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	15
Program Memory Size	1.5KB (1.5K 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	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908jk1cdw

Table of Contents

Section 1. General Description

Section 2. Memory

Section 3. Random-Access Memory (RAM)

Section 4. FLASH Memory (FLASH)

Section 5. Configuration Register (CONFIG)

Section 6. Central Processor Unit (CPU)

Section 10. Timer Interface Module (TIM)

Section 19. Mechanical Specifications

Section 20. Ordering Information

List of Figures

Figure	Title	Page

4.7 FLASH Program Operation

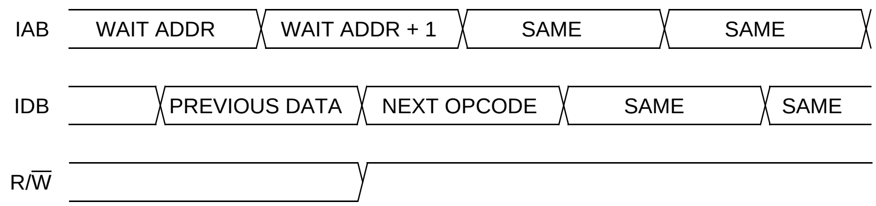
Programming of the FLASH memory is done on a row basis. A row consists of 32 consecutive bytes starting from addresses $XX00, $XX20, $XX40, $XX60, $XX80, $XXA0, $XXC0 or $XXE0. Use this step-by-step procedure to program a row of FLASH memory: (**Figure 4-2** shows a flowchart of the programming algorithm.)

NOTE: *In order to avoid program disturbs, the row must be erased before any byte on that row is programmed.*

1. Set the PGM bit. This configures the memory for program operation and enables the latching of address and data for programming.
2. Write any data to any FLASH location within the address range of the row to be programmed.
3. Wait for a time, t_{nvs} (10μs).
4. Set the HVEN bit.
5. Wait for a time, t_{pgs} (5μs).
6. Write data to the byte being programmed.
7. Wait for time, t_{PROG} (30μs).
8. Repeat step 6 and 7 until all the bytes within the row are programmed.
9. Clear the PGM bit.
10. Wait for time, t_{nvh} (5μs).
11. Clear the HVEN bit.
12. After time, t_{rcv} (1μs), the memory can be accessed in read mode again.

This program sequence is repeated throughout the memory until all data is programmed.

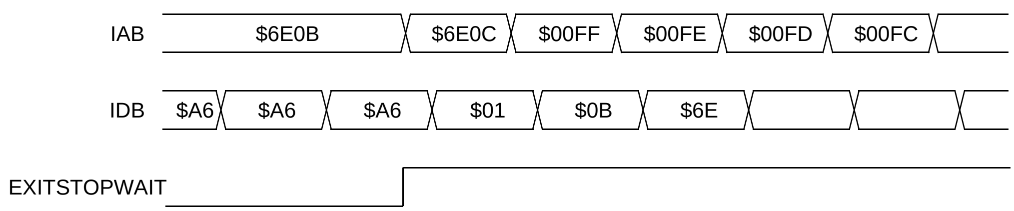
status register (BSR). If the COP disable bit, COPD, in the mask option register is logic zero, then the computer operating properly module (COP) is enabled and remains active in wait mode.



NOTE: Previous data can be operand data or the WAIT opcode, depending on the last instruction.

Figure 7-15. Wait Mode Entry Timing

Figure 7-16 and **Figure 7-17** show the timing for WAIT recovery.



