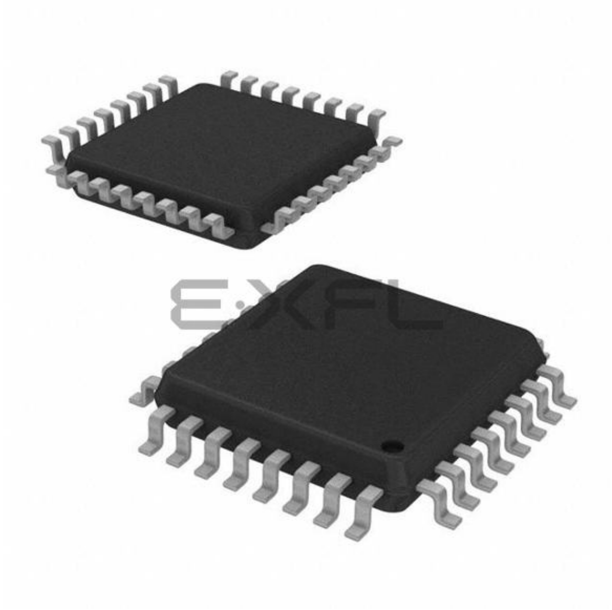
NXP USA Inc. - MC908GR60AMFJE Datasheet



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	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	21
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908gr60amfje

Chapter 3
Analog-to-Digital Converter (ADC)

Chapter 9
Keyboard Interrupt Module (KBI)

Chapter 10
Low-Power Modes

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000C	Data Direction Register E (DDRE) See page 146.	Read:	0	0	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
		Write:	▨	▨						
		Reset:	0	0	0	0	0	0	0	0
$000D	Port A Input Pullup Enable Register (PTAPUE) See page 137.	Read:	PTAPUE7	PTAPUE6	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000E	Port C Input Pullup Enable Register (PTCPUE) See page 142.	Read:	0	PTCPUE6	PTCPUE5	PTCPUE4	PTCPUE3	PTCPUE2	PTCPUE1	PTCPUE0
		Write:	▨							
		Reset:	0	0	0	0	0	0	0	0
$000F	Port D Input Pullup Enable Register (PTDPUE) See page 145.	Read:	PTDPUE7	PTDPUE6	PTDPUE5	PTDPUE4	PTDPUE3	PTDPUE2	PTDPUE1	PTDPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0010	SPI Control Register (SPCR) See page 217.	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) See page 218.	Read:	SPRF	ERRIE	OVRF	MODF	SPTE	MODFEN	SPR1	SPR0
		Write:	▨		▨	▨	▨			
		Reset:	0	0	0	0	1	0	0	0
$0012	SPI Data Register (SPDR) See page 220.	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							
$0013	ESCI Control Register 1 (SCC1) See page 166.	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0014	ESCI Control Register 2 (SCC2) See page 168.	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0015	ESCI Control Register 3 (SCC3) See page 169.	Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
		Write:	▨							
		Reset:	U	0	0	0	0	0	0	0
$0016	ESCI Status Register 1 (SCS1) See page 170.	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	1	1	0	0	0	0	0	0
$0017	ESCI Status Register 2 (SCS2) See page 173.	Read:	0	0	0	0	0	0	BKF	RPF
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented R = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 2 of 9)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0459	TIM2 Channel 3 Status and Control Register (T2SC3) See page 255.	Read:	CH3F	CH3IE	0	MS3A	ELS3B	ELS3A	TOV3	CH3MAX
		Write:	0		▨					
		Reset:	0	0	0	0	0	0	0	0
$045A	TIM2 Channel 3 Register High (T2CH3H) See page 258.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$045B	TIM2 Channel 3 Register Low (T2CH3L) See page 258.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$045C	TIM2 Channel 4 Status and Control Register (T2SC4) See page 255.	Read:	CH4F	CH4IE	MS4B	MS4A	ELS4B	ELS4A	TOV4	CH4MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$045D	TIM2 Channel 4 Register High (T2CH4H) See page 258.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$045E	TIM2 Channel 4 Register Low (T2CH4L) See page 258.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$045F	TIM2 Channel 5 Status and Control Register (T2SC5) See page 255.	Read:	CH5F	CH5IE	0	MS5A	ELS5B	ELS5A	TOV 5	CH5MAX
		Write:	0		▨					
		Reset:	0	0	0	0	0	0	0	0
$0460	TIM2 Channel 5 Register High (T2CH5H) See page 258.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0461	TIM2 Channel 5 Register Low (T2CH5L) See page 258.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$FE00	Break Status Register (BSR) See page 199.	Read:	R	R	R	R	R	R	SBSW	R
		Write:							NOTE 1	
		Reset:	0	0	0	0	0	0	0	0

