NXP USA Inc. - MC908JL16CDWER 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	I²C, SCI
Peripherals	LED, LVD, POR, PWM
Number of I/O	23
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908jl16cdwer

MC68HC908JL16

Data Sheet

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://freescale.com/

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

7.4.2.5 Inversion of Transmitted Output 90
7.4.2.6 Transmitter Interrupts 90
7.4.3 Receiver 90
7.4.3.1 Character Length 90
7.4.3.2 Character Reception 90
7.4.3.3 Data Sampling 92
7.4.3.4 Framing Errors 93
7.4.3.5 Baud Rate Tolerance 94
7.4.3.6 Receiver Wakeup 95
7.4.3.7 Receiver Interrupts 96
7.4.3.8 Error Interrupts 96
7.5 Low-Power Modes 96
7.5.1 Wait Mode 96
7.5.2 Stop Mode 97
7.6 SCI During Break Module Interrupts 97
7.7 I/O Signals 97
7.7.1 TxD (Transmit Data) 97
7.7.2 RxD (Receive Data) 97
7.8 I/O Registers 97
7.8.1 SCI Control Register 1 98
7.8.2 SCI Control Register 2 99
7.8.3 SCI Control Register 3 101
7.8.4 SCI Status Register 1 103
7.8.5 SCI Status Register 2 105
7.8.6 SCI Data Register 106
7.8.7 SCI Baud Rate Register 106

Chapter 8
Multi-Master IIC Interface (MMIIC)

8.1 Introduction 109
8.2 Features 109
8.3 I/O Pins 110
8.4 Functional Description 111
8.4.1 IIC Protocol 111
8.4.2 START Signal 111
8.4.3 Slave Address Transmission 112
8.4.4 Data Transfer 112
8.4.5 STOP Signal 112
8.4.6 Repeated START Signal 113
8.4.7 Arbitration Procedure 113
8.4.8 Clock Synchronization 113
8.4.9 Handshaking 114
8.4.10 Clock Stretching 114
8.4.11 Modes of Operation 114
8.5 Interrupts 114
8.6 Low-Power Modes 114
8.6.1 Wait Mode 114

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0026	TIM1 Channel 0 Register High (T1CH0H)	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0027	TIM1 Channel 0 Register Low (T1CH0L)	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0028	TIM1 Channel 1 Status and Control Register (T1SC1)	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0		▨					
		Reset:	0	0	0	0	0	0	0	0
$0029	TIM1 Channel 1 Register High (T1CH1H)	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$002A	TIM1 Channel 1 Register Low (T1CH1L)	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$002B ↓ $002F	Unimplemented		▨	▨	▨	▨	▨	▨	▨	▨
$0030	TIM2 Status and Control Register (T2SC)	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST	▨			
		Reset:	0	0	1	0	0	0	0	0
$0031	TIM2 Counter Register High (T2CNTH)	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
$0032	TIM2 Counter Register Low (T2CNTL)	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
$0033	TIM2 Counter Modulo Register High (T2MODH)	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0034	TIM2 Counter Modulo Register Low (T2MODL)	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1

U = Unaffected X = Indeterminate ▨ = Unimplemented R = Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 4 of 7)

2.5.6 FLASH 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 to protect blocks of memory from unintentional erase or program operations due to system malfunction. This protection is done by use of a FLASH block protect register (FLBPR). The FLBPR determines the range of the FLASH memory which is to be protected. The range of the protected area starts from a location defined by FLBPR and ends to the bottom of the FLASH memory ($FFFF). When the memory is protected, the HVEN bit cannot be set in either erase or program operations.

NOTE

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

When the FLBPR is program with all 0s, the entire memory is protected from being programmed and erased. When all the bits are erased (all 1s), the entire memory is accessible for program and erase.

When bits within the FLBPR are programmed, they lock a block of memory, address ranges as shown in 2.5.7 FLASH Block Protect Register. Once the FLBPR is programmed with a value other than $FF, any erase or program of the FLBPR or the protected block of FLASH memory is prohibited. The FLBPR itself can be erased or programmed only with an external voltage, V_{TST}, present on the $\overline{IRQ}$ pin. This voltage also allows entry from reset into the monitor mode.

2.5.7 FLASH Block Protect Register

The FLASH block protect register (FLBPR) is implemented as a byte within the FLASH memory, and therefore can only be written during a programming sequence of the FLASH memory. The value in this register determines the starting location of the protected range within the FLASH memory.

Address: $FFCF

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
Reset:	Unaffected by reset; $FF when blank							

Non-volatile FLASH register; write by programming.

Figure 2-5. FLASH Block Protect Register (FLBPR)

BPR[7:0] — FLASH Block Protect Bits

BPR[7:0] represent bits [13:6] of a 16-bit memory address. Bits [15:14] are 1s and bits [5:0] are 0s.