NOTE: EXITSTOPWAIT = $\overline{RST}$ pin OR CPU interrupt OR break interrupt

Figure 7-16. Wait Recovery from Interrupt or Break

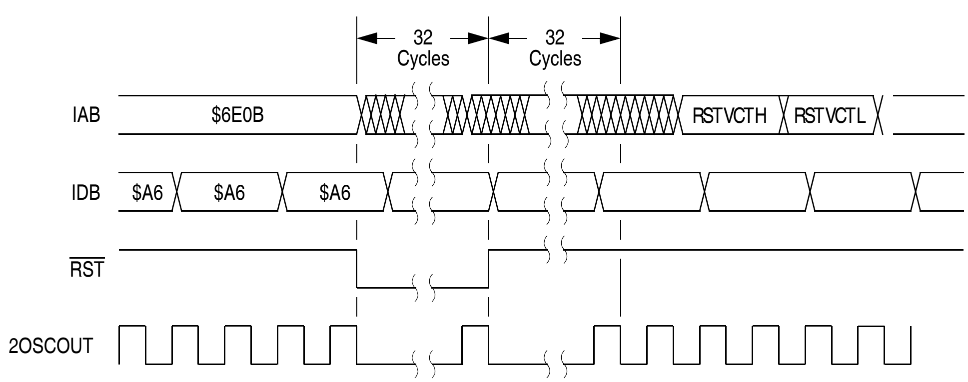


Figure 7-17. Wait Recovery from Internal Reset

Section 9. Monitor ROM (MON)

9.1 Contents

9.2 Introduction

This section describes the monitor ROM (MON) and the monitor mode entry methods. The monitor ROM allows complete testing of the MCU through a single-wire interface with a host computer. This mode is also used for programming and erasing of FLASH memory in the MCU. Monitor mode entry can be achieved without use of the higher test voltage, $V_{DD} + V_{HI}$, as long as vector addresses $FFFE and $FFFF are blank, thus reducing the hardware requirements for in-circuit programming.

Addr.	Register Name									
$0029	TIM Channel 1 Register High (TCH1H)	Read: / Write:	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
		Reset:	Indeterminate after reset							
$002A	TIM Channel 1 Register Low (TCH1L)	Read: / Write:	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
		Reset:	Indeterminate after reset							

= Unimplemented

Figure 10-2. TIM I/O Register Summary

10.5.1 TIM Counter Prescaler

The TIM clock source is one of the seven prescaler outputs. The prescaler generates seven clock rates from the internal bus clock. The prescaler select bits, PS[2:0], in the TIM status and control register (TSC) select the TIM clock source.

10.5.2 Input Capture

With the input capture function, the TIM can capture the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the TIM latches the contents of the TIM counter into the TIM channel registers, TCHxH:TCHxL. The polarity of the active edge is programmable. Input captures can generate TIM CPU interrupt requests.

10.5.3 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.

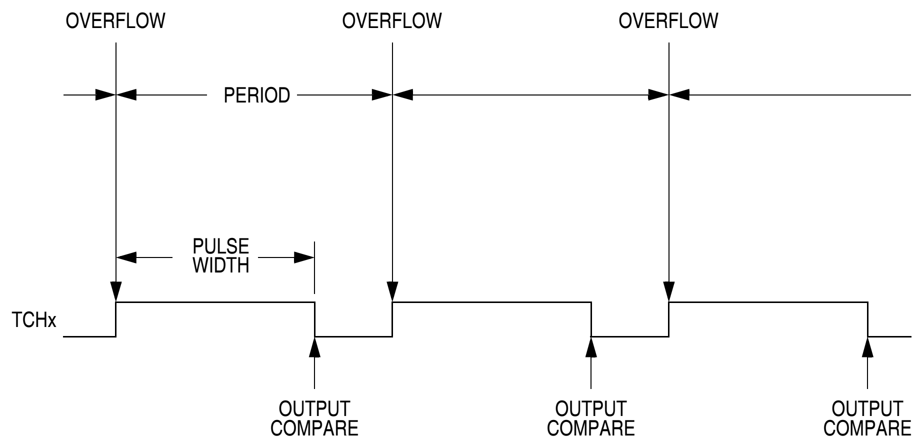


Figure 10-3. PWM Period and Pulse Width

The value in the TIM counter modulo registers and the selected prescaler output determines the frequency of the PWM output. The frequency of an 8-bit PWM signal is variable in 256 increments. Writing $00FF (255) to the TIM counter modulo registers produces a PWM period of 256 times the internal bus clock period if the prescaler select value is 000 (see **10.10.1 TIM Status and Control Register (TSC)**).