1. Writing a 0 clears SBSW.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE01	SIM Reset Status Register (SRSR) See page 199.	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		POR:	1	0	0	0	0	0	0	0
$FE02	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								
		Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented R = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 7 of 9)

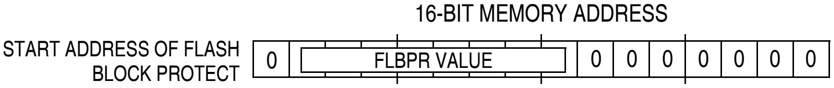


Figure 2-9. FLASH-2 Block Protect Start Address

Table 2-3. FLASH-2 Protected Ranges

FL2BPR[7:0]	Protected Range
$FF	No Protection
$FE	$7F00–$7FFF
$FD ↓ $0B	$7E80–$7FFF ↓ $0580–$7FFF
$0A	$0500–$7FFF
$09	$0480–$7FFF
$08 ↓ $04	$0462–$7FFF ↓ $0462–$7FFF
$03	$0462–$7FFF
$02	$0462–$7FFF
$01	$0462–$7FFF
$00	$0462–$7FFF

Decreasing the value in FL2BPR by one increases the protected range by one page (128 bytes). However, programming the block protect register with $FE protects a range twice that size, 256 bytes, in the corresponding array. $FE means that locations $7F00–$7FFF are protected in FLASH-2.

The FLASH memory does not exist at some locations. The block protection range configuration is unaffected if FLASH memory does not exist in that range. Refer to Figure 2-1 and make sure that the desired locations are protected.

2.7.3 FLASH-2 Block Protection

Due to the ability of the on-board charge pump to erase and program the FLASH memory in the target application, provision is made for protecting blocks of memory from unintentional erase or program operations due to system malfunction. This protection is done by using the FLASH-2 block protection register (FL2BPR). FL2BPR determines the range of the FLASH-2 memory which is to be protected. The range of the protected area starts from a location defined by FL2BPR and ends at the bottom of the FLASH-2 memory ($7FFF). When the memory is protected, the HVEN bit can not be set in either ERASE or PROGRAM operations.

NOTE

In performing a program or erase operation, the FLASH-2 block protect register must be read after setting the PGM or ERASE bit and before asserting the HVEN bit.

When the FLASH-2 block protect register is programmed with all 0's, the entire memory is protected from being programmed and erased. When all the bits are erased (all 1's), the entire memory is accessible for program and erase.

2.7.5 FLASH-2 Page Erase Operation

Use this step-by-step procedure to erase a page (128 bytes) of FLASH-2 memory:

1. Set the ERASE bit and clear the MASS bit in the FLASH-2 control register (FL2CR).
2. Read the FLASH-2 block protect register (FL2BPR).
3. Write any data to any FLASH-2 address within the address range of the page (128 byte block) to be erased.
4. Wait for time, t_{NVS} (minimum 10 μs).
5. Set the HVEN bit.
6. Wait for time, t_{ERASE} (minimum 1 ms or 4 ms).
7. Clear the ERASE bit.
8. Wait for time, t_{NVH} (minimum 5 μs).
9. Clear the HVEN bit.
10. Wait for a time, t_{RCV}, (typically 1 μs) after which the memory can be accessed in normal read mode.

NOTE

A. *Programming and erasing of FLASH locations can not be performed by code being executed from the same FLASH array.*

B. *While these operations must be performed in the order shown, other unrelated operations may occur between the steps. However, care must be taken to ensure that these operations do not access any address within the FLASH array memory space such as the COP control register (COPCTL) at $FFFF.*

C. *It is highly recommended that interrupts be disabled during program/erase operations.*

In applications that require more than 1000 program/erase cycles, use the 4 ms page erase specification to get improved long-term reliability. Any application can use this 4 ms page erase specification. However, in applications where a FLASH location will be erased and reprogrammed less than 1000 times, and speed is important, use the 1 ms page erase specification to get a shorter cycle time.

