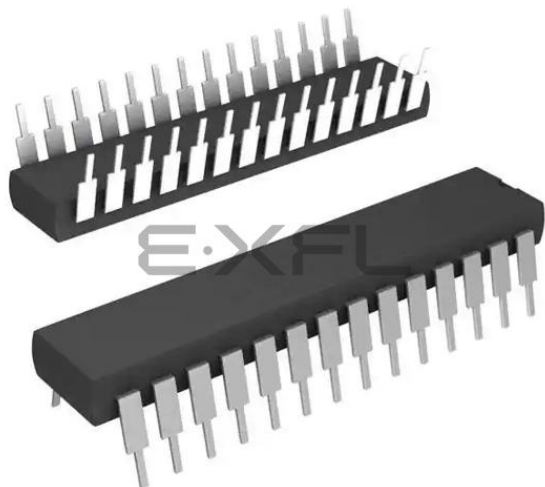




Welcome to [E-XFL.COM](#)

What is "[Embedded - Microcontrollers](#)"?

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

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LED, LVD, POR, PWM
Number of I/O	23
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.3V
Data Converters	A/D 12x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.600", 15.24mm)
Supplier Device Package	28-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908jl3cp

4.4	FLASH Control Register	40
4.5	FLASH Block Erase Operation	41
4.6	FLASH Mass Erase Operation	42
4.7	FLASH Program Operation	43
4.8	FLASH Protection	44
4.9	FLASH Block Protect Register	46

Section 5. Configuration Register (CONFIG)

5.1	Contents	47
5.2	Introduction	47
5.3	Functional Description	48

Section 6. Central Processor Unit (CPU)

6.1	Contents	51
6.2	Introduction	51
6.3	Features	52
6.4	CPU Registers	52
6.4.1	Accumulator	53
6.4.2	Index Register	54
6.4.3	Stack Pointer	54
6.4.4	Program Counter	55
6.4.5	Condition Code Register	55
6.5	Arithmetic/Logic Unit (ALU)	58
6.6	Low-Power Modes	58
6.6.1	Wait Mode	58
6.6.2	Stop Mode	59
6.7	CPU During Break Interrupts	59
6.8	Instruction Set Summary	59
6.9	Opcode Map	59

Section 19. Mechanical Specifications

19.1	Contents	203
19.2	Introduction	203
19.3	20-Pin PDIP	204
19.4	20-Pin SOIC	204
19.5	28-Pin PDIP	205
19.6	28-Pin SOIC	205

Section 20. Ordering Information

20.1	Contents	207
20.2	Introduction	207
20.3	MC Order Numbers	208

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0000	Port A Data Register (PTA)	Read:	0	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:	Unaffected by reset							
\$0001	Port B Data Register (PTB)	Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:	Unaffected by reset							
\$0002	Unimplemented	Read:								
		Write:								
\$0003	Port D Data Register (PTD)	Read:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Write:								
		Reset:	Unaffected by reset							
\$0004	Data Direction Register A (DDRA)	Read:	0	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0005	Data Direction Register B (DDRB)	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0006	Unimplemented	Read:								
		Write:								
\$0007	Data Direction Register D (DDRD)	Read:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0008 ↓ \$0009	Unimplemented	Read:								
		Write:								
\$000A	Port D Control Register (PDCR)	Read:	0	0	0	0	SLOWD7	SLOWD6	PTDPU7	PTDPU6
		Write:								
		Reset:	0	0	0	0	0	0	0	0
				= Unimplemented			R	= Reserved		

Figure 2-2. Control, Status, and Data Registers (Sheet 1 of 5)



Section 6. Central Processor Unit (CPU)

6.1 Contents

6.2	Introduction	51
6.3	Features	52
6.4	CPU Registers	52
6.4.1	Accumulator	53
6.4.2	Index Register	54
6.4.3	Stack Pointer	54
6.4.4	Program Counter	55
6.4.5	Condition Code Register	55
6.5	Arithmetic/Logic Unit (ALU)	58
6.6	Low-Power Modes	58
6.6.1	Wait Mode	58
6.6.2	Stop Mode	59
6.7	CPU During Break Interrupts	59
6.8	Instruction Set Summary	59
6.9	Opcode Map	59

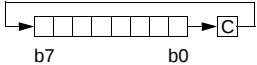
6.2 Introduction

The M68HC08 CPU (central processor unit) is an enhanced and fully object-code-compatible version of the M68HC05 CPU. The *CPU08 Reference Manual* (Freescale document order number CPU08RM/AD) contains a description of the CPU instruction set, addressing modes, and architecture.

