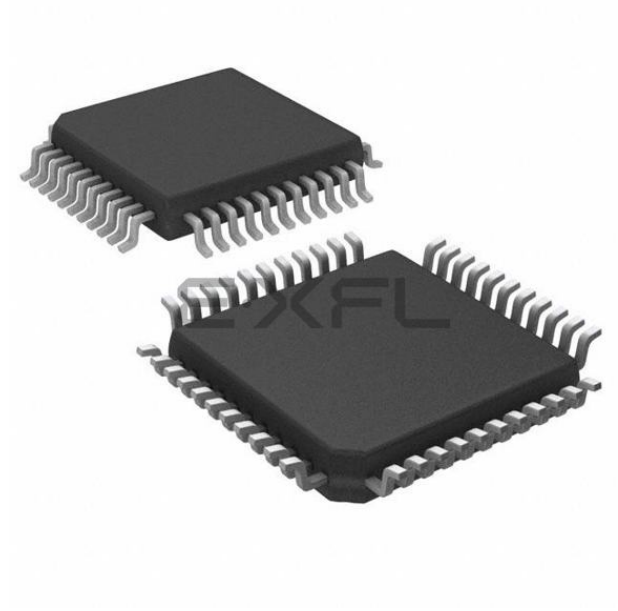


Welcome to E-XFL.COM

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	SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	36
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 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-QFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908gt16cfb

Chapter 19
Development Support

Chapter 20
Electrical Specifications

- System protection features:
  - Optional computer operating properly (COP) reset
  - Low-voltage detection with optional reset and selectable trip points for 3.0-V and 5.0-V operation
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- Low-power design; fully static with stop and wait modes
- Standard low-power modes of operation:
  - Wait mode
  - Stop mode
- Master reset pin and power-on reset (POR)
- 16 Kbytes of on-chip 100k cycle write/erase capable Flash memory (8 Kbytes on MC68HC908GT8)
- 512 bytes of on-chip random-access memory (RAM)
- 720 bytes of Flash programming routines ROM
- Serial peripheral interface module (SPI)
- Serial communications interface module (SCI)
- Two 16-bit, 2-channel timer interface modules (TIM1 and TIM2) with selectable input capture, output compare, and pulse-width modulation (PWM) capability on each channel
- 8-channel, 8-bit successive approximation analog-to-digital converter (ADC)
- Break module (BRK) to allow single breakpoint setting during in-circuit debugging
- Internal pullups on $\overline{\text{IRQ}}$ and $\overline{\text{RST}}$ to reduce customer system cost
- Up to 36 general-purpose input/output (I/O) pins, including:
  - 28 shared-function I/O pins
  - Six or eight dedicated I/O pins, depending on package choice
- Selectable pullups on inputs only on ports A, C, and D. Selection is on an individual port bit basis. During output mode, pullups are disengaged.
- High current 10-mA sink/10-mA source capability on all port pins
- Higher current 20-mA sink/source capability on PTC0–PTC4
- Timebase module with clock prescaler circuitry for eight user selectable periodic real-time interrupts with optional active clock source during stop mode for periodic wakeup from stop using an external 32-kHz crystal or internal oscillator
- User selection of having the oscillator enabled or disabled during stop mode
- 8-bit keyboard wakeup port
- Available packages:
  - 42-pin shrink dual in-line package (SDIP)
  - 44-pin quad flat pack (QFP)
- Specific features of the MC68HC908GT16 in 42-pin SDIP are:
  - Port C is only 5 bits: PTC0–PTC4
  - Port D is 8 bits: PTD0–PTD7; dual 2-channel TIM modules
- Specific features of the MC68HC908GT16 in 44-pin QFP are:
  - Port C is 7 bits: PTC0–PTC6
  - Port D is 8 bits: PTD0–PTD7; dual 2-channel TIM modules

2.5 Random-Access Memory (RAM)

Addresses $0040 through $023F are RAM locations. The location of the stack RAM is programmable. The 16-bit stack pointer allows the stack to be anywhere in the 64-Kbyte memory space.

NOTE

For correct operation, the stack pointer must point only to RAM locations.

Within page zero are 192 bytes of RAM. Because the location of the stack RAM is programmable, all page zero RAM locations can be used for I/O control and user data or code. When the stack pointer is moved from its reset location at $00FF out of page zero, direct addressing mode instructions can efficiently access all page zero RAM locations. Page zero RAM, therefore, provides ideal locations for frequently accessed global variables.

Before processing an interrupt, the CPU uses five bytes of the stack to save the contents of the CPU registers.

NOTE

For M6805 compatibility, the H register is not stacked.

During a subroutine call, the CPU uses two bytes of the stack to store the return address. The stack pointer decrements during pushes and increments during pulls.

NOTE

Be careful when using nested subroutines. The CPU may overwrite data in the RAM during a subroutine or during the interrupt stacking operation.

2.6 Flash Memory

This sub-section describes the operation of the embedded Flash memory. This memory can be read, programmed, and erased from a single external supply. The program, erase, and read operations are enabled through the use of an internal charge pump.

2.6.1 Functional Description

The Flash memory is an array of 15,872 bytes (7,680 bytes on MC68HC908GT8) with an additional 36 bytes of user vectors, one byte of block protection and two bytes of ICG user trim storage. *An erased bit reads as 1 and a programmed bit reads as a 0.* Memory in the Flash array is organized into two rows per page basis. The page size is 64 bytes per page and the row size is 32 bytes per row. Hence the minimum erase page size is 64 bytes and the minimum program row size is 32 bytes. Program and erase operation operations are facilitated through control bits in Flash control register (FLCR). Details for these operations appear later in this section.

