		Write:								
$003E	BDLC State Vector Register (BSVR)	Read:	0	0	I3	I2	I1	I0	0	0
		Write:	R	R	R	R	R	R	R	R
$003F	BDLC Data Register (BDR)	Read:	BD7	BD6	BD5	BD4	BD3	BD2	BD1	BD0
		Write:								

R = Reserved

Figure 27-2. BDLC I/O Register Summary

28.1.5 Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Bus Operating Frequency (4.5–5.5 V — V_{DD} Only)	f_{BUS}	—	8.4	MHz
$\overline{RST}$ Pulse Width Low	t_{RL}	1.5	—	t_{cyc}
$\overline{IRQ}$ Interrupt Pulse Width Low (Edge-Triggered)	t_{ILHI}	1.5	—	t_{cyc}
$\overline{IRQ}$ Interrupt Pulse Period	t_{ILIL}	Note 4	—	t_{cyc}
16-Bit Timer[2] Input Capture Pulse Width[3] Input Capture Period	 t_{TH}, t_{TL} t_{TLTL}	 2 Note[4]	 — —	t_{cyc}
MSCAN Wake-up Filter Pulse Width[5]	t_{WUP}	2	5	µs

1. V_{DD} = 5.0 Vdc ± 0.5v, V_{SS} = 0 Vdc, T_A = –40 °C to T_A(MAX), unless otherwise noted.
2. The 2-bit timer prescaler is the limiting factor in determining timer resolution.
3. Refer to Table 25-2. Mode, Edge, and Level Selection and supporting note.
4. The minimum period t_{TLTL} or t_{ILIL} should not be less than the number of cycles it takes to execute the capture interrupt service routine plus TBD t_{cyc}.
5. The minimum pulse width to wake up the MSCAN module is guaranteed by design but not tested.

28.1.6 ADC Characteristics

Characteristic[1]	Min	Max	Unit	Comments
Resolution	8	8	Bits	
Absolute Accuracy (V_{REFL} = 0 V, V_{DDA}/V_{DDAREF} = V_{REFH} = 5 V ± 0.5 V)	–1	+1	LSB	Includes Quantization
Conversion Range[2]	V_{REFL}	V_{REFH}	V	V_{REFL} = V_{SSA}
Power-Up Time	16	17	µs	Conversion Time Period
Input Leakage[3] (Ports B and D)	–1	1	µA	
Conversion Time	16	17	ADC Clock Cycles	Includes Sampling Time
Monotonicity	Inherent within Total Error			
Zero Input Reading	00	01	Hex	V_{IN} = V_{REFL}
Full-Scale Reading	FE	FF	Hex	V_{IN} = V_{REFH}
Sample Time[2]	5	—	ADC Clock Cycles	
Input Capacitance	—	8	pF	Not Tested
ADC Internal Clock	500 k	1.048 M	Hz	Tested Only at 1 MHz
Analog Input Voltage	V_{REFL}	V_{REFH}	V	

1. V_{DD} = 5.0 Vdc ± 0.5 V, V_{SS} = 0 Vdc, V_{DDA}/V_{DDAREF} = 5.0 Vdc ± 0.5 V, V_{SSA} = 0 Vdc, V_{REFH} = 5.0 Vdc ± 0.5 V
2. Source impedances greater than 10 kΩ adversely affect internal RC charging time during input sampling.
3. The external system error caused by input leakage current is approximately equal to the product of R source and input current.

Appendix A
MC68HC908AS60 and MC68HC908AZ60

A.1 Changes from the MC68HC908AS60 and MC68HC908AZ60 (non-A suffix devices)

A.1.1 Specification

Specifications for MC68HC908AS60A and MC68HC908AZ60A devices have been integrated, reflecting the many commonalties.

A.1.2 FLASH

A.1.2.1 FLASH Architecture

FLASH-1 and FLASH-2 are made from a new nonvolatile memory (NVM) technology. The architecture is now arranged in pages of 128 bytes and 2 rows per page. Programming is now carried out on a whole row (64 bytes) at a time. Erasing is now carried out on a whole page (128 bytes) at a time. In this new technology an erased bit now reads as a logic 1 and a programmed bit now reads as a logic 0.

A.1.2.2 FLASH Control Registers

FLASH-1 control register is moved from $FE0B to $FF88. FLASH-2 control register is moved from $FE11 to $FE08. Bits 4 to 7 in the FLASH control registers are no longer used since clock control is now achieved automatically and erasing of variable block sizes is no longer a feature. Bit 2 of the FLASH control registers no longer activates a so-called ‘margin read’ operation but instead is the bit that controls a mass (bulk) erase operation.

A.1.2.3 FLASH Programming Procedure

Programming of the FLASH is largely as before within the new architecture constraints outlined above. However, an extra dummy write operation to any address in the page is required prior to programming data into one of the two rows in the page. Margin reading of programmed data is no longer required. Nor is read / verify / re-pulse of the programming a requirement.

A.1.2.4 FLASH Programming Time

The most significant change resulting from the new FLASH technology is that the byte programming time is reduced to a maximum of 40us. This represents several orders of magnitude improvement from the previous technology.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FF70	EE2DIV Hi Nonvolatile Register (EE2DIVHNVR)	Read:	EEDIVS-ECD	R	R	R	R	EEDIV10	EEDIV9	EEDIV8
$FF71	EE2DIV Lo Nonvolatile Register (EE2DIVLNVR)	Read: Write:	EEDIV7	EEDIV6	EEDIV5	EEDIV4	EEDIV3	EEDIV2	EEDIV1	EEDIV0
$FF7A	EE2DIV Divider High Register (EE2DIVH)	Read:	EEDIVS-ECD	0	0	0	0	EEDIV10	EEDIV9	EEDIV8
		Write:								
$FF7B	EE2DIV Divider Low Register (EE2DIVL)	Read: Write:	EEDIV7	EEDIV6	EEDIV5	EEDIV4	EEDIV3	EEDIV2	EEDIV1	EEDIV0
$FE7C	EEPROM-2 Nonvolatile Register (EE2NVR)	Read: Write:	UNUSED	UNUSED	UNUSED	EEPRTCT	EEBP3	EEBP2	EEBP1	EEBP0
$FE7D	EEPROM-2 Control Register (EE2CR)	Read:	UNUSED	0	EEOFF	EERAS1	EERAS0	EELAT	AUTO	EEPGM
		Write:								
$FE7F	EEPROM-2 Array Configuration Register (EE2ACR)	Read:	UNUSED	UNUSED	UNUSED	EEPRTCT	EEBP3	EEBP2	EEBP1	EEBP0
		Write:								
$FF80	FLASH-1 Block Protect Register (FL1BPR)	Read: Write:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
$FF81	FLASH-2 Block Protect Register (FL2BPR)	Read: Write:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
$FF88	FLASH-1 Control Register (FL1CR)	Read:	0	0	0	0	HVEN	VERF	ERASE	PGM
		Write:								
$FFFF	COP Control Register (COPCTL)	Read:	LOW BYTE OF RESET VECTOR							
		Write:	WRITING TO $FFFF CLEARS COP COUNTER							

= Unimplemented R = Reserved

Figure B-3. Additional Status and Control Registers (Sheet 2 of 2)