Table 6-1. Instruction Set Summary

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
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)	$M \leftarrow (\overline{M}) = \$FF - (M)$ $A \leftarrow (\overline{A}) = \$FF - (M)$ $X \leftarrow (\overline{X}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$	0	–	–	↑	↑	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd ff 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 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	$A \leftarrow (A) - 1$ or $M \leftarrow (M) - 1$ or $X \leftarrow (X) - 1$ $PC \leftarrow (PC) + 3 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 2 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 2 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 3 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 2 + rel ? (result) \neq 0$ $PC \leftarrow (PC) + 4 + rel ? (result) \neq 0$	–	–	–	–	–	–	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	$M \leftarrow (M) - 1$ $A \leftarrow (A) - 1$ $X \leftarrow (X) - 1$ $M \leftarrow (M) - 1$ $M \leftarrow (M) - 1$ $M \leftarrow (M) - 1$	↑	–	–	↑	↑	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff	4 1 1 4 3 5
DIV	Divide	$A \leftarrow (H:A)/(X)$ H ← Remainder	–	–	–	–	↑	↑	INH	52		7

Table 6-1. Instruction Set Summary

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
ROR <i>opr</i> RORA RORX ROR <i>opr</i> ,X ROR ,X ROR <i>opr</i> ,SP	Rotate Right through Carry		↑	–	–	↑	↑	↑	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	4 1 1 4 3 5
RSP	Reset Stack Pointer	SP ← \$FF	–	–	–	–	–	–	INH	9C		1
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↑	↑	↑	↑	↑	↑	INH	80		7
RTS	Return from Subroutine	SP ← SP + 1; Pull (PCH) SP ← SP + 1; Pull (PCL)	–	–	–	–	–	–	INH	81		4
SBC # <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> ,X SBC <i>opr</i> ,X SBC ,X SBC <i>opr</i> ,SP SBC <i>opr</i> ,SP	Subtract with Carry	A ← (A) – (M) – (C)	↑	–	–	↑	↑	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A2 B2 C2 D2 E2 F2 9EE2 9ED2	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
SEC	Set Carry Bit	C ← 1	–	–	–	–	–	1	INH	99		1
SEI	Set Interrupt Mask	I ← 1	–	–	1	–	–	–	INH	9B		2
STA <i>opr</i> STA <i>opr</i> STA <i>opr</i> ,X STA <i>opr</i> ,X STA ,X STA <i>opr</i> ,SP STA <i>opr</i> ,SP	Store A in M	M ← (A)	0	–	–	↑	↑	–	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh ll ee ff ff ff ff ee ff	3 4 4 3 2 4 5
STHX <i>opr</i>	Store H:X in M	(M:M + 1) ← (H:X)	0	–	–	↑	↑	–	DIR	35	dd	4
STOP	Enable $\overline{\text{IRQ}}$ Pin; Stop Oscillator	I ← 0; Stop Oscillator	–	–	0	–	–	–	INH	8E		1
STX <i>opr</i> STX <i>opr</i> STX <i>opr</i> ,X STX <i>opr</i> ,X STX ,X STX <i>opr</i> ,SP STX <i>opr</i> ,SP	Store X in M	M ← (X)	0	–	–	↑	↑	–	DIR EXT IX2 IX1 IX SP1 SP2	BF CF DF EF FF 9EEF 9EDF	dd hh ll ee ff ff ff ff ee ff	3 4 4 3 2 4 5

7.5.3 SIM Counter and Reset States

External reset has no effect on the SIM counter. (See [7.7.2 Stop Mode](#) for details.) The SIM counter is free-running after all reset states. (See [7.4.2 Active Resets from Internal Sources](#) for counter control and internal reset recovery sequences.)

7.6 Exception Control

Normal, sequential program execution can be changed in three different ways:

- Interrupts
 - Maskable hardware CPU interrupts
 - Non-maskable software interrupt instruction (SWI)
- Reset
- Break interrupts

7.6.1 Interrupts

An interrupt temporarily changes the sequence of program execution to respond to a particular event. [Figure 7-8](#) flow charts the handling of system interrupts.