16-bit memory address

Start address of FLASH block protect	1	1									0	0	0	0	0	0
	▨	▨	BPR[7:0]								▨	▨	▨	▨	▨	▨

The resultant 16-bit address is used for specifying the start address of the FLASH memory for block protection. The FLASH is protected from this start address to the end of FLASH memory, at $FFFF. With this mechanism, the protect start address can be XX00, XX40, XX80, or XXC0 (at page boundaries — 64 bytes) within the FLASH memory.

4.3.2 Active Resets from Internal Sources

All internal reset sources actively pull the $\overline{\text{RST}}$ pin low for 32 ICLK cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles (Figure 4-5). An internal reset can be caused by an illegal address, illegal opcode, COP time-out, or POR. (See Figure 4-6. Sources of Internal Reset.) Note that for POR resets, the SIM cycles through 4096 ICLK 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 4-5.

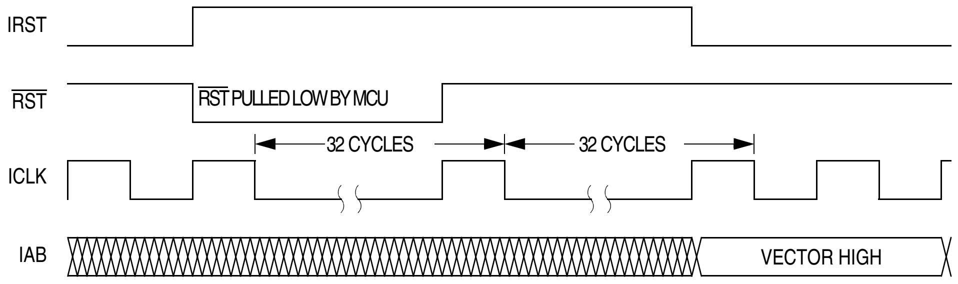

Figure 4-5. Internal Reset Timing

The COP reset is asynchronous to the bus clock.

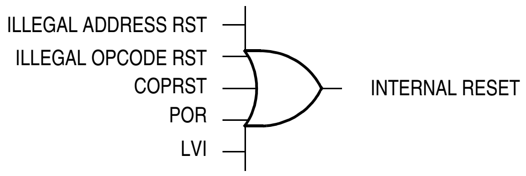

Figure 4-6. Sources of Internal Reset

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

4.3.2.1 Power-On Reset

When power is first applied to the MCU, the power-on reset module (POR) generates a pulse to indicate that power-on has occurred. The external reset pin ($\overline{\text{RST}}$) is held low while the SIM counter counts out 4096 ICLK cycles. Sixty-four ICLK cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur.

At power-on, the following events occur:

- A POR pulse is generated.
- The internal reset signal is asserted.
- The SIM enables OSCOUT.
- Internal clocks to the CPU and modules are held inactive for 4096 ICLK cycles to allow stabilization of the oscillator.
- The $\overline{\text{RST}}$ pin is driven low during the oscillator stabilization time.
- The POR bit of the reset status register (RSR) is set and all other bits in the register are cleared.

Interrupts are latched, and arbitration is performed in the SIM at the start of interrupt processing. The arbitration result is a constant that the CPU uses to determine which vector to fetch. Once an interrupt is latched by the SIM, no other interrupt can take precedence, regardless of priority, until the latched interrupt is serviced (or the I bit is cleared).

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. Figure 4-9 shows interrupt entry timing. Figure 4-10 shows interrupt recovery timing.