The COP counter is a free-running 6-bit counter preceded by the 12-bit SIM counter. If not cleared by software, the COP counter overflows and generates an asynchronous reset after 262,128 or 8176 CGMXCLK cycles, depending on the state of the COP rate select bit, COPRS, in the configuration register. With a 262,128 CGMXCLK cycle overflow option, a 4.9152-MHz crystal gives a COP timeout period of 53.3 ms. Writing any value to location $FFFF before an overflow occurs prevents a COP reset by clearing the COP counter and stages 12–5 of the SIM counter.

NOTE

Service the COP immediately after reset and before entering or after exiting stop mode to guarantee the maximum time before the first COP counter overflow.

A COP reset pulls the $\overline{\mathrm{RST}}$ pin low for 32 CGMXCLK cycles and sets the COP bit in the reset status register (RSR).

In monitor mode, the COP is disabled if the $\overline{\mathrm{RST}}$ pin or the $\overline{\mathrm{IRQ}}$ is held at V_{TST}. During the break state, V_{TST} on the $\overline{\mathrm{RST}}$ pin disables the COP.

NOTE

Place COP clearing instructions in the main program and not in an interrupt subroutine. Such an interrupt subroutine could keep the COP from generating a reset even while the main program is not working properly.

6.3 I/O Signals

The following paragraphs describe the signals shown in Figure 6-1.

6.3.1 CGMXCLK

CGMXCLK is the crystal oscillator output signal. CGMXCLK frequency is equal to the crystal frequency.

6.3.2 STOP Instruction

The STOP instruction clears the SIM counter.

6.3.3 COPCTL Write

Writing any value to the COP control register (COPCTL) clears the COP counter and clears stages 12–5 of the SIM counter. Reading the COP control register returns the low byte of the reset vector. See 6.4 COP Control Register.

6.3.4 Power-On Reset

The power-on reset (POR) circuit clears the SIM counter 4096 CGMXCLK cycles after power-up.

6.3.5 Internal Reset

An internal reset clears the SIM counter and the COP counter.

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

A falling edge 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 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 latches 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 a high level — As long as the $\overline{\text{IRQ}}$ pin is low, IRQ remains active.

The vector fetch or software clear and the return of the $\overline{\text{IRQ}}$ pin to a high level may occur in any order. The interrupt request remains pending as long as the $\overline{\text{IRQ}}$ pin is low. 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 19 Development Support.

To allow software to clear the IRQ latch during a break interrupt, write a 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 0 to the BCFE bit. With BCFE at 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.

Chapter 9
Keyboard Interrupt Module (KBI)

9.1 Introduction

The keyboard interrupt module (KBI) provides eight independently maskable external interrupts which are accessible via PTA0–PTA7. When a port pin is enabled for keyboard interrupt function, an internal pullup/pulldown device is also enabled on the pin.

9.2 Features

Features include:

- Eight keyboard interrupt pins with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Hysteresis buffers
- Programmable edge-only or edge- and level- interrupt sensitivity
- Edge detect programmable for rising or falling edges
- Level detect programmable for high or low levels
- Exit from low-power modes
- Pullup/pulldown device automatically configured based on polarity of edge/level selection

9.3 Functional Description

Writing to the KBIE7–KBIE0 bits in the keyboard interrupt enable register independently enables or disables each port A pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin also enables its internal pullup/pulldown device. On falling edge or low level selection a pullup device is configured. On rising edge or high level selection a pulldown device is configured.

- A falling edge is detected when an enabled keyboard input signal is seen as a 1 (the deasserted level) during one bus cycle and then a 0 (the asserted level) during the next cycle.
- A rising edge is detected when the input signal is seen as a 0 during one bus cycle and then a 1 during the next cycle.

A keyboard interrupt is latched when one or more keyboard pins are asserted. The MODEK bit in the keyboard status and control register controls the triggering mode of the keyboard interrupt.

LVISTOP, LVIPWRD, LVI5OR3, and LVIRSTD are in the configuration register (CONFIG1). See Figure 5-2. Configuration Register 1 (CONFIG1) for details of the LVI's configuration bits. Once an LVI reset occurs, the MCU remains in reset until V_{DD} rises above a voltage, V_{TRIPR}, which causes the MCU to exit reset. See 14.3.2.5 Low-Voltage Inhibit (LVI) Reset for details of the interaction between the SIM and the LVI. The output of the comparator controls the state of the LVIOUT flag in the LVI status register (LVISR).

An LVI reset also drives the $\overline{\text{RST}}$ pin low to provide low-voltage protection to external peripheral devices.