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).

8.7 Oscillator During Break Mode

The oscillator continues to drive OSCOUT and 2OSCOUT when the device enters the break state.

Section 10. Timer Interface Module (TIM)

10.1 Contents

10.2	Introduction	116
10.3	Features	116
10.4	Pin Name Conventions	116
10.5	Functional Description	117
10.5.1	TIM Counter Prescaler	119
10.5.2	Input Capture	119
10.5.3	Output Compare	119
10.5.3.1	Unbuffered Output Compare	120
10.5.3.2	Buffered Output Compare	120
10.5.4	Pulse Width Modulation (PWM)	121
10.5.4.1	Unbuffered PWM Signal Generation	122
10.5.4.2	Buffered PWM Signal Generation	123
10.5.4.3	PWM Initialization	124
10.6	Interrupts	125
10.7	Wait Mode	125
10.8	TIM During Break Interrupts	126
10.9	I/O Signals	126
10.10	I/O Registers	127
10.10.1	TIM Status and Control Register (TSC)	127
10.10.2	TIM Counter Registers (TCNTH:TCNTL)	129
10.10.3	TIM Counter Modulo Registers (TMODH:TMODL)	130
10.10.4	TIM Channel Status and Control Registers (TSC0:TSC1)	131
10.10.5	TIM Channel Registers (TCH0H/L:TCH1H/L)	135

Timer Interface Module (TIM)

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$0020	TIM Status and Control Register (TSC)	Read: TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write: 0			TRST				
		Reset: 0	0	1	0	0	0	0	0
\$0021	TIM Counter Register High (TCNTH)	Read: Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:							
		Reset: 0	0	0	0	0	0	0	0
\$0022	TIM Counter Register Low (TCNTL)	Read: Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:							
		Reset: 0	0	0	0	0	0	0	0
\$0023	TIM Counter Modulo Register High (TMODH)	Read: Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:							
		Reset: 1	1	1	1	1	1	1	1
\$0024	TIM Counter Modulo Register Low (TMDL)	Read: Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:							
		Reset: 1	1	1	1	1	1	1	1
\$0025	TIM Channel 0 Status and Control Register (TSC0)	Read: CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write: 0							
		Reset: 0	0	0	0	0	0	0	0
\$0026	TIM Channel 0 Register High (TCH0H)	Read: Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Write:							
		Reset:	Indeterminate after reset						
\$0027	TIM Channel 0 Register Low (TCH0L)	Read: Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Write:							
		Reset:	Indeterminate after reset						
\$0028	TIM Channel 1 Status and Control Register (TSC1)	Read: CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write: 0							
		Reset: 0	0	0	0	0	0	0	0

Figure 10-2. TIM I/O Register Summary

write a new, smaller pulse width 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 PWM pulse width on channel x:

- When changing to a shorter pulse width, 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 pulse. The interrupt routine has until the end of the PWM period to write the new value.
- When changing to a longer pulse width, enable channel x 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 PWM 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 PWM period.

NOTE: *In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare also can cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.*

10.5.4.2 Buffered PWM Signal Generation

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

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The TIM channel 0 registers initially control the pulse width on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIM channel registers (0 or 1) that control the pulse width are the ones written to last. TSC0 controls and monitors the buffered PWM function, and TIM channel 1 status and control register

Table 10-2. Prescaler Selection

PS2	PS1	PS0	TIM Clock Source
0	0	0	Internal Bus Clock ÷ 1
0	0	1	Internal Bus Clock ÷ 2
0	1	0	Internal Bus Clock ÷ 4
0	1	1	Internal Bus Clock ÷ 8
1	0	0	Internal Bus Clock ÷ 16
1	0	1	Internal Bus Clock ÷ 32
1	1	0	Internal Bus Clock ÷ 64
1	1	1	Not available

10.10.2 TIM Counter Registers (TCNTH:TCNTL)

The two read-only TIM counter registers contain the high and low bytes of the value in the TIM counter. Reading the high byte (TCNTH) latches the contents of the low byte (TCNTL) into a buffer. Subsequent reads of TCNTH do not affect the latched TCNTL value until TCNTL is read. Reset clears the TIM counter registers. Setting the TIM reset bit (TRST) also clears the TIM counter registers.

