



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

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

Details

Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	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 ~ 85°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/mc908gr16acfaer

Email: info@E-XFL.COM

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



Input/Output (I/O) Section

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0				
\$0020	Timer 2 Channel 0 Status and	Read:	CH0F	CHOIE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX				
\$0030	See page 227.	Poost:	0	0	0	0	0	0	0	0				
		Road	0	0	0	0	0	0	0	0				
\$0031 Reg	Timer 2 Channel 0 Register High (T2CH0H)	Write:	Bit 15	14	13	12	11	10	9	Bit 8				
	See page 227.	Reset:		r	1	Indetermina	te after reset	1	1	1				
\$0032	Timer 2 Channel 0 Register Low (T2CH0L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0				
	See page 230.	Reset:		Indeterminate after reset										
	Timer 2 Channel 1 Status and	Read:	CH1F	CHILE	0	MS1A	EL S1B	EL S1A	TOV1	СН1МАХ				
\$0033	Control Register (T2SC1)	Write:	0	OTTIE		MOTA	ELGID	ELSTA	1001	CHIMAX				
	See page 225.	Reset:	0	0	0	0	0	0	0	0				
\$0034	Timer 2 Channel 1 Register High (T2CH1H)	Read: Write:	Bit 15	14	13	12	11	10	9	Bit 8				
See page 230.		Reset:			1	Indetermina	te after reset	1	1	1				
\$0035	Timer 2 Channel 1 Register Low (T2CH1L)	Read: Write:	Bit 7	6	5	4	3	2	1	Bit 0				
	See page 230.	Reset:				Indetermina	te after reset							
\$0036	PLL Control Register (PCTL)	Read: Write:	PLLIE	PLLF	PLLON	BCS	R	R	VPR1	VPR0				
	See page 67.	Reset:	0	0	1	0	0	0	0	0				
\$0037	PLL Bandwidth Control Register (PBWC)	Read: Write:	AUTO	LOCK	ACQ	0	0	0	0	R				
	See page 68.	Reset:	0	0	0	0	0	0	0	0				
\$0038	PLL Multiplier Select High Begister (PMSH)	Read: Write:	0	0	0	0	MUL11	MUL10	MUL9	MUL8				
φυυυυ	See page 69.	Reset	0	0	0	0	0	0	0	0				
\$0039	PLL Multiplier Select Low Register (PMSL)	Read: Write:	MUL7	MUL6	MUL5	MUL4	MUL3	MUL2	MUL1	MULO				
+	See page 70.	Reset:	0	1	0	0	0	0	0	0				
\$003A	PLL VCO Select Range Register (PMRS)	Read: Write:	VRS7	VRS6	VRS5	VRS4	VRS3	VRS2	VRS1	VRS0				
<i>vvvvvvvvvvvvv</i>	See page 70.	Reset:	0	1	0	0	0	0	0	0				
		Read:	0	0	0	0	-	-	-	-				
\$003B	Reserved	Write:					R	R	R	R				
		Reset:	0	0	0	0	0	0	0	1				
				= Unimplem	nented	R = Reserve	ed	U = Unaffect	ted					



MC68HC908GR16A Data Sheet, Rev. 1.0



Clock Generator Module (CGM)

4.3.4 Acquisition and Tracking Modes

The PLL filter is manually or automatically configurable into one of two operating modes:

- Acquisition mode In acquisition mode, the filter can make large frequency corrections to the VCO. This mode is used at PLL start up or when the PLL has suffered a severe noise hit and the VCO frequency is far off the desired frequency. When in acquisition mode, the ACQ bit is clear in the PLL bandwidth control register. (See 4.5.2 PLL Bandwidth Control Register.)
- Tracking mode In tracking mode, the filter makes only small corrections to the frequency of the VCO. PLL jitter is much lower in tracking mode, but the response to noise is also slower. The PLL enters tracking mode when the VCO frequency is nearly correct, such as when the PLL is selected as the base clock source. (See 4.3.8 Base Clock Selector Circuit.) The PLL is automatically in tracking mode when not in acquisition mode or when the ACQ bit is set.