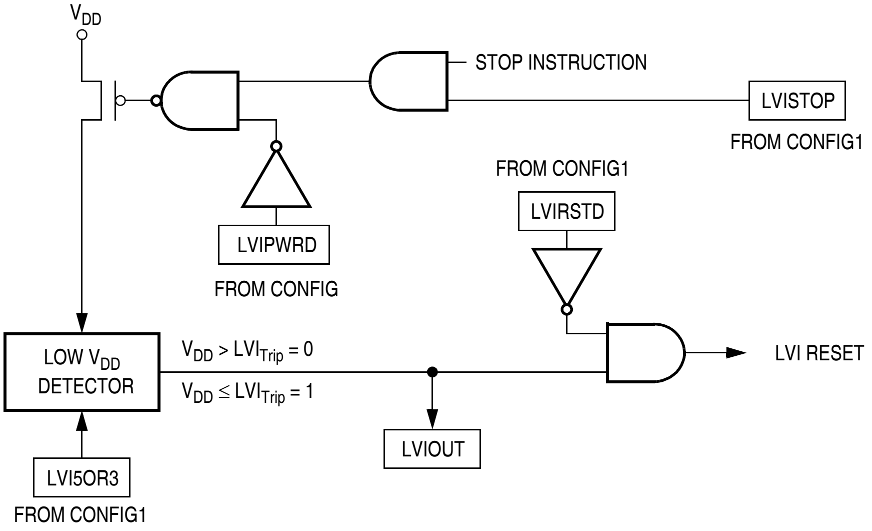


Figure 11-1. LVI Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE0C	LVI Status Register (LVISR) See page 129.	Read:	LVIOUT	0	0	0	0	0	0	0
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 11-2. LVI I/O Register Summary

11.3.1 Polled LVI Operation

In applications that can operate at V_{DD} levels below the V_{TRIPF} level, software can monitor V_{DD} by polling the LVIOUT bit. In the configuration register, the LVIPWRD bit must be 0 to enable the LVI module, and the LVIRSTD bit must be 1 to disable LVI resets.

11.3.2 Forced Reset Operation

In applications that require V_{DD} to remain above the V_{TRIPF} level, enabling LVI resets allows the LVI module to reset the MCU when V_{DD} falls below the V_{TRIPF} level. In the configuration register, the LVIPWRD and LVIRSTD bits must be cleared to enable the LVI module and to enable LVI resets.

12.4 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.

12.4.1 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							
Alternate 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 during the CPU read cycle. Those ports not selected as analog input channels are considered digital I/O ports.

12.4.2 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)

DDRB7–DDRB0 — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB7–DDRB0, configuring all port B pins as inputs.

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

T2CH5–T2CH2 — Timer 2 Channel I/O Bits

The PTF7/T2CH5–PTF4/T2CH2 pins are the TIM2 input capture/output compare pins. The edge/level select bits, ELSxB:ELSxA, determine whether the PTF7/T2CH5–PTF4/T2CH2 pins are timer channel I/O pins or general-purpose I/O pins. See Chapter 17 Timer Interface Module (TIM1) and Chapter 18 Timer Interface Module (TIM2).

12.8.2 Data Direction Register F

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

Address: $0444

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRF7	DDRF6	DDRF5	DDRF4	DDRF3	DDRF2	DDRF1	DDRF0
Reset:	0	0	0	0	0	0	0	0

Figure 12-21. Data Direction Register F (DDRF)

DDRF7–DDRF0 — Data Direction Register F Bits

These read/write bits control port F data direction. Reset clears DDRF7–DDRF0, configuring all port F pins as inputs.

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

NOTE

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

Figure 12-22 shows the port F I/O logic.