The address ranges for the user memory and vectors are:

- $C000–$FDFF; user memory ($E000–$FDFF on MC68HC908GT8)
- $FE08; Flash control register
- $FF7E; Flash block protect register
- $FF80; ICG user trim register (ICGTR5)
- $FF81; ICG user trim register (ICGTR3)
- $FFDC–$FFFF; these locations are reserved for user-defined interrupt and reset vectors

Chapter 6
Central Processor Unit (CPU)

6.1 Introduction

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

6.2 Features

Features of the CPU include:

- Object code fully upward-compatible with M68HC05 Family
- 16-bit stack pointer with stack manipulation instructions
- 16-bit index register with x-register manipulation instructions
- 8-MHz CPU internal bus frequency
- 64-Kbyte program/data memory space
- 16 addressing modes
- Memory-to-memory data moves without using accumulator
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- Enhanced binary-coded decimal (BCD) data handling
- Modular architecture with expandable internal bus definition for extension of addressing range beyond 64 Kbytes
- Low-power stop and wait modes

6.3 CPU Registers

Figure 6-1 shows the five CPU registers. CPU registers are not part of the memory map.

7.3.2.3 Frequency Comparator

The frequency comparator effectively compares the low-frequency base clock (IBASE) to a nominal frequency, f_{NOM}. First, the frequency comparator converts IBASE to a voltage by charging a known capacitor with a current reference for a period dependent on IBASE. This voltage is compared to a voltage reference with comparators, whose outputs are fed to the digital loop filter. The dependence of these outputs on the capacitor size, current reference, and voltage reference causes up to ±25 percent error in f_{NOM}.

7.3.2.4 Digital Loop Filter

The digital loop filter (DLF) uses the outputs of the frequency comparator to adjust the internal clock (ICLK) clock period. The DLF generates the DCO divider control bits (DDIV[3:0]) and the DCO stage control bits (DSTG[7:0]), which are fed to the DCO. The DLF first concatenates the DDIV and DSTG registers (DDIV[3:0]:DSTG[7:0]) and then adds or subtracts a value dependent on the relative error in the low-frequency base clock's period, as shown in Table 7-1. In some extreme error conditions, such as operating at a V_{DD} level which is out of specification, the DLF may attempt to use a value above the maximum ($9FF) or below the minimum ($000). In both cases, the value for DDIV will be between $A and $F. In this range, the DDIV value will be interpreted the same as $9 (the slowest condition). Recovering from this condition requires subtracting (increasing frequency) in the normal fashion until the value is again below $9FF. (If the desired value is $9xx, the value may settle at $Axx through $Fxx. This is an acceptable operating condition.) If the error is less than ±5 percent, the internal clock generator's filter stable indicator (FICGS) is set, indicating relative frequency accuracy to the clock monitor.

Table 7-1. Correction Sizes from DLF to DCO

Frequency Error of IBASE Compared to f_{NOM}	DDVI[3:0]:DSTG[7:0] Correction	Current to New DDIV[3:0]:DSTG[7:0](1)		Relative Correction in DCO	
IBASE < 0.85 f_{NOM}	–32 (–$020)	Minimum	$xFF to $xDF	–2/31	–6.45%
		Maximum	$x20 to $x00	–2/19	–10.5%
0.85 f_{NOM} < IBASE IBASE < 0.95 f_{NOM}	–8 (–$008)	Minimum	$xFF to $xF7	–0.5/31	–1.61%
		Maximum	$x08 to $x00	–0.5/17.5	–2.86%
0.95 f_{NOM} < IBASE IBASE < f_{NOM}	–1 (–$001)	Minimum	$xFF to $xFE	–0.0625/31	–0.202%
		Maximum	$x01 to $x00	–0.0625/17.0625	–0.366%
f_{NOM} < IBASE IBASE < 1.05 f_{NOM}	+1 (+$001)	Minimum	$xFE to $xFF	+0.0625/30.9375	+0.202%
		Maximum	$x00 to $x01	+0.0625/17	+0.368%
1.05 f_{NOM} < IBASE IBASE < 1.15 f_{NOM}	+8 (+$008)	Minimum	$xF7 to $xFF	+0.5/30.5	+1.64%
		Maximum	$x00 to $x08	+0.5/17	+2.94%
1.15 f_{NOM} < IBASE	+32 (+$020)	Minimum	$xDF to $xFF	+2/29	+6.90%
		Maximum	$x00 to $x20	+2/17	+11.8%

1. x = Maximum error is independent of value in DDIV[3:0]. DDIV increments or decrements when an addition to DSTG[7:0] carries or borrows.

7.4.4.4 Ring Oscillator Fine-Adjust Circuit

The ring oscillator fine-adjust circuit causes the ring oscillator to effectively operate at non-integer numbers of stage delays by operating at two different points for a variable number of cycles specified by the lower five DCO stage control bits (DSTG[4:0]). For example:

- When DSTG[7:5] is %011, the ring oscillator nominally operates at 23 stage delays.
- When DSTG[4:0] is %00000, the ring will always operate at 23 stage delays.
- When DSTG[4:0] is %00001, the ring will operate at 25 stage delays for one of 32 cycles and at 23 stage delays for 31 of 32 cycles.
- Likewise, when DSTG[4:0] is %11111, the ring operates at 25 stage delays for 31 of 32 cycles and at 23 stage delays for one of 32 cycles.
- When DSTG[7:5] is %111, similar results are achieved by including a variable divide-by-two, so the ring operates at 31 stages for some cycles and at 17 stage delays, with a divide-by-two for an effective 34 stage delays, for the remainder of the cycles.

Adjusting the DSTG[0] bit has a 0.202 percent to 0.368 percent effect on the output clock period. This corresponds to the minimum size correction made by the DLF, and the inherent, long-term quantization error in the output frequency.

7.4.5 Switching Internal Clock Frequencies

The frequency of the internal clock (ICLK) may need to be changed for some applications. For example, if the reset condition does not provide the correct frequency, or if the clock is slowed down for a low-power mode (or sped up after a low-power mode), the frequency must be changed by programming the internal clock multiplier factor (N). The frequency of ICLK is N times the frequency of IBASE, which is 307.2 kHz ±25 percent.

Before switching frequencies by changing the N value, the clock monitor must be disabled. This is because when N is changed, the frequency of the low-frequency base clock (IBASE) will change proportionally until the digital loop filter has corrected the error. Since the clock monitor uses IBASE, it could erroneously detect an inactive clock. The clock monitor cannot be re-enabled until the internal clock is stable again (ICGS is set).