NOTE: *If you read TCNTH during a break interrupt, be sure to unlatch TCNTL by reading TCNTL before exiting the break interrupt. Otherwise, TCNTL retains the value latched during the break.*

11.4.5 Accuracy and Precision

The conversion process is monotonic and has no missing codes.

11.5 Interrupts

When the AIEN bit is set, the ADC module is capable of generating a CPU interrupt after each ADC conversion. A CPU interrupt is generated if the COCO bit is at logic 0. The COCO bit is not used as a conversion complete flag when interrupts are enabled.

11.6 Low-Power Modes

The following subsections describe the ADC in low-power modes.

11.6.1 Wait Mode

The 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 the CH[4:0] bits in the ADC Status and Control register to logic 1's before executing the WAIT instruction.

11.6.2 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. Allow one conversion cycle to stabilize the analog circuitry before attempting a new ADC conversion after exiting stop mode.

11.7 I/O Signals

The ADC module has 12 channels that are shared with I/O port B and port D.

If the MODEK bit is clear, the keyboard interrupt pin is falling-edge-sensitive only. With MODEK clear, a vector fetch or software clear immediately clears the keyboard interrupt request.

Reset clears the keyboard interrupt request and the MODEK bit, clearing the interrupt request even if a keyboard interrupt pin stays at logic 0.

The keyboard flag bit (KEYF) in the keyboard status and control register can be used to see if a pending interrupt exists. The KEYF bit is not affected by the keyboard interrupt mask bit (IMASKK) which makes it useful in applications where polling is preferred.

To determine the logic level on a keyboard interrupt pin, disable the pull-up device, use the data direction register to configure the pin as an input and then read the data register.

NOTE: *Setting a keyboard interrupt enable bit (KBIE_x) forces the corresponding keyboard interrupt pin to be an input, overriding the data direction register. However, the data direction register bit must be a logic 0 for software to read the pin.*

14.4.1 Keyboard Initialization

When a keyboard interrupt pin is enabled, it takes time for the internal pull-up to reach a logic 1. Therefore a false interrupt can occur as soon as the pin is enabled.

To prevent a false interrupt on keyboard initialization:

1. Mask keyboard interrupts by setting the IMASKK bit in the keyboard status and control register.
2. Enable the KBI pins by setting the appropriate KBIE_x bits in the keyboard interrupt enable register.
3. Write to the ACKK bit in the keyboard status and control register to clear any false interrupts.
4. Clear the IMASKK bit.

An interrupt signal on an edge-triggered pin can be acknowledged immediately after enabling the pin. An interrupt signal on an edge- and

14.5 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.

14.6 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.

14.7 Keyboard Module During Break Interrupts

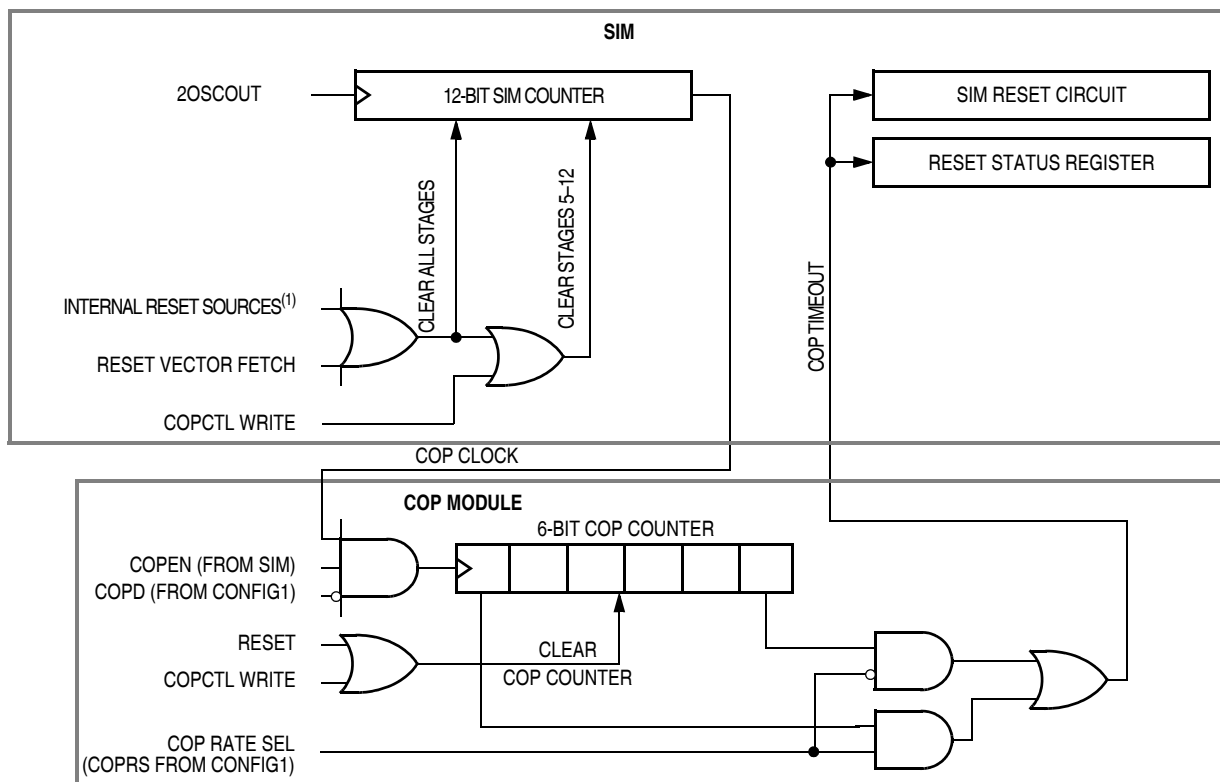
The system integration module (SIM) controls whether the keyboard interrupt latch can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state.