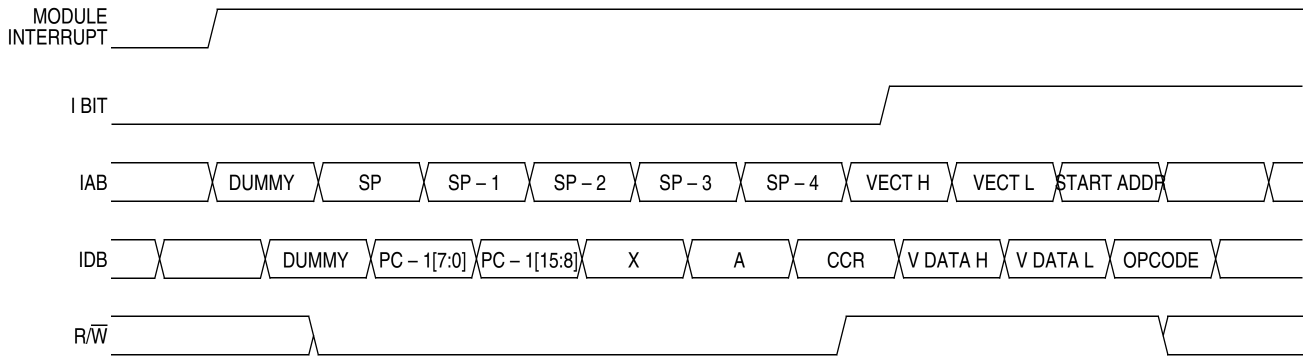

Figure 4-9. Interrupt Entry

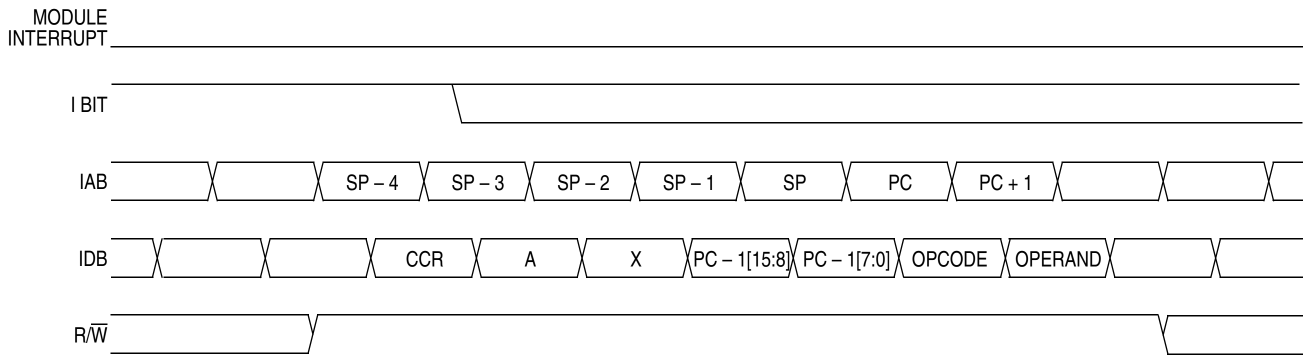

Figure 4-10. Interrupt Recovery

4.5.1.1 Hardware Interrupts

A hardware interrupt does not stop the current instruction. Processing of a hardware interrupt begins after completion of the current instruction. When the current instruction is complete, the SIM checks all pending hardware interrupts. If interrupts are not masked (I bit clear in the condition code register), and if the corresponding interrupt enable bit is set, the SIM proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

Chapter 5
Oscillator (OSC)

5.1 Introduction

The oscillator module provides the reference clocks for the MCU system and bus. Two oscillators are running on the device:

Selectable oscillator — for bus clock

- Crystal oscillator (XTAL) — built-in oscillator that requires an external crystal or ceramic-resonator. This option also allows an external clock that can be driven directly into OSC1.
- RC oscillator (RC) — built-in oscillator that requires an external resistor-capacitor connection only.

The selected oscillator is used to drive the bus clock, the SIM, and other modules on the MCU. The oscillator type is selected by programming a bit FLASH memory. The RC and crystal oscillator cannot run concurrently; one is disabled while the other is selected; because the RC and XTAL circuits share the same OSC1 pin.

Non-selectable oscillator — for COP

- Internal oscillator — built-in RC oscillator that requires no external components.

This internal oscillator is used to drive the computer operating properly (COP) module and the SIM. The internal oscillator runs continuously after a POR or reset, and is always available.

5.2 Oscillator Selection

The oscillator type is selected by programming a bit in a FLASH memory location; the mask option register (MOR), at $FFD0. (See 3.5 Mask Option Register (MOR).)

NOTE

On the ROM device, the oscillator is selected by a ROM-mask layer at factory.

Chapter 8
Multi-Master IIC Interface (MMIIC)

8.1 Introduction

The Multi-master IIC (MMIIC) Interface is designed for internal serial communication between the MCU and other IIC devices. A hardware circuit generates “start” and “stop” signal, while byte by byte data transfer is interrupt driven by the software algorithm. Therefore, it can greatly help the software in dealing with other devices to have higher system efficiency in a typical digital monitor system.

The MMIIC not only can be applied in internal communications, but can also be used as a typical command reception serial bus for factory setup and alignment purposes. It also provides the flexibility of hooking additional devices to an existing system for future expansion without adding extra hardware.

This Multi-master IIC module uses the SCL clock line and the SDA data line to communicate with external DDC host or IIC interface. These two pins are user selectable using the CONFIG2 register (see Figure 3-3. Configuration Register 2 (CONFIG2)) to share either PTA2/PTA3 or PTD6/PTD7 based on their application needs.

The maximum data rate typically is 400k-bps. The maximum communication length and the number of devices that can be connected are limited by a maximum bus capacitance of 400 pF.

NOTE

The outputs of SCL and SDA pins are open-drain type, these pins contain ESD clamping diodes to V_{DD} and therefore cannot be driven to higher than V_{DD} + 0.3 V.