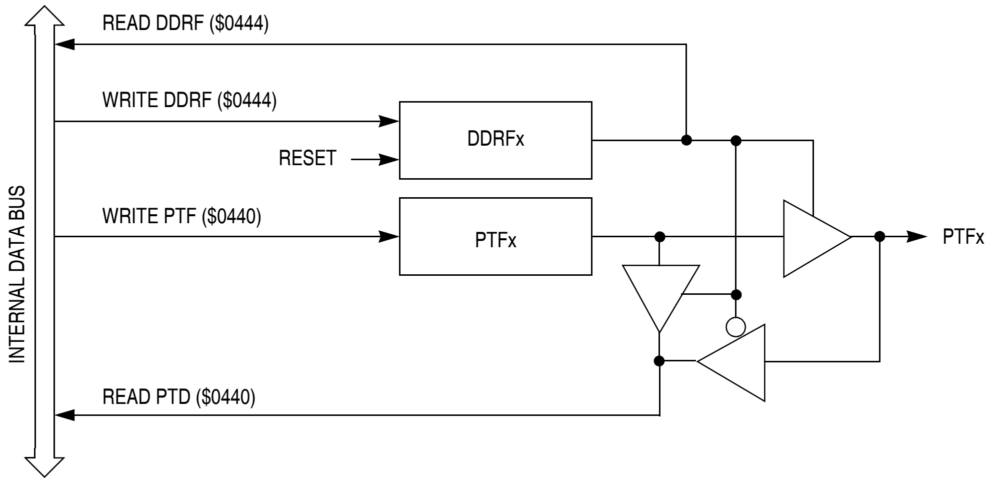


Figure 12-22. Port F I/O Circuit

When bit DDRFx is a 1, reading address $0440 reads the PTFx data latch. When bit DDRFx is a 0, reading address $0440 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 12-7 summarizes the operation of the port F pins.

To initiate an ESCI transmission:

1. Enable the ESCI by writing a 1 to the enable ESCI bit (ENSCI) in ESCI control register 1 (SCC1).
2. Enable the transmitter by writing a 1 to the transmitter enable bit (TE) in ESCI control register 2 (SCC2).
3. Clear the ESCI transmitter empty bit (SCTE) by first reading ESCI status register 1 (SCS1) and then writing to the SCDR. For 9-bit data, also write the T8 bit in SCC3.
4. Repeat step 3 for each subsequent transmission.

At the start of a transmission, transmitter control logic automatically loads the transmit shift register with a preamble of 1s. After the preamble shifts out, control logic transfers the SCDR data into the transmit shift register. A 0 start bit automatically goes into the least significant bit (LSB) position of the transmit shift register. A 1 stop bit goes into the most significant bit (MSB) position.

The ESCI transmitter empty bit, SCTE, in SCS1 becomes set when the SCDR transfers a byte to the transmit shift register. The SCTE bit indicates that the SCDR can accept new data from the internal data bus. If the ESCI transmit interrupt enable bit, SCTIE, in SCC2 is also set, the SCTE bit generates a transmitter CPU interrupt request.

When the transmit shift register is not transmitting a character, the TxD pin goes to the idle condition, high. If at any time software clears the ENSCI bit in ESCI control register 1 (SCC1), the transmitter and receiver relinquish control of the port E pins.

13.4.2.3 Break Characters

Writing a 1 to the send break bit, SBK, in SCC2 loads the transmit shift register with a break character. For TXINV = 0 (output not inverted), a transmitted break character contains all 0s and has no start, stop, or parity bit. Break character length depends on the M bit in SCC1 and the LINR bits in SCBR. As long as SBK is at 1, transmitter logic continuously loads break characters into the transmit shift register. After software clears the SBK bit, the shift register finishes transmitting the last break character and then transmits at least one 1. The automatic 1 at the end of a break character guarantees the recognition of the start bit of the next character.

When LINR is cleared in SCBR, the ESCI recognizes a break character when a start bit is followed by eight or nine 0 data bits and a 0 where the stop bit should be, resulting in a total of 10 or 11 consecutive 0 data bits. When LINR is set in SCBR, the ESCI recognizes a break character when a start bit is followed by 9 or 10 0 data bits and a 0 where the stop bit should be, resulting in a total of 11 or 12 consecutive 0 data bits.

Receiving a break character has these effects on ESCI registers:

- Sets the framing error bit (FE) in SCS1
- Sets the ESCI receiver full bit (SCRF) in SCS1
- Clears the ESCI data register (SCDR)
- Clears the R8 bit in SCC3
- Sets the break flag bit (BKF) in SCS2
- May set the overrun (OR), noise flag (NF), parity error (PE), or reception in progress flag (RPF) bits

14.3 Reset and System Initialization

The MCU has these reset sources:

- Power-on reset module (POR)
- External reset pin ($\overline{\text{RST}}$)
- Computer operating properly module (COP)
- Low-voltage inhibit module (LVI)
- Illegal opcode
- Illegal address
- Forced monitor mode entry reset (MODRST)

All of these resets produce the vector $FFFE:$FFFF ($FEFE:$FEFF in monitor mode) and assert the internal reset signal (IRST). IRST causes all registers to be returned to their default values and all modules to be returned to their reset states.