4.3.5 Manual and Automatic PLL Bandwidth Modes

The PLL can change the bandwidth or operational mode of the loop filter manually or automatically. Automatic mode is recommended for most users.

In automatic bandwidth control mode (AUTO = 1), the lock detector automatically switches between acquisition and tracking modes. Automatic bandwidth control mode also is used to determine when the VCO clock, CGMVCLK, is safe to use as the source for the base clock, CGMOUT. (See 4.5.2 PLL Bandwidth Control Register.) If PLL interrupts are enabled, the software can wait for a PLL interrupt request and then check the LOCK bit. If interrupts are disabled, software can poll the LOCK bit continuously (for example, during PLL start up) or at periodic intervals. In either case, when the LOCK bit is set, the VCO clock is safe to use as the source for the base clock. (See 4.3.8 Base Clock Selector Circuit.) If the VCO is selected as the source for the base clock and the LOCK bit is clear, the PLL has suffered a severe noise hit and the software must take appropriate action, depending on the application. (See 4.6 Interrupts for information and precautions on using interrupts.)

The following conditions apply when the PLL is in automatic bandwidth control mode:

- The ACQ bit (See 4.5.2 PLL Bandwidth Control Register.) is a read-only indicator of the mode of the filter. (See 4.3.4 Acquisition and Tracking Modes.)
- The ACQ bit is set when the VCO frequency is within a certain tolerance and is cleared when the VCO frequency is out of a certain tolerance. (See 4.8 Acquisition/Lock Time Specifications for more information.)
- The LOCK bit is a read-only indicator of the locked state of the PLL.
- The LOCK bit is set when the VCO frequency is within a certain tolerance and is cleared when the VCO frequency is out of a certain tolerance. (See 4.8 Acquisition/Lock Time Specifications for more information.)
- CPU interrupts can occur if enabled (PLLIE = 1) when the PLL's lock condition changes, toggling the LOCK bit. (See 4.5.1 PLL Control Register.)

The PLL also may operate in manual mode (AUTO = 0). Manual mode is used by systems that do not require an indicator of the lock condition for proper operation. Such systems typically operate well below f_{BUSMAX} .



Chapter 5 **Configuration Register (CONFIG)**

5.1 Introduction

This section describes the configuration registers, CONFIG1 and CONFIG2. The configuration registers enable or disable these options:

- Stop mode recovery time (32 CGMXCLK cycles or 4096 CGMXCLK cycles)
- COP timeout period (262,128 or 8176 COPCLK cycles)
- STOP instruction
- Computer operating properly module (COP) •
- Low-voltage inhibit (LVI) module control and voltage trip point selection
- Enable/disable the oscillator (OSC) during stop mode
- Enable/disable an extra divide by 128 prescaler in timebase module

5.2 Functional Description

The configuration registers are used in the initialization of various options. The configuration registers can be written once after each reset. All of the configuration register bits are cleared during reset. Since the various options affect the operation of the microcontroller unit (MCU), it is recommended that these registers be written immediately after reset. The configuration registers are located at \$001E and \$001F and may be read at anytime.

NOTE

On a FLASH device, the options except LVI5OR3 are one-time writable by the user after each reset. The LVI5OR3 bit is one-time writable by the user only after each POR (power-on reset). The CONFIG registers are not in the FLASH memory but are special registers containing one-time writable latches after each reset. Upon a reset, the CONFIG registers default to predetermined settings as shown in Figure 5-1 and Figure 5-2.



Figure 5-1. Configuration Register 2 (CONFIG2)

MC68HC908GR16A Data Sheet, Rev. 1.0



Central Processor Unit (CPU)

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

Table 7-1.	Instruction	Set	Summarv	(Sheet 5	of 6)
				(0	



Chapter 10 Low-Power Modes

10.1 Introduction

The microcontroller (MCU) may enter two low-power modes: wait mode and stop mode. They are common to all HC08 MCUs and are entered through instruction execution. This section describes how each module acts in the low-power modes.