The value in the TIM channel registers determines the pulse width of the PWM output. The pulse width of an 8-bit PWM signal is variable in 256 increments. Writing $0080 (128) to the TIM channel registers produces a duty cycle of 128/256 or 50%.

10.5.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in **10.5.4 Pulse Width Modulation (PWM)**. The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change a pulse width value could cause incorrect operation for up to two PWM 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 PWM period. Also, using a TIM overflow interrupt routine to

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

10.8 TIM During Break Interrupts

A break interrupt stops the TIM counter.

The system integration module (SIM) controls whether status bits in other modules 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. (See **7.8.3 Break Flag Control Register (BFCR)**.)

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

To protect status bits during the break state, write a logic zero to the BCFE bit. With BCFE at logic zero (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a two-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is at logic zero. After the break, doing the second step clears the status bit.

10.9 I/O Signals

Port D shares two of its pins with the TIM. The two TIM channel I/O pins are PTD4/TCH0 and PTD5/TCH1.

Each channel I/O pin is programmable independently as an input capture pin or an output compare pin. PTD4/TCH0 can be configured as a buffered output compare or buffered PWM pin.

12.5 Port B

Port B is an 8-bit special function port that shares all eight of its port pins with the Analog-to-Digital converter (ADC) module, **See** **Section 11.**

12.5.1 Port B Data Register (PTB)

The port B data register contains a data latch for each of the eight port B 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							
Alternative Function:	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC2	ADC0

Figure 12-6. Port B Data Register (PTB)

PTB[7:0] — 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.

12.5.2 Data Direction Register B (DDRB)

Data direction register B determines whether each port B pin is an input or an output. Writing a logic one to a DDRB bit enables the output buffer for the corresponding port B pin; a logic zero 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)

12.6.2 Data Direction Register D (DDRD)

Data direction register D determines whether each port D pin is an input or an output. Writing a logic one to a DDRD bit enables the output buffer for the corresponding port D pin; a logic zero disables the output buffer.

Address: $0007

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Reset:	0	0	0	0	0	0	0	0

Figure 12-10. Data Direction Register D (DDRD)

DDRD[7:0] — Data Direction Register D Bits

These read/write bits control port D data direction. Reset clears DDRD[7:0], configuring all port D pins as inputs.

1 = Corresponding port D pin configured as output

0 = Corresponding port D pin configured as input

NOTE: *Avoid glitches on port D pins by writing to the port D data register before changing data direction register D bits from 0 to 1.* ***Figure 12-11*** *shows the port D I/O logic.*