The following flow is an example of how to change the clock frequency:

- Verify there is no clock monitor interrupt by reading the CMF bit.
- Turn off the clock monitor.
- If desired, switch to the external clock (see 7.4.1 Switching Clock Sources).
- Change the value of N.
- Switch back to internal (see 7.4.1 Switching Clock Sources), if desired.
- Turn on the clock monitor (see 7.4.2 Enabling the Clock Monitor), if desired.

7.4.6 Nominal Frequency Settling Time

Because the clock period of the internal clock (ICLK) is dependent on the digital loop filter outputs (DDIV and DSTG) which cannot change instantaneously, ICLK temporarily will operate at an incorrect clock period when any operating condition changes. This happens whenever the part is reset, the ICG multiply factor (N) is changed, the ICG trim factor (TRIM) is changed, or the internal clock is enabled after inactivity (stop mode or disabled operation). The time that the ICLK takes to adjust to the correct period is known as the settling time.

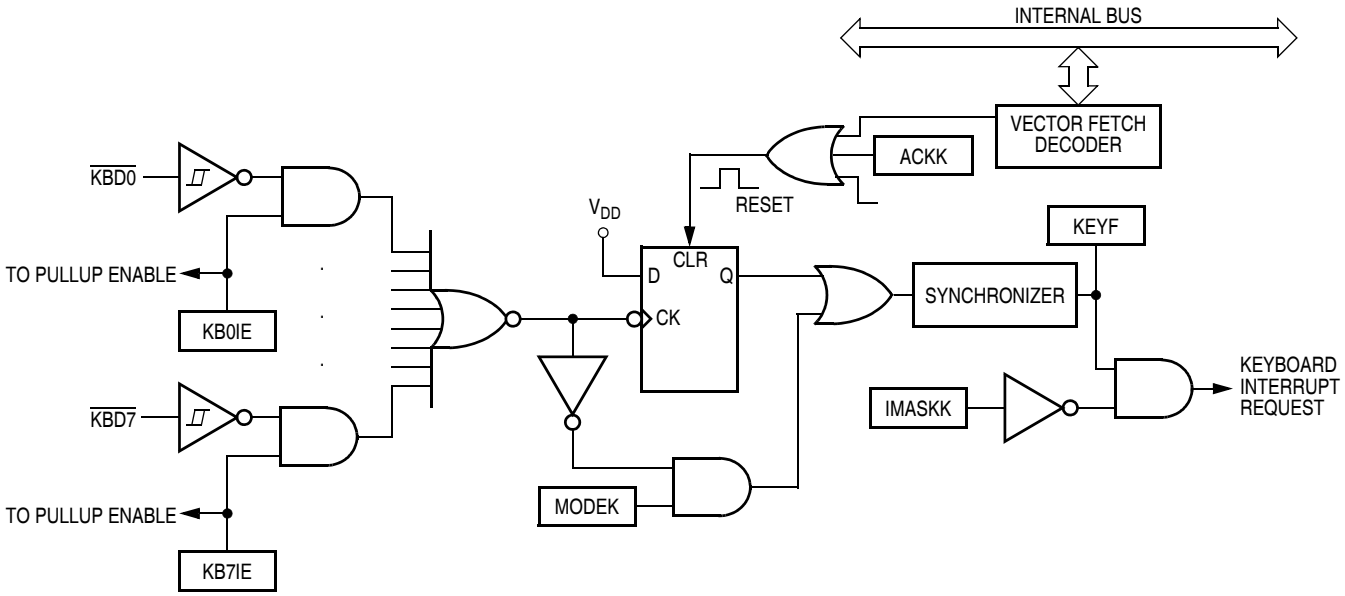


Figure 9-2. Keyboard Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001A	Keyboard Status and Control Register (INTKBSCR) See page 111.	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$001B	Keyboard Interrupt Enable Register (INTKBIER) See page 112.	Read:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 9-3. I/O Register Summary

If the MODEK bit is set, the keyboard interrupt pins are both falling edge- and low-level sensitive, and both of the following actions must occur to clear a keyboard interrupt request:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the interrupt request. Software may generate the interrupt acknowledge signal by writing a 1 to the ACKK bit in the keyboard status and control register (INTKBSCR). The ACKK bit is useful in applications that poll the keyboard interrupt pins and require software to clear the keyboard interrupt request. Writing to the ACKK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACKK does not affect subsequent transitions on the keyboard interrupt pins. A falling edge that occurs after writing to the ACKK bit latches another interrupt request. If the keyboard interrupt mask bit, IMASKK, is clear, the CPU loads the program counter with the vector address at locations $FFE0 and $FFE1.
- Return of all enabled keyboard interrupt pins to logic 1 — As long as any enabled keyboard interrupt pin is at logic 0, the keyboard interrupt remains set.

12.2 Port A

Port A is an 8-bit special-function port that shares all eight of its pins with the keyboard interrupt (KBI) module. Port A also has software configurable pullup devices if configured as an input port.

12.2.1 Port A Data Register

The port A data register (PTA) contains a data latch for each of the eight port A pins.

Address: $0000

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
Reset:	Unaffected by reset							
Alternative Function:	KBD7	KBD6	KBD5	KBD4	KBD3	KBD2	KBD1	KBD0

Figure 12-2. Port A Data Register (PTA)

PTA7–PTA0 — Port A Data Bits

These read/write bits are software programmable. Data direction of each port A pin is under the control of the corresponding bit in data direction register A. Reset has no effect on port A data.

KBD7–KBD0 — Keyboard Inputs

The keyboard interrupt enable bits, KBIE7–KBIE0, in the keyboard interrupt control register (KBICR) enable the port A pins as external interrupt pins. See Chapter 9 Keyboard Interrupt Module (KBI).

12.2.2 Data Direction Register A

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

Address: $0004

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset:	0	0	0	0	0	0	0	0

Figure 12-3. Data Direction Register A (DDRA)

DDRA7–DDRA0 — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA7–DDRA0, configuring all port A pins as inputs.