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.

15.3 Functional Description

Figure 15-1 shows the structure of the COP module.



NOTE:

1. See SIM section for more details.

Figure 15-1. COP Block Diagram

The COP counter is a free-running 6-bit counter preceded by the 12-bit system integration module (SIM) counter. If not cleared by software, the COP counter overflows and generates an asynchronous reset after $2^{18} - 2^4$ or $2^{13} - 2^4$ 20SCOUT cycles; depending on the state of the COP rate select bit, COPRS, in configuration register 1. With a $2^{18} - 2^4$ 20SCOUT cycle overflow option, a 8MHz crystal gives a COP timeout period of 32.766 ms. Writing any value to location \$FFFF before an overflow occurs prevents a COP reset by clearing the COP counter and stages 12 through 5 of the SIM counter.

18.12 Typical Supply Currents

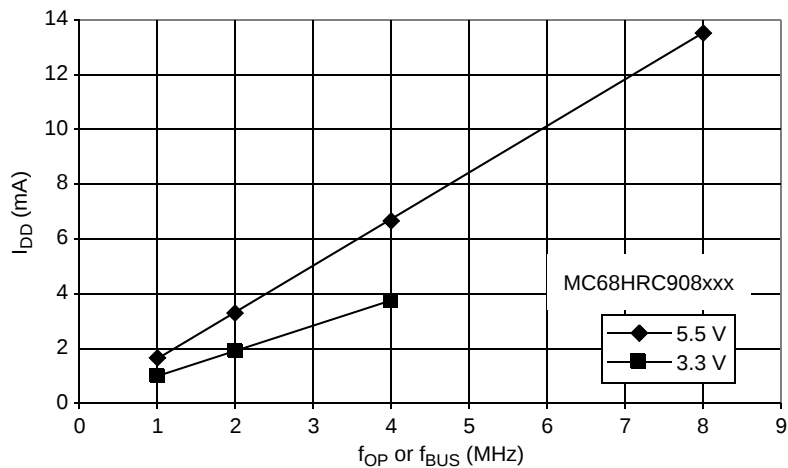


Figure 18-3. Typical Operating I_{DD} , with All Modules Turned On (25 °C)

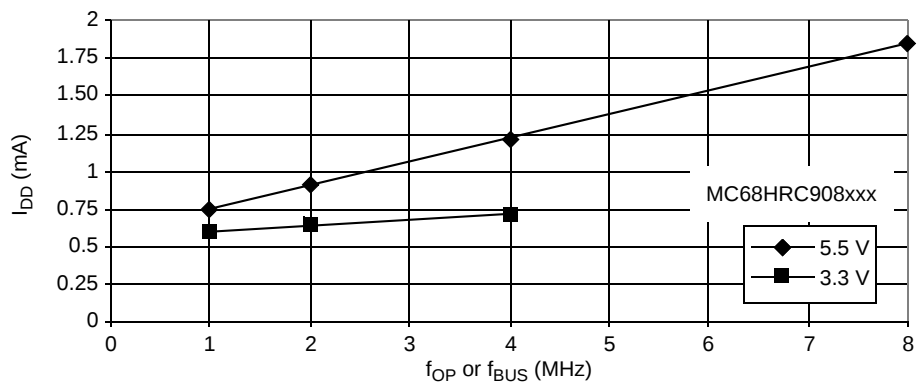


Figure 18-4. Typical Wait Mode I_{DD} , with ADC Turned On (25 °C)