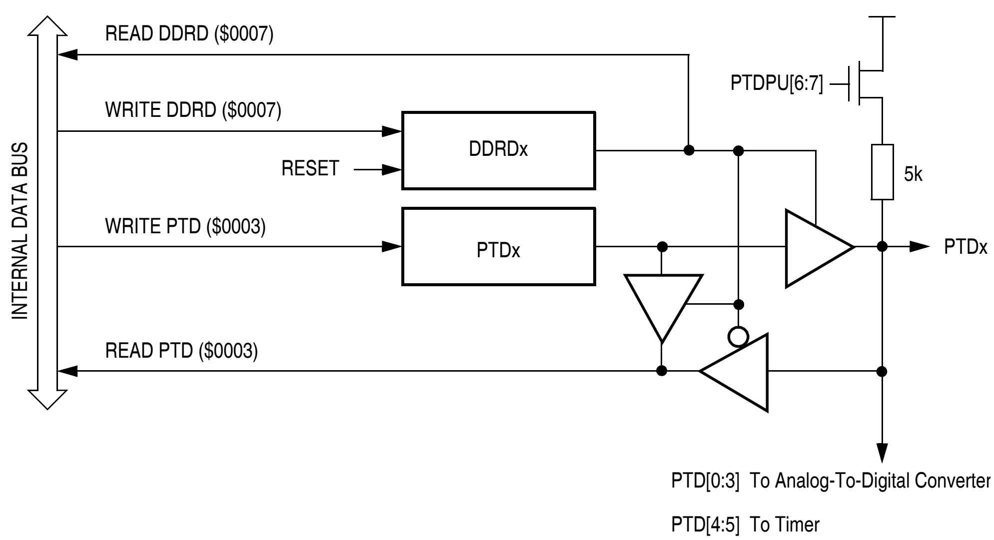


Figure 12-11. Port D I/O Circuit

level-triggered interrupt pin must be acknowledged after a delay that depends on the external load.

Another way to avoid a false interrupt:

1. Configure the keyboard pins as outputs by setting the appropriate DDRA bits in the data direction register A.
2. Write logic 1s to the appropriate port A data register bits.
3. Enable the KBI pins by setting the appropriate KBIEx bits in the keyboard interrupt enable register.

14.4.2 Keyboard Status and Control Register

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

Address: $001A

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

(shaded) = Unimplemented

Figure 14-3. Keyboard Status and Control Register (KBSCR)

Bits 7–4 — Not used

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

KEYF — Keyboard Flag Bit

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

1 = Keyboard interrupt pending
0 = No keyboard interrupt pending

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 16-3. Configuration Register 1 (CONFIG1)

LVID —þLow Voltage Inhibit Disable Bit

1 = Low voltage inhibit disabled

0 = Low voltage inhibit enabled

LVIT1, LVIT0 — LVI Trip Voltage Selection

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.

LVIT1	**LVIT0**	**Trip Voltage[1]**	**Comments**
0	0	V_{LVR3} (2.4V)	For V_{DD}=3V operation
0	1	V_{LVR3} (2.4V)	For V_{DD}=3V operation
1	0	V_{LVR5} (4.0V)	For V_{DD}=5V operation
1	1	Reserved	

1. See **Section 18. Electrical Specifications** for full parameters.

16.6 Low-Power Modes

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

16.6.1 Wait Mode

The LVI module, when enabled, will continue to operate in WAIT Mode.

16.6.2 Stop Mode

The LVI module, when enabled, will continue to operate in STOP Mode.

17.3 Features

Features of the break module include the following:

- Accessible I/O registers during the break Interrupt
- CPU-generated break interrupts
- Software-generated break interrupts
- COP disabling during break interrupts

17.4 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal ($\overline{\text{BKPT}}$) to the SIM. The SIM then causes the CPU to load the instruction register with a software interrupt instruction (SWI) after completion of the current CPU instruction. 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 logic one 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 begins after the CPU completes its current instruction. A return from interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation. **Figure 17-1** shows the structure of the break module.

17.4.1 Flag Protection During Break Interrupts

The system integration module (SIM) controls whether or not module status bits 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. (See **7.8.3 Break Flag Control Register (BFCR)** and see the Break Interrupts subsection for each module.)

17.4.2 CPU During Break Interrupts

The CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with \$FFFC:\$FFFD (\$FEFC:\$FEFD in monitor mode)

The break interrupt begins after completion of the CPU instruction in progress. If the break address register match occurs on the last cycle of a CPU instruction, the break interrupt begins immediately.

17.4.3 TIM During Break Interrupts

A break interrupt stops the timer counter.

17.4.4 COP During Break Interrupts

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

17.5 Break Module Registers

These registers control and monitor operation of the break module:

- Break status and control register (BRKSCR)
- Break address register high (BRKH)
- Break address register low (BRKL)
- Break status register (BSR)
- Break flag control register (BFCR)

Section 20. Ordering Information

20.1 Contents

20.2 Introduction

This section contains ordering numbers for the MC68H(R)C908JL3, MC68H(R)C908JK3, and MC68H(R)C908JK1.