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

NOTE

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

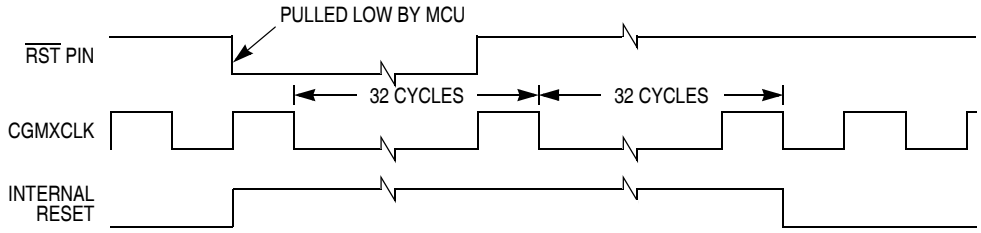


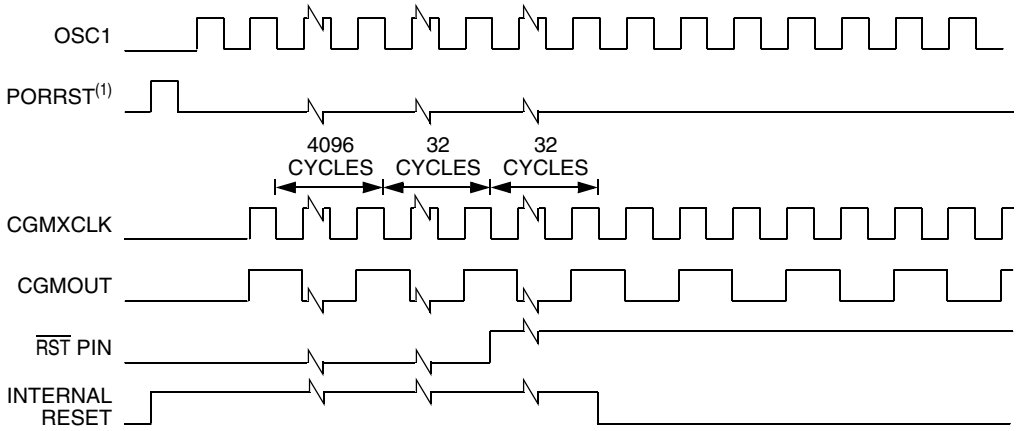
Figure 13-1. Internal Reset Timing

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 completely to 0 V 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.

A power-on reset:

- Holds the clocks to the CPU and modules inactive for an oscillator stabilization delay of 4096 CGMXCLK cycles
- Drives the $\overline{RST}$ pin low during the oscillator stabilization delay
- Releases the RST pin 32 CGMXCLK cycles after the oscillator stabilization delay
- Releases the CPU to begin the reset vector sequence 64 CGMXCLK cycles after the oscillator stabilization delay
- Sets the POR bit in the SIM reset status register and clears all other bits in the register



1. PORRST is an internally generated power-on reset pulse.

Figure 13-2. Power-On Reset Recovery

13.2.3.2 Computer Operating Properly (COP) Reset

A COP reset is an internal reset caused by an overflow of the COP counter. A COP reset sets the COP bit in the system integration module (SIM) reset status register.

To clear the COP counter and prevent a COP reset, write any value to the COP control register at location $FFFF.

13.2.3.3 Low-Voltage Inhibit Reset

A low-voltage inhibit (LVI) reset is an internal reset caused by a drop in the power supply voltage to the LVI_{TRIPF} voltage.

An LVI reset:

- Holds the clocks to the CPU and modules inactive for an oscillator stabilization delay of 4096 CGMXCLK cycles after the power supply voltage rises to the LVI_{TRIPR} voltage
- Drives the $\overline{RST}$ pin low for as long as V_{DD} is below the LVI_{TRIPR} voltage and during the oscillator stabilization delay
- Releases the $\overline{RST}$ pin 32 CGMXCLK cycles after the oscillator stabilization delay
- Releases the CPU to begin the reset vector sequence 64 CGMXCLK cycles after the oscillator stabilization delay
- Sets the LVI bit in the SIM reset status register

13.2.3.4 Illegal Opcode Reset

An illegal opcode reset is an internal reset caused by an opcode that is not in the instruction set. An illegal opcode reset sets the ILOP bit in the SIM reset status register.

If the stop enable bit, STOP, in the CONFIG1 register is a 0, the STOP instruction causes an illegal opcode reset.

13.2.3.5 Illegal Address Reset

An illegal address reset is an internal reset caused by opcode fetch from an unmapped address. An illegal address reset sets the ILAD bit in the SIM reset status register.

A data fetch from an unmapped address does not generate a reset.

13.2.4 SIM Reset Status Register

This read-only register contains flags to show reset sources. All flag bits are automatically cleared following a read of the register. Reset service can read the SIM reset status register to clear the register after power-on reset and to determine the source of any subsequent reset.

The register is initialized on power-up as shown with the POR bit set and all other bits cleared. During a POR or any other internal reset, the $\overline{RST}$ pin is pulled low. After the pin is released, it will be sampled 32 CGMXCLK cycles later. If the pin is not above a V_{IH} at that time, then the PIN bit in the SRSR may be set in addition to whatever other bits are set.

NOTE

Only a read of the SIM reset status register clears all reset flags. After multiple resets from different sources without reading the register, multiple flags remain set.

14.4.3.3 Data Sampling

The receiver samples the RxD pin at the RT clock rate. The RT clock is an internal signal with a frequency 16 times the baud rate. To adjust for baud rate mismatch, the RT clock is resynchronized at these times (see Figure 14-7):

- After every start bit
- After the receiver detects a data bit change from 1 to 0 (after the majority of data bit samples at RT8, RT9, and RT10 returns a valid 1 and the majority of the next RT8, RT9, and RT10 samples returns a valid 0)

To locate the start bit, data recovery logic does an asynchronous search for a 0 preceded by three 1s. When the falling edge of a possible start bit occurs, the RT clock begins to count to 16.