8.2 Features

- Compatibility with multi-master IIC bus standard
- Software controllable acknowledge bit generation
- Interrupt driven byte by byte data transfer
- Calling address identification interrupt
- Auto detection of R/W bit and switching of transmit or receive mode
- Detection of START, repeated START, and STOP signals
- Auto generation of START and STOP condition in master mode
- Arbitration loss detection and No-ACK awareness in master mode
- 8 selectable baud rate master clocks
- Automatic recognition of the received acknowledge bit

NOTE

Route V_{REFH} carefully for maximum noise immunity and place bypass capacitors as near as possible to the package.

AC current in the form of current spikes required to supply charge to the capacitor array at each successive approximation step is drawn through the V_{REFH} and V_{REFL} loop. The best external component to meet this current demand is a 0.1 μF capacitor with good high frequency characteristics. This capacitor is connected between V_{REFH} and V_{REFL} and must be placed as close as possible to the package pins. Resistance in the path is not recommended because the current will cause a voltage drop which could result in conversion errors. Inductance in this path must be minimum (parasitic only).

9.7.4 ADC10 Voltage Reference Low Pin (V_{REFL})

V_{REFL} is the power supply for setting the low-reference voltage for the converter. In some packages, V_{REFL} is connected internally to V_{SSA}. If externally available, connect the V_{REFL} pin to the same voltage potential as V_{SSA}. There will be a brief current associated with V_{REFL} when the sampling capacitor is charging. If externally available, connect the V_{REFL} pin to the same potential as V_{SSA} at the single point ground location.

9.7.5 ADC10 Channel Pins (ADn)

The ADC10 has multiple input channels. Empirical data shows that capacitors on the analog inputs improve performance in the presence of noise or when the source impedance is high. 0.01 μF capacitors with good high-frequency characteristics are sufficient. These capacitors are not necessary in all cases, but when used they must be placed as close as possible to the package pins and be referenced to V_{SSA}.

9.8 Registers

These registers control and monitor operation of the ADC10:

- ADC10 status and control register, ADCSC
- ADC10 data registers, ADRH and ADRL
- ADC10 clock register, ADCLK

9.8.1 ADC10 Status and Control Register

This section describes the function of the ADC10 status and control register (ADCSC). Writing ADCSC aborts the current conversion and initiates a new conversion (if the ADCH[4:0] bits are equal to a value other than all 1s).

Address: $003C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
Write:	▨							
Reset:	0	0	0	1	1	1	1	1

▨ = Unimplemented

Figure 9-3. ADC10 Status and Control Register (ADCSC)

9.8.4 ADC10 Clock Register (ADCLK)

This register selects the clock frequency for the ADC10 and the modes of operation.

Address: $003F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	ADLPC	ADIV1	ADIV0	ADICLK	MODE1	MODE0	ADLSMP	ACLKEN
Reset:	0	0	0	0	0	0	0	0

Figure 9-7. ADC10 Clock Register (ADCLK)

ADLPC — ADC10 Low-Power Configuration Bit

ADLPC controls the speed and power configuration of the successive approximation converter. This is used to optimize power consumption when higher sample rates are not required.

1 = Low-power configuration: The power is reduced at the expense of maximum clock speed.
0 = High-speed configuration

ADIV[1:0] — ADC10 Clock Divider Bits

ADIV1 and ADIV0 select the divide ratio used by the ADC10 to generate the internal clock ADCK. Table 9-3 shows the available clock configurations.

Table 9-3. ADC10 Clock Divide Ratio

ADIV1	ADIV0	Divide Ratio (ADIV)	Clock Rate
0	0	1	Input clock ÷ 1
0	1	2	Input clock ÷ 2
1	0	4	Input clock ÷ 4
1	1	8	Input clock ÷ 8

ADICLK — Input Clock Select Bit

If ACLKEN is clear, ADICLK selects either the bus clock or an alternate clock source as the input clock source to generate the internal clock ADCK. If the alternate clock source is less than the minimum clock speed, use the internally-generated bus clock as the clock source. As long as the internal clock ADCK, which is equal to the selected input clock divided by ADIV, is at a frequency (f_{ADCK}) between the minimum and maximum clock speeds (considering ALPC), correct operation can be guaranteed.

1 = The internal bus clock is selected as the input clock source
0 = The alternate clock source IS SELECTED

MODE[1:0] — 10- or 8-Bit or External-Triggered Mode Selection

This bit selects between 10- or 8-bit operation. The successive approximation converter generates a result which is rounded to 8- or 10-bit value based on the mode selection. This rounding process sets the transfer function to transition at the midpoint between the ideal code voltages, causing a quantization error of 1/2LSB.

Reset returns 8-bit mode.