An internal reset clears the SIM counter (see 14.4 SIM Counter), but an external reset does not. Each of the resets sets a corresponding bit in the SIM reset status register (SRSR). See 14.7 SIM Registers.

A reset immediately stops the operation of the instruction being executed. Reset initializes certain control and status bits. Reset selects CGMXCLK divided by four as the bus clock.

14.3.1 External Pin Reset

The $\overline{\text{RST}}$ pin circuit includes an internal pullup device. Pulling the asynchronous $\overline{\text{RST}}$ pin low halts all processing. The PIN bit of the SIM reset status register (SRSR) is set as long as $\overline{\text{RST}}$ is held low for at least the minimum t_{RL} time and no other reset sources are present. Figure 14-4 shows the relative timing.

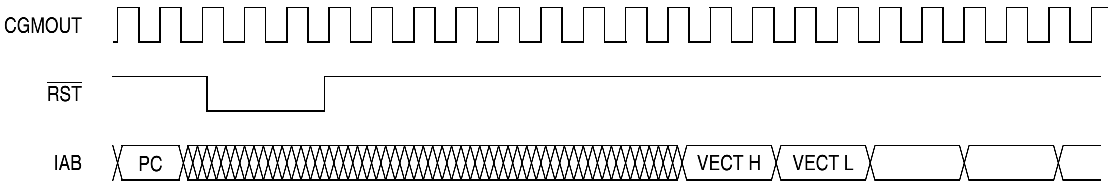


Figure 14-4. External Reset Timing

14.3.2 Active Resets from Internal Sources

All internal reset sources actively pull the $\overline{\text{RST}}$ pin low for 32 CGMXCLK cycles to allow resetting of external peripherals. The internal reset continues to be asserted for an additional 32 cycles at which point the reset vector will be fetched. See Figure 14-5. An internal reset can be caused by an illegal address, illegal opcode, COP timeout, LVI, or POR. See Figure 14-6.

NOTE

For LVI or POR resets, the SIM cycles through 4096 CGMXCLK cycles during which the SIM forces the $\overline{\text{RST}}$ pin low. The internal reset signal then follows the sequence from the falling edge of $\overline{\text{RST}}$ shown in Figure 14-5.

The COP reset is asynchronous to the bus clock.

The active reset feature allows the part to issue a reset to peripherals and other chips within a system built around the MCU.

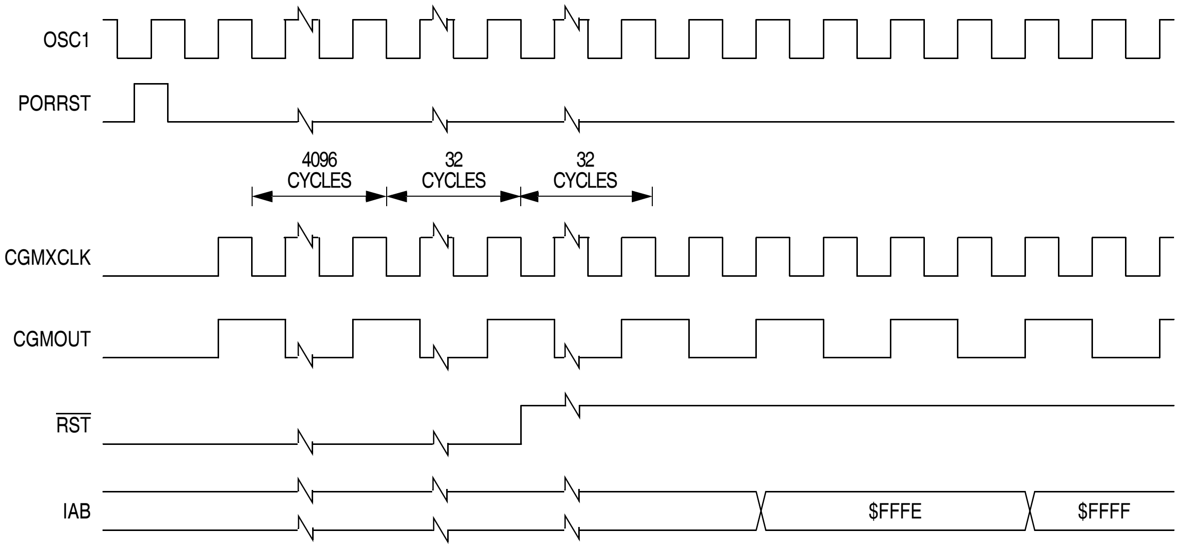


Figure 14-7. POR Recovery

14.3.2.3 Illegal Opcode Reset

The SIM decodes signals from the CPU to detect illegal instructions. An illegal instruction sets the ILOP bit in the SIM reset status register (SRSR) and causes a reset.