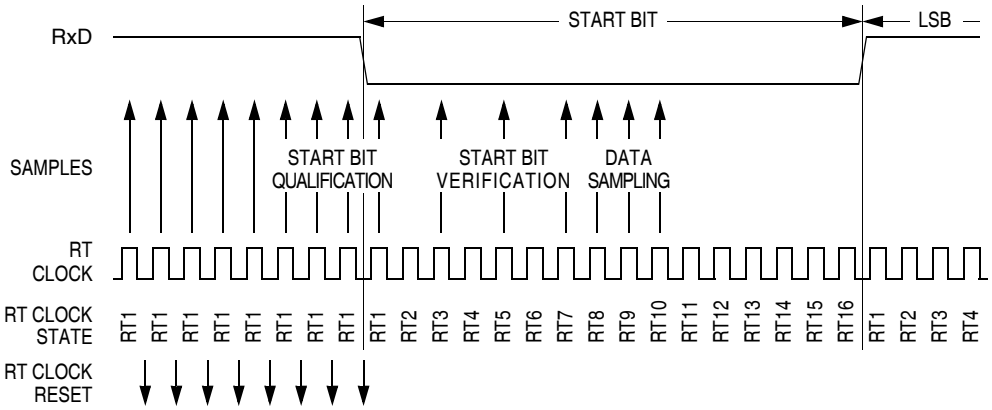


Figure 14-7. Receiver Data Sampling

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7. Table 14-2 summarizes the results of the start bit verification samples.

Table 14-2. Start Bit Verification

RT3, RT5, and RT7 Samples	Start Bit Verification	Noise Flag
000	Yes	0
001	Yes	1
010	Yes	1
011	No	0
100	Yes	1
101	No	0
110	No	0
111	No	0

If start bit verification is not successful, the RT clock is reset and a new search for a start bit begins.

To determine the value of a data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. Table 14-3 summarizes the results of the data bit samples.

Slow Data Tolerance

Figure 14-8 shows how much a slow received character can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.

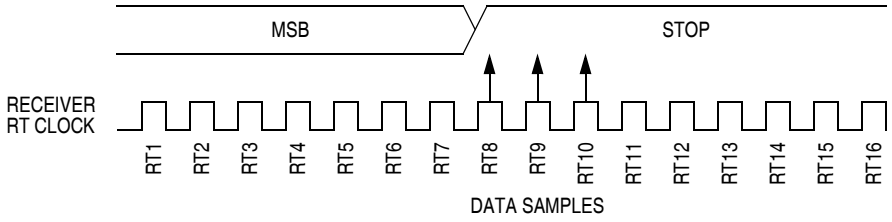


Figure 14-8. Slow Data

For an 8-bit character, data sampling of the stop bit takes the receiver 9 bit times × 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in Figure 14-8, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 9 bit times × 16 RT cycles + 3 RT cycles = 147 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit character with no errors is:

$$\left|\frac{154 - 147}{154}\right| \times 100 = 4.54\%$$

For a 9-bit character, data sampling of the stop bit takes the receiver 10 bit times × 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in Figure 14-8, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 10 bit times × 16 RT cycles + 3 RT cycles = 163 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is:

$$\left|\frac{170 - 163}{170}\right| \times 100 = 4.12\%$$

Fast Data Tolerance

Figure 14-9 shows how much a fast received character can be misaligned without causing a noise error or a framing error. The fast stop bit ends at RT10 instead of RT16 but is still there for the stop bit data samples at RT8, RT9, and RT10.

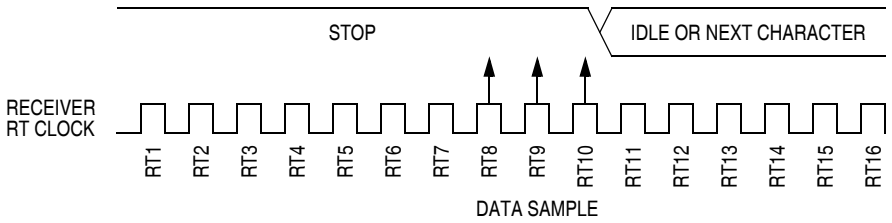


Figure 14-9. Fast Data

AROVFL— Arbiter Counter Overflow Bit

This read-only bit indicates an arbiter counter overflow. Clear AROVFL by writing any value to SCIACTL. Writing 0s to AM1 and AM0 resets the counter keeps it in this idle state. Reset clears AROVFL.

1 = Arbiter counter overflow has occurred
0 = No arbiter counter overflow has occurred

ARD8— Arbiter Counter MSB

This read-only bit is the MSB of the 9-bit arbiter counter. Clear ARD8 by writing any value to SCIACTL. Reset clears ARD8.

14.9.2 ESCI Arbiter Data Register

Address: $000B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ARD7	ARD6	ARD5	ARD4	ARD3	ARD2	ARD1	ARD0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-20. ESCI Arbiter Data Register (SCIADAT)

ARD7–ARD0 — Arbiter Least Significant Counter Bits

These read-only bits are the eight LSBs of the 9-bit arbiter counter. Clear ARD7–ARD0 by writing any value to SCIACTL. Writing 0s to AM1 and AM0 permanently resets the counter and keeps it in this idle state. Reset clears ARD7–ARD0.

14.9.3 Bit Time Measurement

Two bit time measurement modes, described here, are available according to the state of ACLK.

1. ACLK = 0 — The counter is clocked with one quarter of the bus clock. The counter is started when a falling edge on the RxD pin is detected. The counter will be stopped on the next falling edge. ARUN is set while the counter is running, AFIN is set on the second falling edge on RxD (for instance, the counter is stopped). This mode is used to recover the received baud rate. See Figure 14-21.
2. ACLK = 1 — The counter is clocked with one half of the ESCI input clock generated by the ESCI prescaler. The counter is started when a 0 is detected on RxD (see Figure 14-22). A 0 on RxD on enabling the bit time measurement with ACLK = 1 leads to immediate start of the counter (see Figure 14-23). The counter will be stopped on the next rising edge of RxD. This mode is used to measure the length of a received break.

14.9.4 Arbitration Mode

If AM[1:0] is set to 10, the arbiter module operates in arbitration mode. On every rising edge of SCI_TxD (output of the ESCI module, internal chip signal), the counter is started. When the counter reaches $38 (ACLK = 0) or $08 (ACLK = 1), RxD is statically sensed. If in this case, RxD is sensed low (for example, another bus is driving the bus dominant) ALOST is set. As long as ALOST is set, the TxD pin is forced to 1, resulting in a seized transmission.