Table 9-4. Mode Selection

MODE1	MODE0	Mode
0	0	8-bit, right-justified, ADCSC write-triggered mode enabled
0	1	10-bit, right-justified, ADCSC write-triggered mode enabled
1	0	Reserved.
1	1	10-bit, right-justified, external triggered mode enabled

KBIE7–KBIE0 — Port-A Keyboard Interrupt Enable Bits

Each of these read/write bits enables the corresponding keyboard interrupt pin on port-A to latch interrupt requests. Reset clears the keyboard interrupt enable register.

1 = KBIx pin enabled as keyboard interrupt pin

0 = KBIx pin not enabled as keyboard interrupt pin

12.6 Low-Power Modes

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

12.6.1 Wait Mode

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

12.6.2 Stop Mode

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

12.7 Keyboard Module During Break Interrupts

The system integration module (SIM) controls whether the keyboard interrupt latch can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state.

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

To protect the latch during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), writing to the keyboard acknowledge bit (ACKK) in the keyboard status and control register during the break state has no effect.

Chapter 13
Computer Operating Properly (COP)

13.1 Introduction

The computer operating properly (COP) module contains a free-running counter that generates a reset if allowed to overflow. The COP module helps software recover from runaway code. Prevent a COP reset by clearing the COP counter periodically. The COP module can be disabled through the COPD bit in the CONFIG1 register.

13.2 Functional Description

Figure 13-1 shows the structure of the COP module.

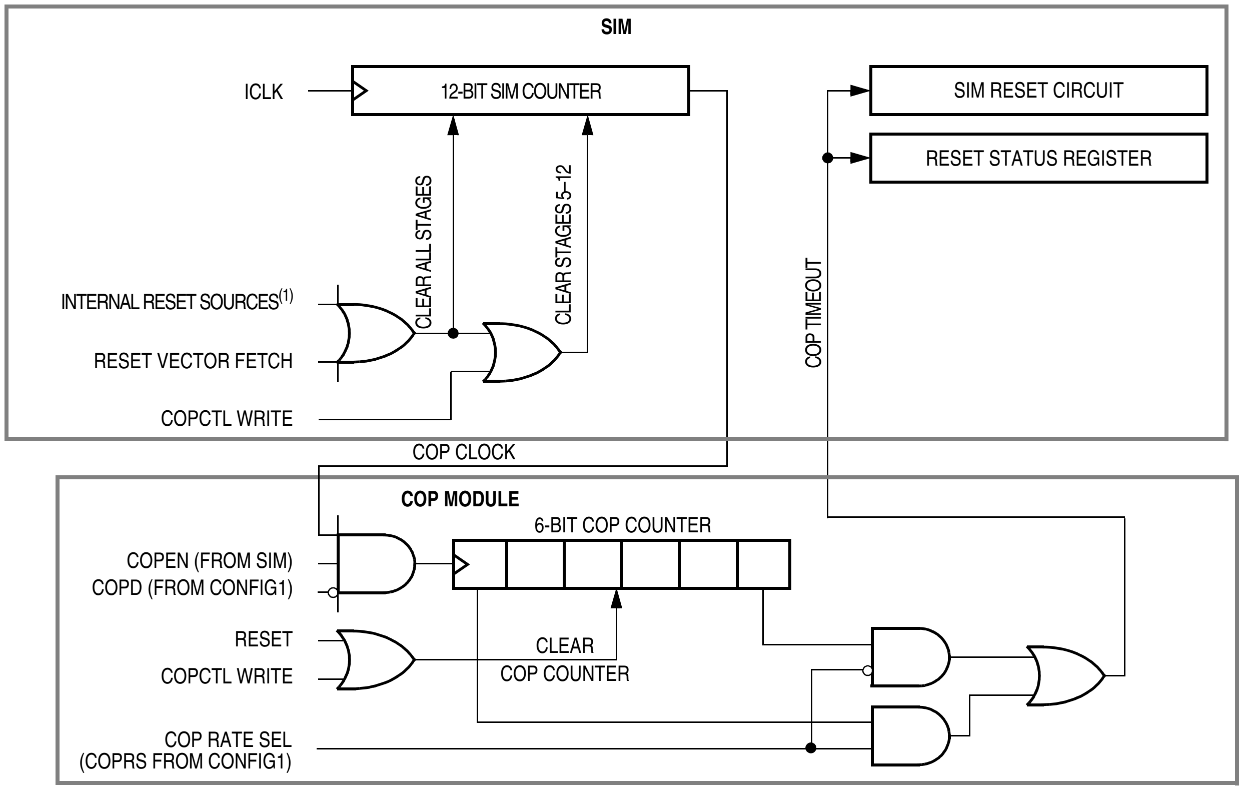

Figure 13-1. COP Block Diagram

13.7.1 Wait Mode

The COP continues to operate during wait mode. To prevent a COP reset during wait mode, periodically clear the COP counter in a CPU interrupt routine.