If the stop enable bit, STOP, in the CONFIG1 register is 0, the SIM treats the STOP instruction as an illegal opcode and causes an illegal opcode reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

14.3.2.4 Illegal Address Reset

An opcode fetch from an unmapped address generates an illegal address reset. The SIM verifies that the CPU is fetching an opcode prior to asserting the ILAD bit in the SIM reset status register (SRSR) and resetting the MCU. A data fetch from an unmapped address does not generate a reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

14.3.2.5 Low-Voltage Inhibit (LVI) Reset

The low-voltage inhibit module (LVI) asserts its output to the SIM when the V_{DD} voltage falls to the V_{TRIPF} voltage. The LVI bit in the SIM reset status register (SRSR) is set, and the external reset pin ($\overline{\text{RST}}$) is asserted if the LVIPWRD and LVIRSTD bits in the CONFIG1 register are 0. The $\overline{\text{RST}}$ pin will be held low while the SIM counter counts out 4096 + 32 CGMXCLK cycles after V_{DD} rises above V_{TRIPR}. Thirty-two CGMXCLK cycles later, the CPU is released from reset to allow the reset vector sequence to occur. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

14.3.2.6 Monitor Mode Entry Module Reset (MODRST)

The monitor mode entry module reset (MODRST) asserts its output to the SIM when monitor mode is entered in the condition where the reset vectors are erased ($FF) (see 19.3.1.1 Normal Monitor Mode). When MODRST gets asserted, an internal reset occurs. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

Chapter 16
Timebase Module (TBM)

16.1 Introduction

This section describes the timebase module (TBM). The TBM will generate periodic interrupts at user selectable rates using a counter clocked by the external clock source. This TBM version uses 15 divider stages, eight of which are user selectable. A configuration option bit to select an additional 128 divide of the external clock source can be selected. See Chapter 5 Configuration Register (CONFIG)

16.2 Features

Features of the TBM module include:

- External clock or an additional divide-by-128 selected by configuration option bit as clock source
- Software configurable periodic interrupts with divide-by: 8, 16, 32, 64, 128, 2048, 8192, and 32768 taps of the selected clock source
- Configurable for operation during stop mode to allow periodic wakeup from stop

16.3 Functional Description

This module can generate a periodic interrupt by dividing the clock source supplied from the clock generator module, CGMXCLK.

The counter is initialized to all 0s when TBON bit is cleared. The counter, shown in Figure 16-1, starts counting when the TBON bit is set. When the counter overflows at the tap selected by TBR2–TBR0, the TBIF bit gets set. If the TBIE bit is set, an interrupt request is sent to the CPU. The TBIF flag is cleared by writing a 1 to the TACK bit. The first time the TBIF flag is set after enabling the timebase module, the interrupt is generated at approximately half of the overflow period. Subsequent events occur at the exact period.

The timebase module may remain active after execution of the STOP instruction if the crystal oscillator has been enabled to operate during stop mode through the OSCENINSTOP bit in the configuration register. The timebase module can be used in this mode to generate a periodic wakeup from stop mode.

16.4 Interrupts

The timebase module can periodically interrupt the CPU with a rate defined by the selected TBMCLK and the select bits TBR2–TBR0. When the timebase counter chain rolls over, the TBIF flag is set. If the TBIE bit is set, enabling the timebase interrupt, the counter chain overflow will generate a CPU interrupt request.

NOTE

Interrupts must be acknowledged by writing a 1 to the TACK bit.

17.3.3 Output Compare

With the output compare function, the TIM1 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 TIM1 can set, clear, or toggle the channel pin. Output compares can generate TIM1 CPU interrupt requests.

17.3.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in 17.3.3 Output Compare. The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIM1 channel registers.

An unsynchronized write to the TIM1 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 TIM1 overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIM1 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 TIM1 overflow interrupts and write the new value in the TIM1 overflow interrupt routine. The TIM1 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.

17.3.3.2 Buffered Output Compare

Channels 0 and 1 can be linked to form a buffered output compare channel whose output appears on the T1CH0 pin. The TIM1 channel registers of the linked pair alternately control the output.