10.1.1 Wait Mode

The WAIT instruction puts the MCU in a low-power standby mode in which the central processor unit (CPU) clock is disabled but the bus clock continues to run. Power consumption can be further reduced by disabling the low-voltage inhibit (LVI) module through bits in the CONFIG1 register. See Chapter 5 Configuration Register (CONFIG).

10.1.2 Stop Mode

Stop mode is entered when a STOP instruction is executed. The CPU clock is disabled and the bus clock is disabled if the OSCENINSTOP bit in the CONFIG2 register is a 0. See Chapter 5 Configuration Register (CONFIG).

10.2 Analog-to-Digital Converter (ADC)

10.2.1 Wait Mode

The analog-to-digital converter (ADC) continues normal operation during wait mode. Any enabled CPU interrupt request from the ADC can bring the MCU out of wait mode. If the ADC is not required to bring the MCU out of wait mode, power down the ADC by setting ADCH4–ADCH0 bits in the ADC status and control register before executing the WAIT instruction.

10.2.2 Stop Mode

The ADC module is inactive after the execution of a STOP instruction. Any pending conversion is aborted. ADC conversions resume when the MCU exits stop mode after an external interrupt. Allow one conversion cycle to stabilize the analog circuitry.



10.3 Break Module (BRK)

10.3.1 Wait Mode

The break (BRK) module is active in wait mode. In the break routine, the user can subtract one from the return address on the stack if the SBSW bit in the break status register is set.

10.3.2 Stop Mode

The break module is inactive in stop mode. The STOP instruction does not affect break module register states.

10.4 Central Processor Unit (CPU)

10.4.1 Wait Mode

The WAIT instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling interrupts. After exit from wait mode by interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

10.4.2 Stop Mode

The STOP instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling external interrupts. After exit from stop mode by external interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

After exiting stop mode, the CPU clock begins running after the oscillator stabilization delay.

10.5 Clock Generator Module (CGM)

10.5.1 Wait Mode

The clock generator module (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.

10.5.2 Stop Mode

If the OSCENINSTOP bit in the CONFIG2 register is cleared (default), then the STOP instruction disables the CGM (oscillator and phase-locked loop) and holds low all CGM outputs (CGMXCLK, CGMOUT, and CGMINT).

If the OSCENINSTOP bit in the CONFIG2 register is set, then the phase locked loop is shut off, but the oscillator will continue to operate in stop mode.



10.12 Timer Interface Module (TIM1 and TIM2)

10.12.1 Wait Mode

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

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

10.12.2 Stop Mode

The TIM is inactive in stop mode. The STOP instruction does not affect register states or the state of the TIM counter. TIM operation resumes when the MCU exits stop mode after an external interrupt.

10.13 Timebase Module (TBM)

10.13.1 Wait Mode

The timebase module (TBM) remains active after execution of the WAIT instruction. In wait mode, the timebase register is not accessible by the CPU.

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

10.13.2 Stop Mode

The timebase module may remain active after execution of the STOP instruction if the oscillator has been enabled to operate during stop mode through the OSCENINSTOP bit in the CONFIG2 register. The timebase module can be used in this mode to generate a periodic wakeup from stop mode.

If the oscillator has not been enabled to operate in stop mode, the timebase module will not be active during stop mode. In stop mode, the timebase register is not accessible by the CPU.

If the timebase functions are not required during stop mode, reduce the power consumption by stopping the timebase before enabling the STOP instruction.