If SCI_TxD is sensed 0 without having sensed a 0 before on RxD, the counter will be reset, arbitration operation will be restarted after the next rising edge of SCI_TxD.

Chapter 15
System Integration Module (SIM)

15.1 Introduction

This section describes the system integration module (SIM). Together with the central processor unit (CPU), the SIM controls all microconroller unit (MCU) activities. A block diagram of the SIM is shown in Figure 15-1. Table 15-1 is a summary of the SIM input/output (I/O) registers. The SIM is a system state controller that coordinates CPU and exception timing.

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.

Table 15-1. Signal Name Conventions

Signal Name	Description
CGMXCLK	Selected clock source from internal clock generator module (ICG)
CGMOUT	Clock output from ICG 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/$\overline{W}$	Read/write signal

15.3.2 Active Resets from Internal Sources

All internal reset sources actively pull the $\overline{\text{RST}}$ pin low for 32 CGMXCLK cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles. See Figure 15-5. An internal reset can be caused by an illegal address, illegal opcode, COP timeout, LVI, or POR. See Figure 15-6.

NOTE

For LVI or POR resets, the SIM cycles through 4096 CGMXCLK cycles during which the SIM forces the $\overline{\text{RST}}$ pin low. The internal reset signal then follows the sequence from the falling edge of $\overline{\text{RST}}$ shown in Figure 15-5.

The COP reset is asynchronous to the bus clock.

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

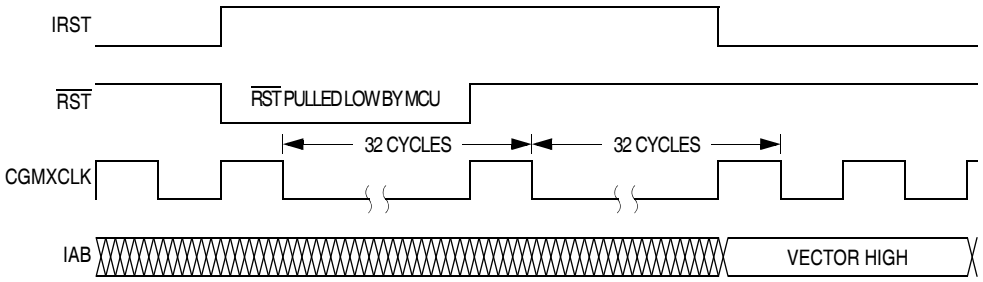


Figure 15-5. Internal Reset Timing

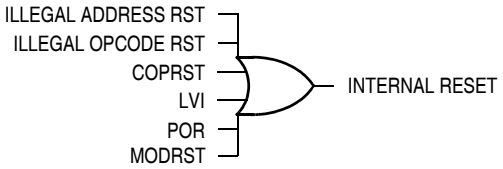


Figure 15-6. Sources of Internal Reset

Table 15-2. PIN Bit Set Timing

Reset Recovery Type	Actual Number of Cycles
POR/LVI	4163 (4096 + 64 + 3)
All others	67 (64 + 3)

15.3.2.1 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 + 32 CGMXCLK cycles. Thirty-two CGMXCLK cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur.

18.4.4 Pulse Width Modulation (PWM)

By using the toggle-on-overflow feature with an output compare channel, the TIM can generate a PWM signal. The value in the TIM counter modulo registers determines the period of the PWM signal. The channel pin toggles when the counter reaches the value in the TIM counter modulo registers. The time between overflows is the period of the PWM signal.

As Figure 18-4 shows, the output compare value in the TIM channel registers determines the pulse width of the PWM signal. The time between overflow and output compare is the pulse width. Program the TIM to clear the channel pin on output compare if the state of the PWM pulse is logic 1. Program the TIM to set the pin if the state of the PWM pulse is logic 0.

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 18.9.1 TIM Status and Control Register.

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%.

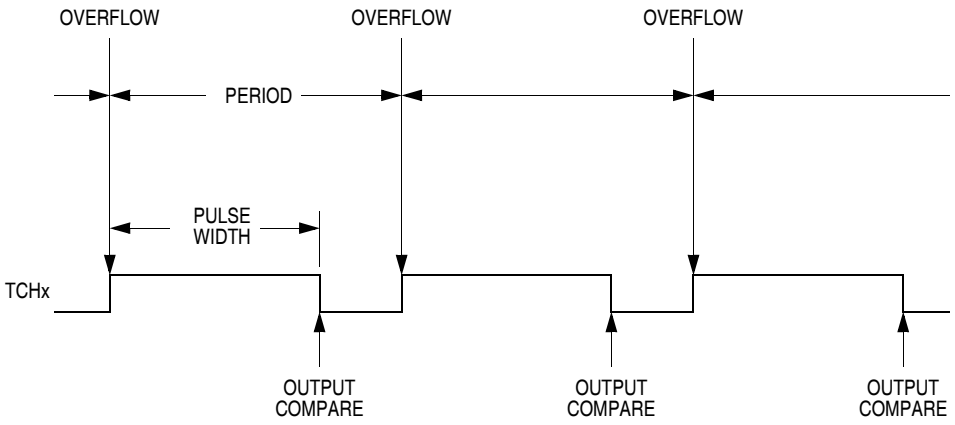


Figure 18-4. PWM Period and Pulse Width

18.4.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in 18.4.4 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 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 the internal address bus matches the value written in the break address registers or when software writes a 1 to the BRKA bit in the break status and control register, the CPU starts a break interrupt by:

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

The break interrupt timing is:

- When a break address is placed at the address of the instruction opcode, the instruction is not executed until after completion of the break interrupt routine.
- When a break address is placed at an address of an instruction operand, the instruction is executed before the break interrupt.
- When software writes a 1 to the BRKA bit, the break interrupt occurs just before the next instruction is executed.

By updating a break address and clearing the BRKA bit in a break interrupt routine, a break interrupt can be generated continuously.