Setting the MS0B bit in TIM1 channel 0 status and control register (TSC0) links channel 0 and channel 1. The output compare value in the TIM1 channel 0 registers initially controls the output on the T1CH0 pin. Writing to the TIM1 channel 1 registers enables the TIM1 channel 1 registers to synchronously control the output after the TIM1 overflows. At each subsequent overflow, the TIM1 channel registers (0 or 1) that control the output are the ones written to last. T1SC0 controls and monitors the buffered output compare function, and TIM1 channel 1 status and control register (T1SC1) is unused. While the MS0B bit is set, the channel 1 pin, T1CH1, 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.

18.8.4 TIM2 Channel Status and Control Registers

Each of the TIM2 channel status and control registers:

- Flags input captures and output compares
- Enables input capture and output compare interrupts
- Selects input capture, output compare, or PWM operation
- Selects high, low, or toggling output on output compare
- Selects rising edge, falling edge, or any edge as the active input capture trigger
- Selects output toggling on TIM2 overflow
- Selects 0% and 100% PWM duty cycle
- Selects buffered or unbuffered output compare/PWM operation

Address: $0030 T2SC0

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Address: $0033 T2SC1

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
Write:	0		▨					
Reset:	0	0	0	0	0	0	0	0

Address: $0456 T2SC2

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	TOV2	CH2MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Address: $0459 T2SC3

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH3F	CH3IE	0	MS3A	ELS3B	ELS3A	TOV3	CH3MAX
Write:	0		▨					
Reset:	0	0	0	0	0	0	0	0

Address: $045C T2SC4

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH4F	CH4IE	MS4B	MS4A	ELS4B	ELS4A	TOV4	CH4MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Address: $045F T2SC5

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH5F	CH5IE	0	MS5A	ELS5B	ELS5A	TOV5	CH5MAX
Write:	0		▨					
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 18-8. TIM2 Channel Status and Control Registers (T2SC0:T2SC5)

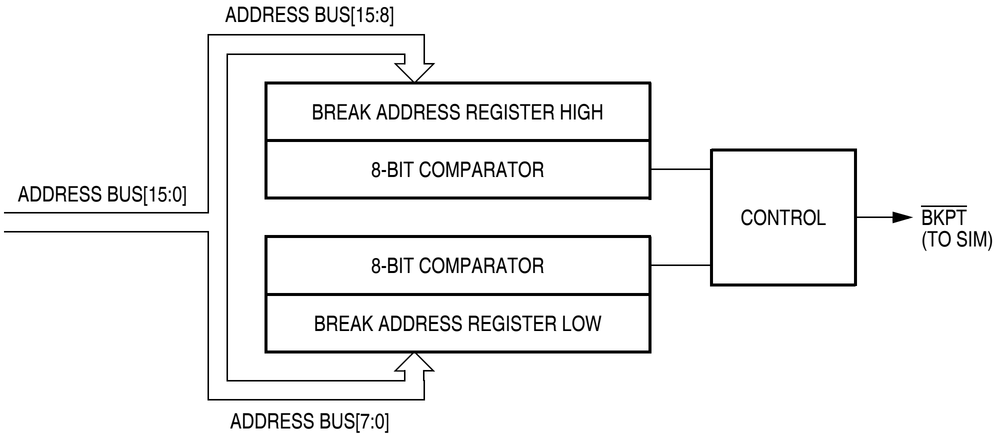


Figure 19-2. Break Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE00	Break Status Register (BSR) See page 266.	Read:	R	R	R	R	R	R	SBSW	R
		Write:							Note[1]	
		Reset:							0	
$FE03	Break Flag Control Register (BFCR) See page 266.	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE09	Break Address High Register (BRKH) See page 265.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0A	Break Address Low Register (BRKL) See page 265.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$FE0B	Break Status and Control Register (BRKSCR) See page 265.	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:			▨	▨	▨	▨	▨	▨
		Reset:	0	0	0	0	0	0	0	0

[1] Writing a 0 clears SBSW.

▨ = Unimplemented   R = Reserved

Figure 19-3. Break I/O Register Summary

Appendix A
MC68HC908GR48A

A.1 Introduction

The MC68HC908GR48A is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

The information contained in this document pertains to the MC68HC908GR48A with the exceptions shown in this appendix.

A.2 Block Diagram

See Figure A-1.

A.3 Memory

The MC68HC908GR48A can address 48 Kbytes of memory space. The memory map, shown in Figure A-2, includes:

- 48 Kbytes of user FLASH memory
- 1536 bytes of random-access memory (RAM)
- 52 bytes of user-defined vectors