13.7.2 Stop Mode

Stop mode turns off the ICLK input to the COP and clears the COP prescaler. Service the COP immediately before entering or after exiting stop mode to ensure a full COP timeout period after entering or exiting stop mode.

To prevent inadvertently turning off the COP with a STOP instruction, a configuration option is available that disables the STOP instruction. When the STOP bit in the configuration register has the STOP instruction is disabled, execution of a STOP instruction results in an illegal opcode reset.

13.8 COP Module During Break Mode

The COP is disabled during a break interrupt when V_{TST} is present on the $\overline{\text{RST}}$ pin.

14.4 LVI Control Register (CONFIG2/CONFIG1)

The LVI module is controlled by three bits in the configuration registers, CONFIG1 and CONFIG2.

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	IRQPUD	R	R	LVIT1	LVIT0	R	R	STOP_ ICLKDIS
Reset:	0	0	0	U	U	0	0	0
POR:	0	0	0	0	0	0	0	0

R = Reserved U = Unaffected

Figure 14-2. Configuration Register 2 (CONFIG2)

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	COPRS	R	R	LVID	R	SSREC	STOP	COPD
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 14-3. Configuration Register 1 (CONFIG1)

LVID — Low Voltage Inhibit Disable Bit

LVID disables the LVI module. Reset clears LVID.

1 = Low voltage inhibit disabled

0 = Low voltage inhibit enabled

LVIT1, LVIT0 — LVI Trip Voltage Selection Bits

These two bits determine at which level of V_{DD} the LVI module will come into action. LVIT1 and LVIT0 are cleared by a power-on reset only.

Table 14-1. Trip Voltage Selection

LVIT1	LVIT0	Comments[1]
0	0	For V_{DD} = 3 V operation
0	1	For V_{DD} = 3 V operation
1	0	For V_{DD} = 5 V operation
1	1	Reserved

1. See Chapter 17 Electrical Specifications for full parameters.

14.5 Low-Power Modes

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

14.5.1 Wait Mode

The LVI module, when enabled, will continue to operate in wait mode.

14.5.2 Stop Mode

The LVI module, when enabled, will continue to operate in stop mode.

Table 15-1. Instruction Set Summary (Sheet 3 of 6)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
CLR *opr* CLRA CLRX CLRH CLR *opr*,X CLR ,X CLR *opr*,SP	Clear	M ← $00 A ← $00 X ← $00 H ← $00 M ← $00 M ← $00 M ← $00	0	–	–	0	1	–	DIR INH INH INH IX1 IX SP1	3F 4F 5F 8C 6F 7F 9E6F	dd ff ff	3 1 1 1 3 2 4
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X CMP *opr*,SP CMP *opr*,SP	Compare A with M	(A) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A1 B1 C1 D1 E1 F1 9EE1 9ED1	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
COM *opr* COMA COMX COM *opr*,X COM ,X COM *opr*,SP	Complement (One's Complement)	M ← $(\overline{M})$ = $FF – (M) A ← $(\overline{A})$ = $FF – (M) X ← $(\overline{X})$ = $FF – (M) M ← $(\overline{M})$ = $FF – (M) M ← $(\overline{M})$ = $FF – (M) M ← $(\overline{M})$ = $FF – (M)	0	–	–	↕	↕	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd ff ff	4 1 1 4 3 5
CPHX #*opr* CPHX *opr*	Compare H:X with M	(H:X) – (M:M + 1)	↕	–	–	↕	↕	↕	IMM DIR	65 75	ii ii+1 dd	3 4
CPX #*opr* CPX *opr* CPX *opr* CPX ,X CPX *opr*,X CPX *opr*,X CPX *opr*,SP CPX *opr*,SP	Compare X with M	(X) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A3 B3 C3 D3 E3 F3 9EE3 9ED3	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
DAA	Decimal Adjust A	$(A)_{10}$	U	–	–	↕	↕	↕	INH	72		2
DBNZ *opr,rel* DBNZA *rel* DBNZX *rel* DBNZ *opr*,X,*rel* DBNZ X,*rel* DBNZ *opr*,SP,*rel*	Decrement and Branch if Not Zero	A ← (A) – 1 or M ← (M) – 1 or X ← (X) – 1 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 4 + *rel* ? (result) ≠ 0	–	–	–	–	–	–	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	5 3 3 5 4 6
DEC *opr* DECA DECX DEC *opr*,X DEC ,X DEC *opr*,SP	Decrement	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1 M ← (M) – 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff	4 1 1 4 3 5
DIV	Divide	A ← (H:A)/(X) H ← Remainder	–	–	–	–	↕	↕	INH	52		7
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X EOR *opr*,SP EOR *opr*,SP	Exclusive OR M with A	A ← (A ⊕ M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A8 B8 C8 D8 E8 F8 9EE8 9ED8	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
INC *opr* INCA INCX INC *opr*,X INC ,X INC *opr*,SP	Increment	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1 M ← (M) + 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff	4 1 1 4 3 5

16.3.9.4 MON_PRGRNGE

In monitor mode, MON_PRGRNGE is used to program a range of FLASH locations with data loaded into the data array.