10.14 Exiting Stop Mode

These events restart the system clocks and load the program counter with the reset vector or with an interrupt vector:

- External reset A 0 on the RST pin resets the MCU and loads the program counter with the contents of locations \$FFFE and \$FFFF.
- External interrupt A high-to-low transition on an external interrupt pin loads the program counter with the contents of locations:
 - \$FFFA and \$FFFB; IRQ pin
 - \$FFE0 and \$FFE1; keyboard interrupt pins
- Low-voltage inhibit (LVI) reset A power supply voltage below the V_{TRIPF} voltage resets the MCU and loads the program counter with the contents of locations \$FFFE and \$FFFF.



Low-Power Modes

Timebase module (TBM) interrupt — A TBM interrupt loads the program counter with the contents
of locations \$FFDC and \$FFDD when the timebase counter has rolled over. This allows the TBM
to generate a periodic wakeup from stop mode.

Upon exit from stop mode, the system clocks begin running after an oscillator stabilization delay. A 12-bit stop recovery counter inhibits the system clocks for 4096 CGMXCLK cycles after the reset or external interrupt.

The short stop recovery bit, SSREC, in the CONFIG1 register controls the oscillator stabilization delay during stop recovery. Setting SSREC reduces stop recovery time from 4096 CGMXCLK cycles to 32 CGMXCLK cycles.

NOTE

Use the full stop recovery time (SSREC = 0) in applications that use an external crystal.



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, LVISOR3, 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 20 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.



Figure 12-4 shows the port A I/O logic.





When bit DDRAx is a 1, reading address \$0000 reads the PTAx data latch. When bit DDRAx is a 0, reading address \$0000 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 12-2 summarizes the operation of the port A pins.

Table 12-2. Port A Pin Functions

PTAPUE	DDRA	ΡΤΑ	I/O Pin	Accesses to DDRA	Access	es to PTA
Bit	Bit	Bit	Mode	Read/Write	Read	Write
1	0	X ⁽¹⁾	Input, V _{DD} ⁽²⁾	DDRA7-DDRA0	Pin	PTA7–PTA0 ⁽³⁾
0	0	Х	Input, Hi-Z ⁽⁴⁾	DDRA7-DDRA0	Pin	PTA7–PTA0 ⁽³⁾
Х	1	Х	Output	DDRA7-DDRA0	PTA7–PTA0	PTA7-PTA0

1. X = Don't care

2. I/O pin pulled up to $V_{\mbox{\scriptsize DD}}$ by internal pullup device

3. Writing affects data register, but does not affect input.

4. Hi-Z = High impedance



Input/Output (I/O) Ports



Chapter 13 Resets and Interrupts

13.1 Introduction

Resets and interrupts are responses to exceptional events during program execution. A reset re-initializes the microcontroller (MCU) to its startup condition. An interrupt vectors the program counter to a service routine.

13.2 Resets

A reset immediately returns the MCU to a known startup condition and begins program execution from a user-defined memory location.

13.2.1 Effects

A reset:

- Immediately stops the operation of the instruction being executed
- Initializes certain control and status bits
- Loads the program counter with a user-defined reset vector address from locations \$FFFE and \$FFFF, \$FEFE and \$FEFF in monitor mode
- Selects CGMXCLK divided by four as the bus clock

13.2.2 External Reset

A 0 applied to the RST pin for a time, t_{RL}, generates an external reset. An external reset sets the PIN bit in the system integration module (SIM) reset status register.

13.2.3 Internal Reset

Sources:

- Power-on reset (POR)
- Computer operating properly (COP)
- Low-power reset circuits
- Illegal opcode
- Illegal address

All internal reset sources pull the RST pin low for 32 CGMXCLK cycles to allow resetting of external devices. The MCU is held in reset for an additional 32 CGMXCLK cycles after releasing the RST pin.

13.2.3.1 Power-On Reset (POR)

A power-on reset (POR) is an internal reset caused by a positive transition on the V_{DD} pin. V_{DD} at the POR must go below V_{POR} to reset the MCU. This distinguishes between a reset and a POR. The POR is not a brown-out detector, low-voltage detector, or glitch detector.