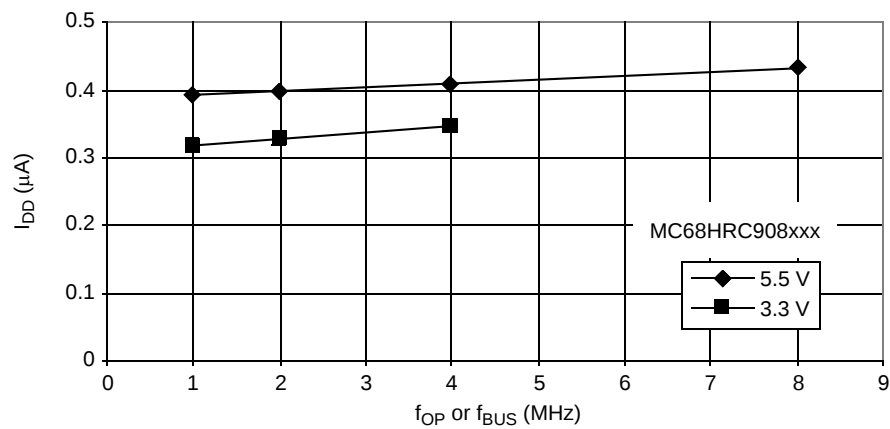


Figure 18-5. Typical Stop Mode I_{DD} , with all Modules Disabled (25 °C)

18.13 ADC Characteristics

Table 18-10. ADC Characteristics

Characteristic	Symbol	Min	Max	Unit	Comments
Supply voltage	V_{DDAD}	2.7 (V_{DD} min)	5.5 (V_{DD} max)	V	
Input voltages	V_{ADIN}	V_{SS}	V_{DD}	V	
Resolution	B_{AD}	8	8	Bits	
Absolute accuracy	A_{AD}	± 0.5	± 1.5	LSB	Includes quantization
ADC internal clock	f_{ADIC}	0.5	1.048	MHz	$t_{AIC} = 1/f_{ADIC}$, tested only at 1 MHz
Conversion range	R_{AD}	V_{SS}	V_{DD}	V	
Power-up time	t_{ADPU}	16		t_{AIC} cycles	
Conversion time	t_{ADC}	16	17	t_{AIC} cycles	
Sample time ⁽¹⁾	t_{ADS}	5	—	t_{AIC} cycles	
Zero input reading ⁽²⁾	Z_{ADI}	00	01	Hex	$V_{IN} = V_{SS}$
Full-scale reading ⁽³⁾	F_{ADI}	FE	FF	Hex	$V_{IN} = V_{DD}$
Input capacitance	C_{ADI}	—	(20) 8	pF	Not tested
Input leakage ⁽³⁾ Port B/port D	—	—	± 1	μA	

NOTES:

1. Source impedances greater than 10 k Ω adversely affect internal RC charging time during input sampling.
2. Zero-input/full-scale reading requires sufficient decoupling measures for accurate conversions.
3. The external system error caused by input leakage current is approximately equal to the product of R source and input current.

Section 19. Mechanical Specifications

19.1 Contents

19.2	Introduction	203
19.3	20-Pin PDIP	204
19.4	20-Pin SOIC	204
19.5	28-Pin PDIP	205
19.6	28-Pin SOIC	205

19.2 Introduction

This section gives the dimensions for:

- 20-pin plastic dual in-line package (case #738)
- 20-pin small outline integrated circuit package (case #751D)
- 28-pin plastic dual in-line package (case #710)
- 28-pin small outline integrated circuit package (case #751F)

The following figures show the latest package drawings at the time of this publication. To make sure that you have the latest package specifications, please visit the Freescale website at <http://freescale.com>. Follow or Worldwide Web on-line instructions to retrieve the current mechanical specifications.