Table 16-14. MON_PRGRNGE Routine

Routine Name	MON_PRGRNGE
Routine Description	Program a range of locations, in monitor mode
Calling Address	$FC28
Stack Used	13 bytes
Data Block Format	Bus speed Data size Starting address (high byte) Starting address (low byte) Data 1 : Data N

The MON_PRGRNGE routine is designed to be used in monitor mode. It performs the same function as the PRGRNGE routine (see 16.3.9.1 PRGRNGE), except that MON_PRGRNGE returns to the main program via an SWI instruction. After a MON_PRGRNGE call, the SWI instruction will return the control back to the monitor code.

16.3.9.5 MON_ERARNGE

In monitor mode, ERARNGE is used to erase a range of locations in FLASH.

Table 16-15. MON_ERARNGE Routine

Routine Name	MON_ERARNGE
Routine Description	Erase a page or the entire array, in monitor mode
Calling Address	$FF2C
Stack Used	9 bytes
Data Block Format	Bus speed Data size Starting address (high byte) Starting address (low byte)

The MON_ERARNGE routine is designed to be used in monitor mode. It performs the same function as the ERARNGE routine (see 16.3.9.2 ERARNGE), except that MON_ERARNGE returns to the main program via an SWI instruction. After a MON_ERARNGE call, the SWI instruction will return the control back to the monitor code.

start of boundary address (the page start address: $XX00, $XX40, $XX80, or $00C0) and DATASIZE must be the same size when accessing the same page.

In some applications, the user may want to repeatedly store and read a set of data from an area of non-volatile memory. This can be easily implemented when EEPROM memory is used because the byte erase is allowed in EEPROM. On the other hand in FLASH memory, a minimum erase size is a page (64 bytes), so unused locations in a page will be wasted when it is used for data storage.

The EE_WRITE routine is designed to emulate EEPROM using FLASH. This allows a FLASH page to implement data storage more efficiently. Each call of the EE_WRITE routine will automatically transfer the data in the data array (in RAM) to the next available blank locations in a page. Once the page is filled up with data, the EE_WRITE routine automatically erases the page and programs updated data in the same page. In a FLASH page, data is programmed to FLASH with in a block that consists of the data array and one boundary byte. The boundary byte contains the remaining number of bytes which can be programmed in the page (see Figure 16-16).

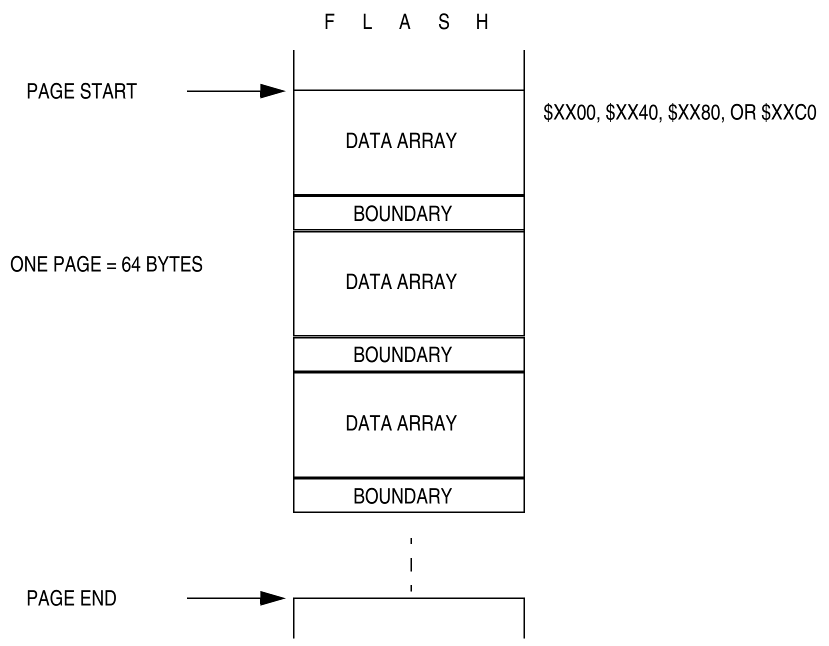

Figure 16-16. EE_WRITE FLASH Memory Usage

When using this routine to store a 3-byte data array, the FLASH page can be programmed 16 times before the an erase is required. In effect, the write/erase endurance is increased by 16 times. When a 15-byte data array is used, the write/erase endurance is increased by 4 times. Due to the FLASH page size limitation, the data array is limited from 2 bytes to 15 bytes.