CAUTION

A break address should be placed at the address of the instruction opcode. When software does not change the break address and clears the BRKA bit in the first break interrupt routine, the next break interrupt will not be generated after exiting the interrupt routine even when the internal address bus matches the value written in the break address registers.

19.2.1.1 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 15.7.3 SIM Break Flag Control Register and the **Break Interrupts** subsection for each module.

19.2.1.2 TIM1 and TIM2 During Break Interrupts

A break interrupt stops the timer counters.

19.2.1.3 COP During Break Interrupts

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

19.2.2 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)
- SIM break status register (SBSR)
- SIM break flag control register (SBFCR)

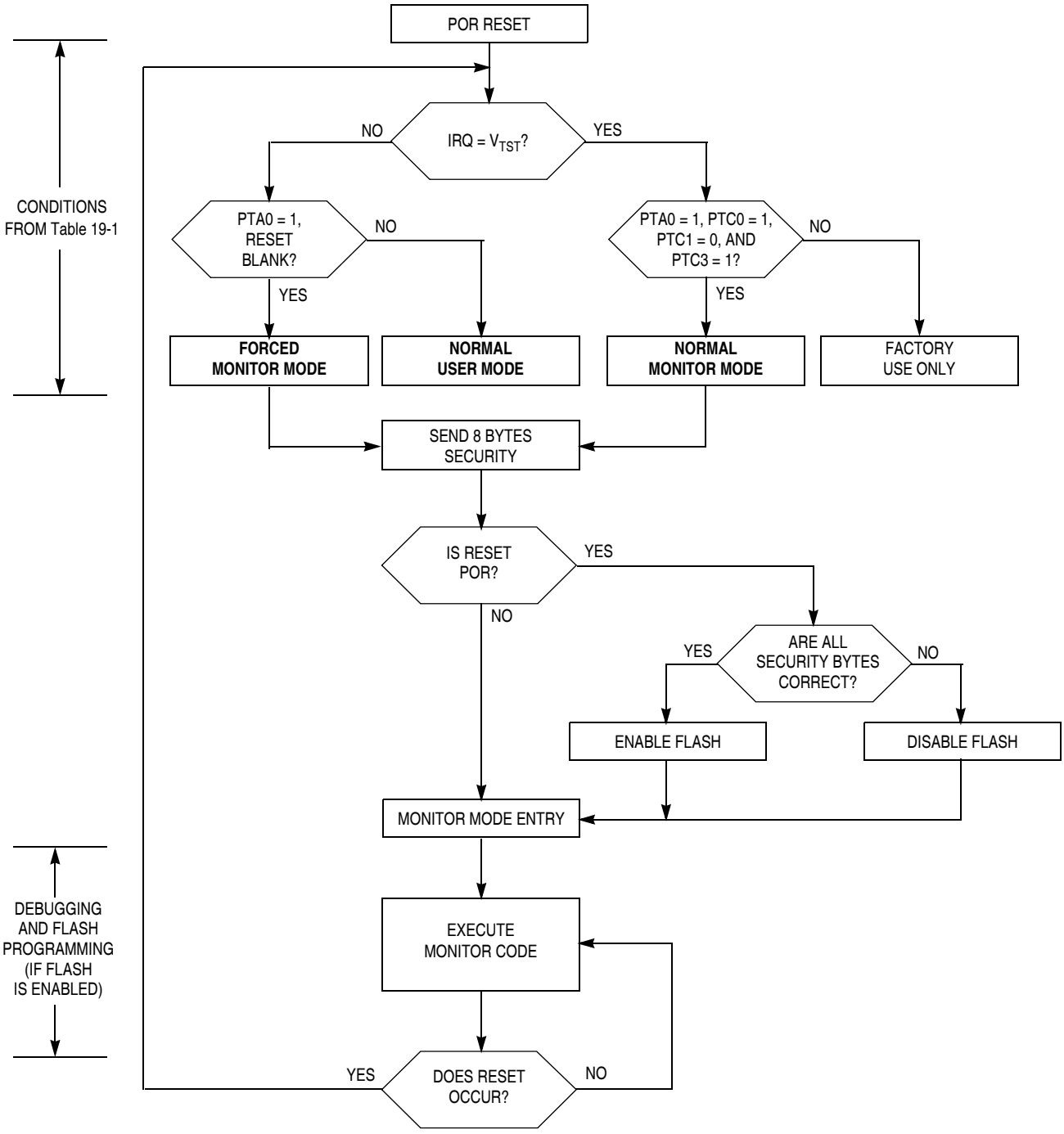
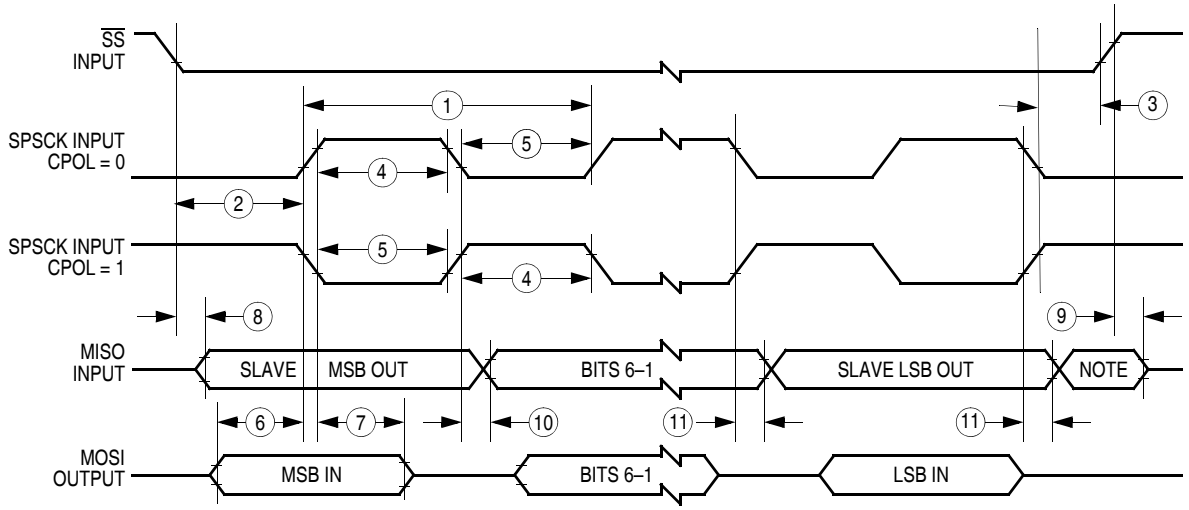