NP

System Integration Module (SIM)

The SIM is responsible for:

- Bus clock generation and control for CPU and peripherals:
 - Stop/wait/reset/break entry and recovery
 - Internal clock control
- Master reset control, including power-on reset (POR) and computer operating properly (COP) timeout
- Interrupt arbitration

Table 15-1 shows the internal signal names used in this section.

Signal Name	Description
CGMXCLK	Buffered version of OSC1 from clock generator module (CGM)
CGMVCLK	PLL output
CGMOUT	PLL-based or OSC1-based clock output from CGM module (Bus clock = CGMOUT divided by two)
IAB	Internal address bus
IDB	Internal data bus
PORRST	Signal from the power-on reset module to the SIM
IRST	Internal reset signal
R/W	Read/write signal

Table 15-1. Signal Name Conventions

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$FE00	SIM Break Status Register (SBSR)	Read: Write:	R	R	R	R	R	R	SBSW Note ⁽¹⁾	R
	See page 187.	Reset:	0	0	0	0	0	0	0	0
			1. Writing a () clears SBS\	Ν.					
	SIM Reset Status Register	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
\$FE01	(SRSR)	Write:								
	See page 188.	POR:	1	0	0	0	0	0	0	0
\$FE03	SIM Break Flag Control Register (SBFCR)	Read: Write:	BCFE	R	R	R	R	R	R	R
	See page 189.	Reset:	0							
	Interrupt Status	Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
\$FE04	Register 1 (INT1) See page 183.	Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
	Interrupt Status	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
\$FE05	Register 2 (INT2)	Write:	R	R	R	R	R	R	R	R
	See page 184.	Reset:	0	0	0	0	0	0	0	0
	Interrupt Status	Read:	0	0	IF20	IF19	IF18	IF17	IF16	IF15
\$FE06	Register 3 (INT3)	Write:	R	R	R	R	R	R	R	R
	See page 184.	Reset:	0	0	0	0	0	0	0	0
				= Unimplem	ented	R	= Reserved			

Figure 15-2. SIM I/O Register Summary

MC68HC908GR16A Data Sheet, Rev. 1.0



System Integration Module (SIM)



Figure 15-7. POR Recovery

15.3.2.2 Computer Operating Properly (COP) Reset

An input to the SIM is reserved for the COP reset signal. The overflow of the COP counter causes an internal reset and sets the COP bit in the SIM reset status register (SRSR). The SIM actively pulls down the RST pin for all internal reset sources.

The COP module is disabled if the $\overline{\text{RST}}$ pin or the $\overline{\text{IRQ}}$ pin is held at V_{TST} while the MCU is in monitor mode. The COP module can be disabled only through combinational logic conditioned with the high voltage signal on the $\overline{\text{RST}}$ or the $\overline{\text{IRQ}}$ pin. This prevents the COP from becoming disabled as a result of external noise. During a break state, V_{TST} on the $\overline{\text{RST}}$ pin disables the COP module.

15.3.2.3 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.

15.3.2.4 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 RST pin for all internal reset sources.

15.3.2.5 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 (RST) is held



Serial Peripheral Interface (SPI) Module







Error Conditions

In this case, an overflow can be missed easily. Since no more SPRF interrupts can be generated until this OVRF is serviced, it is not obvious that bytes are being lost as more transmissions are completed. To prevent this, either enable the OVRF interrupt or do another read of the SPSCR following the read of the SPDR. This ensures that the OVRF was not set before the SPRF was cleared and that future transmissions can set the SPRF bit. Figure 16-11 illustrates this process. Generally, to avoid this second SPSCR read, enable the OVRF to the CPU by setting the ERRIE bit.



Figure 16-11. Clearing SPRF When OVRF Interrupt Is Not Enabled

16.6.2 Mode Fault Error

Setting SPMSTR selects master mode and configures the SPSCK and MOSI pins as outputs and the MISO pin as an input. Clearing SPMSTR selects slave mode and configures the SPSCK and MOSI pins as inputs and the MISO pin as an output. The mode fault bit, MODF, becomes set any time the state of the slave select pin, SS, is inconsistent with the mode selected by SPMSTR.