The coding example below uses the $EF00–$EE3F page for data storage. The data array size is 15 bytes, and the bus speed is 4.9152 MHz. The coding assumes the data block is already loaded in RAM, with the address pointer, FILE_PTR, pointing to the first byte of the data block.

Table 17-7. DC Electrical Characteristics (3 V)

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Low-voltage inhibit, trip falling voltage	V_{TRIPF}	2.40	2.55	2.70	V
Low-voltage inhibit, trip rising voltage	V_{TRIPR}	2.475	2.625	2.775	V
Low-voltage inhibit reset/recovery hysteresis	V_{HYS}	—	75	—	mV

1. V_{DD} = 2.7 to 3.3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
3. Run (operating) I_{DD} measured using external square wave clock source (f_{OP} = 4 MHz). All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD}. Measured with all modules enabled.
4. Wait I_{DD} measured using external square wave clock source (f_{OP} = 4 MHz). All inputs 0.2 V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects wait I_{DD}.
5. Stop I_{DD} measured with OSC1 grounded; no port pins sourcing current.
6. Maximum is highest voltage that POR is guaranteed.
7. If minimum V_{DD} is not reached before the internal POR reset is released, $\overline{RST}$ must be driven low externally until minimum V_{DD} is reached.
8. R_{PU} is measured at V_{DD} = 5.0 V.

17.9 3-V Control Timing

Table 17-8. Control Timing (3 V)

Characteristic[1]	Symbol	Min	Max	Unit
Internal operating frequency[2]	f_{OP}	—	4	MHz
$\overline{RST}$ input pulse width low[3]	t_{IRL}	1.5	—	μs
$\overline{IRQ}$ input pulse width low[3]	t_{IIL}	1.5	—	μs
TIM2 external clock input	f_{T2CLK}	—	2	MHz

1. V_{DD} = 2.7 to 3.3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H; timing shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Some modules may require a minimum frequency greater than dc for proper operation; see appropriate table for information.
3. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.

17.14 MMIIC Electrical Characteristics

Table 17-12. MMIIC DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ	Max	Unit	Comments
Input low	V_{IL}	–0.5	—	0.8	V	Data, clock input low.
Input high	V_{IH}	2.1	—	5.5	V	Data, clock input high.
Output low	V_{OL}	—	—	0.4	V	Data, clock output low; @$I_{PULLUP,MAX}$
Input leakage	I_{LEAK}	—	—	± 5	μA	Input leakage current
Pullup current	I_{PULLUP}	100	—	350	μA	Current through pull-up resistor or current source. See note.[2]

1. V_{DD} = 2.7 to 5.5Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted.
2. The I_{PULLUP} (max) specification is determined primarily by the need to accommodate a maximum of 1.1kΩ equivalent series resistor of removable SMBus devices, such as the smart battery, while maintaining the V_{OL} (max) of the bus.

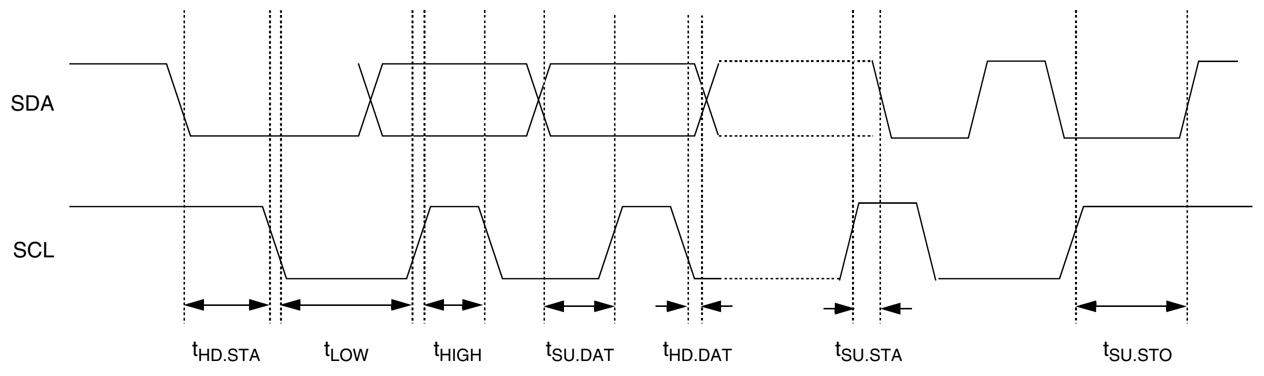

Figure 17-7. MMIIC Signal Timings

See Table 17-13 for MMIIC timing parameters.