Figure 19-9. Simplified Monitor Mode Entry Flowchart

20.5 5.0-V DC Electrical Characteristics

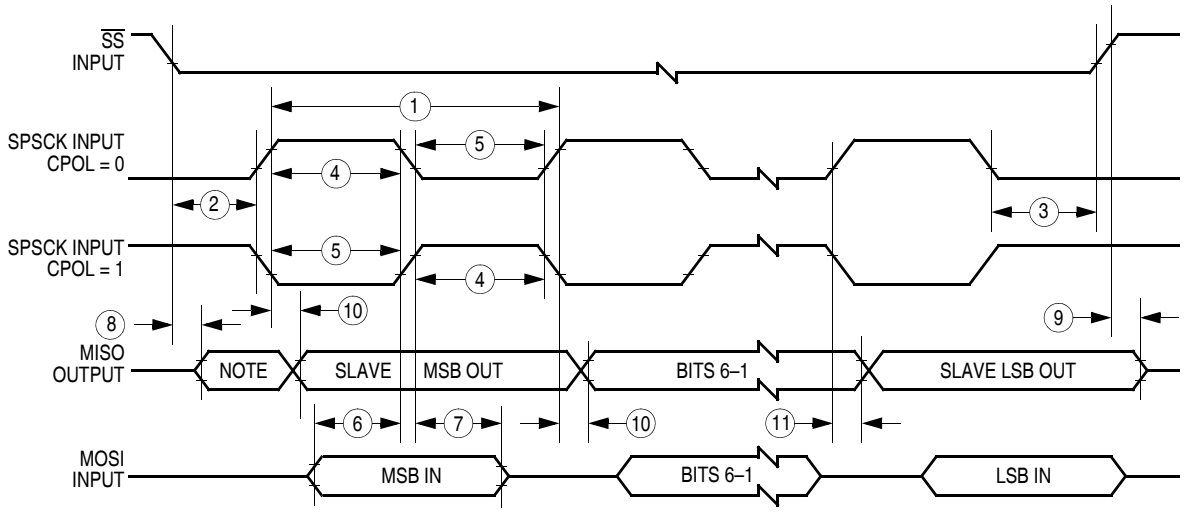
Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage (I_{Load} = –2.0 mA) all I/O pins (I_{Load} = –10.0 mA) all I/O pins (I_{Load} = –20.0 mA) pins PTC0–PTC4 only Maximum combined I_{OH} for port C, port E, port PTD0–PTD3 Maximum combined I_{OH} for port PTD4–PTD7, port A, port B Maximum total I_{OH} for all port pins	 V_{OH} V_{OH} V_{OH} I_{OH1} I_{OH2} I_{OHT}	 V_{DD} – 0.8 V_{DD} – 1.5 V_{DD} – 1.5 — — —	 — — — — — —	 — — — 50 50 100	 V V V mA mA mA
Output low voltage (I_{Load} = 1.6 mA) all I/O pins (I_{Load} = 10 mA) all I/O pins (I_{Load} = 20mA) pins PTC0–PTC4 only Maximum combined I_{OL} for port C, port E, port PTD0–PTD3 Maximum combined I_{OL} for port PTD4–PTD7, port A, port B Maximum total I_{OL} for all port pins	 V_{OL} V_{OL} V_{OL} I_{OL1} I_{OL2} I_{OLT}	 — — — — — —	 — — — — — —	 0.4 1.5 1.5 50 50 100	 V V V mA mA mA
Input high voltage All ports, $\overline{IRQ}$, $\overline{RST}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage All ports, $\overline{IRQ}$, $\overline{RST}$, OSC1	V_{IL}	V_{SS}	—	0.2 × V_{DD}	V
DC injection current, all ports[3]	I_{INJ}	– 2.0	—	+ 2.0	mA
Total DC current injection (sum of all I/O)[3]	I_{INJTOT}	– 25	—	+25	mA
I/O ports Hi-Z leakage current[4]	I_{IL}	—	—	±10	µA
Input current	I_{In}	—	—	±1	µA
Pullup resistors (as input only) Ports PTA7/$\overline{KBD7}$–PTA0/$\overline{KBD0}$, PTC6–PTC0, PTD7/T2CH1–PTD0/$\overline{SS}$	R_{PU}	20	45	65	kΩ
Capacitance Ports (as input or output)	C_{Out} C_{In}	— —	— —	12 8	pF
Monitor mode entry voltage	V_{TST}	V_{DD} + 2.5	—	V_{DD} + 4.0	V
Low-voltage inhibit, trip falling voltage	V_{TRIPF}	3.90	4.25	4.50	V
Low-voltage inhibit, trip rising voltage	V_{TRIPR}	4.20	4.35	4.60	V
Low-voltage inhibit reset/recover hysteresis (V_{TRIPF} + V_{HYS} = V_{TRIPR})	V_{HYS}	—	100	—	mV
POR rearm voltage[5]	V_{POR}	0	—	100	mV
POR reset voltage[6]	V_{PORRST}	0	700	800	mV
POR rise time ramp rate[7]	R_{POR}	0.035	—	—	V/ms

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_A (min) to T_A (max), unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range, 25°C only.
3. Some disturbance of the ADC accuracy is possible during any injection event and is dependent on board layout and power supply decoupling. This parameter is guaranteed by characterization.
4. Pullups and pulldowns are disabled. Port B leakage is specified in 20.16 ADC Characteristics.
5. Maximum is highest voltage that POR is guaranteed.
6. Maximum is highest voltage that POR is possible.
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.



Note: Not defined but normally MSB of character just received

a) SPI Slave Timing (CPHA = 0)



Note: Not defined but normally LSB of character previously transmitted

b) SPI Slave Timing (CPHA = 1)

Figure 20-18. SPI Slave Timing