To prevent SPI pin contention and damage to the MCU, a mode fault error occurs if:

- The SS pin of a slave SPI goes high during a transmission
- The SS pin of a master SPI goes low at any time

For the MODF flag to be set, the mode fault error enable bit (MODFEN) must be set. Clearing the MODFEN bit does not clear the MODF flag but does prevent MODF from being set again after MODF is cleared.

MODF generates a receiver/error CPU interrupt request if the error interrupt enable bit (ERRIE) is also set. The SPRF, MODF, and OVRF interrupts share the same CPU interrupt vector. (See Figure 16-12.) It is not possible to enable MODF or OVRF individually to generate a receiver/error CPU interrupt request. However, leaving MODFEN low prevents MODF from being set.



Timer Interface Module (TIM1 and TIM2)

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

18.4.3 Output Compare

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

18.4.3.1 Unbuffered Output Compare

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

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

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

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

18.4.3.2 Buffered Output Compare

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

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

NOTE

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



Chapter 19 Development Support

19.1 Introduction

This section describes the break module, the monitor module (MON), and the monitor mode entry methods.

19.2 Break Module (BRK)

The break module can generate a break interrupt that stops normal program flow at a defined address to enter a background program.

Features of the break module include:

- Accessible input/output (I/O) registers during the break Interrupt
- Central processor unit (CPU) generated break interrupts
- Software-generated break interrupts
- Computer operating properly (COP) disabling during break interrupts

19.2.1 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal (BKPT) to the system integration module (SIM). The SIM then causes the CPU to load the instruction register with a software interrupt instruction (SWI). The program counter vectors to \$FFFC and \$FFFD (\$FEFC and \$FEFD in monitor mode).

The following events can cause a break interrupt to occur:

- A CPU generated address (the address in the program counter) matches the contents of the break address registers.
- Software writes a 1 to the BRKA bit in the break status and control register.

When a CPU generated address matches the contents of the break address registers, the break interrupt is generated. A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the microcontroller unit (MCU) to normal operation.

Figure 19-2 shows the structure of the break module.

Figure 19-3 provides a summary of the I/O registers.





© FREESCALE SEMICONDUCTOR, INC. All RIGHTS RESERVED.	MECHANICA	L OUTLINE	PRINT VERSION NO	DT TO SCALE
TITLE:		DOCUMENT NO	: 98ASH00962A	REV: G
LQFP, 48 LEAD, 0.	50 PITCH	CASE NUMBER	2: 932–03	14 APR 2005
(/. U X /. U X I. 4)		STANDARD: JE	DEC MS-026-BBC	



How to Reach Us:

Home Page: www.freescale.com

E-mail: support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor Technical Information Center, CH370 1300 N. Alma School Road Chandler, Arizona 85224 +1-800-521-6274 or +1-480-768-2130 support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH Technical Information Center Schatzbogen 7 81829 Muenchen, Germany +44 1296 380 456 (English) +46 8 52200080 (English) +49 89 92103 559 (German) +33 1 69 35 48 48 (French) support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd. Headquarters ARCO Tower 15F 1-8-1, Shimo-Meguro, Meguro-ku, Tokyo 153-0064 Japan 0120 191014 or +81 3 5437 9125 support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd. Technical Information Center 2 Dai King Street Tai Po Industrial Estate Tai Po, N.T., Hong Kong +800 2666 8080 support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center P.O. Box 5405 Denver, Colorado 80217 1-800-441-2447 or 303-675-2140 Fax: 303-675-2150 LDCForFreescaleSemiconductor@hibbertgroup.com RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale[™] and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2004, 2006. All rights reserved.



MC68HC908GR16A Rev. 1.0, 03/2006