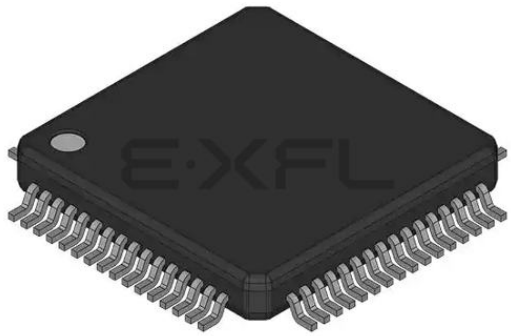


Welcome to E-XFL.COM

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	53
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908gr48acfue

$0000 ↓ $003F	I/O REGISTERS 64 BYTES
$0040 ↓ $043F	RAM-1 1024 BYTES
$0440 ↓ $0461	I/O REGISTERS 34 BYTES
$0462 ↓ $04FF	FLASH-2 158 BYTES
$0500 ↓ $057F	RESERVED 128 BYTES
$0580 ↓ $097F	RAM-2 1024 BYTES
$0980 ↓ $1B7F	FLASH-2 4608 BYTES
$1B80 ↓ $1DFF	RESERVED 640 BYTES
$1E00 ↓ $1E0F	MONITOR ROM 16 BYTES
$1E10 ↓ $1E1F	RESERVED 16 BYTES
$1E20 ↓ $7FFF	FLASH-2 25,056 BYTES
$8000 ↓ $FDFF	FLASH-1 32,256 BYTES

$FE00	SIM BREAK STATUS REGISTER (BSR)
$FE01	SIM RESET STATUS REGISTER (SRSR)
$FE02	RESERVED
$FE03	SIM BREAK FLAG CONTROL REGISTER (BFCR)
$FE04	INTERRUPT STATUS REGISTER 1 (INT1)
$FE05	INTERRUPT STATUS REGISTER 2 (INT2)
$FE06	INTERRUPT STATUS REGISTER 3 (INT3)
$FE07	INTERRUPT STATUS REGISTER 4 (INT4)
$FE08	FLASH-2 CONTROL REGISTER (FL2CR)
$FE09	BREAK ADDRESS REGISTER HIGH (BRKH)
$FE0A	BREAK ADDRESS REGISTER LOW (BRKL)
$FE0B	BREAK STATUS AND CONTROL REGISTER (BRKSCR)
$FE0C	LVI STATUS REGISTER (LVISR)
$FE0D	FLASH-2 TEST CONTROL REGISTER (FLTCR2)
$FE0E	FLASH-1 TEST CONTROL REGISTER (FLTCR1)
$FE0F	UNIMPLEMENTED
$FE10 ↓ $FE1F	UNIMPLEMENTED 16 BYTES RESERVED FOR COMPATIBILITY WITH MONITOR CODE FOR A-FAMILY PART
$FE20 ↓ $FF7F	MONITOR ROM 352 BYTES
$FF80	FLASH-1 BLOCK PROTECT REGISTER (FL1BPR)
$FF81	FLASH-2 BLOCK PROTECT REGISTER (FL2BPR)
$FF82 ↓ $FF87	RESERVED 6 BYTES
$FF88	FLASH-1 CONTROL REGISTER (FL1CR)
$FF89 ↓ $FFCB	RESERVED 67 BYTES
$FFCC ↓ $FFFF(1)	FLASH-1 VECTORS 52 BYTES

1. $FFF6–$FFFD used for eight security bytes

Figure 2-1. MC68HC908GR60A Memory Map

2.6.6 FLASH-1 Program Operation

Programming of the FLASH-1 memory is done on a row basis. A row consists of 64 consecutive bytes with address ranges as follows:

- $XX00 to $XX3F
- $XX40 to $XX7F
- $XX80 to $XXBF
- $XXC0 to $XXFF

During the programming cycle, make sure that all addresses being written to fit within one of the ranges specified above. Attempts to program addresses in different row ranges in one programming cycle will fail.

Use this step-by-step procedure to program a row of FLASH-1 memory.

NOTE

Only bytes which are currently $FF may be programmed.

1. Set the PGM bit in the FLASH-1 control register (FL1CR). This configures the memory for program operation and enables the latching of address and data programming.
2. Read the FLASH-1 block protect register (FL1BPR).
3. Write to any FLASH-1 address within the row address range desired with any data.
4. Wait for time, t_{NVS} (minimum 10 μs).
5. Set the HVEN bit.
6. Wait for time, t_{PGS} (minimum 5 μs).
7. Write data byte to the FLASH-1 address to be programmed.
8. Wait for time, t_{PROG} (minimum 30 μs).
9. Repeat steps 7 and 8 until all the bytes within the row are programmed.
10. Clear the PGM bit.
11. Wait for time, t_{NVH} (minimum 5 μs)
12. Clear the HVEN bit.
13. Wait for a time, t_{RCV}, (typically 1 μs) after which the memory can be accessed in normal read mode.

The FLASH programming algorithm flowchart is shown in Figure 2-6.

NOTE

A. *Programming and erasing of FLASH locations can not be performed by code being executed from the same FLASH array.*

B. *While these operations must be performed in the order shown, other unrelated operations may occur between the steps. However, care must be taken to ensure that these operations do not access any address within the FLASH array memory space such as the COP control register (COPCTL) at $FFFF.*

C. *It is highly recommended that interrupts be disabled during program/erase operations.*

D. *Do not exceed t_{PROG} maximum or t_{HV} maximum. t_{HV} is defined as the cumulative high voltage programming time to the same row before next erase. t_{HV} must satisfy this condition:*

$$t_{NVS} + t_{NVH} + t_{PGS} + (t_{PROG} \times 64) \leq t_{HV} \text{ maximum}$$

NOTE

Quantization error is affected when only the most significant eight bits are used as a result. See Figure 3-3.

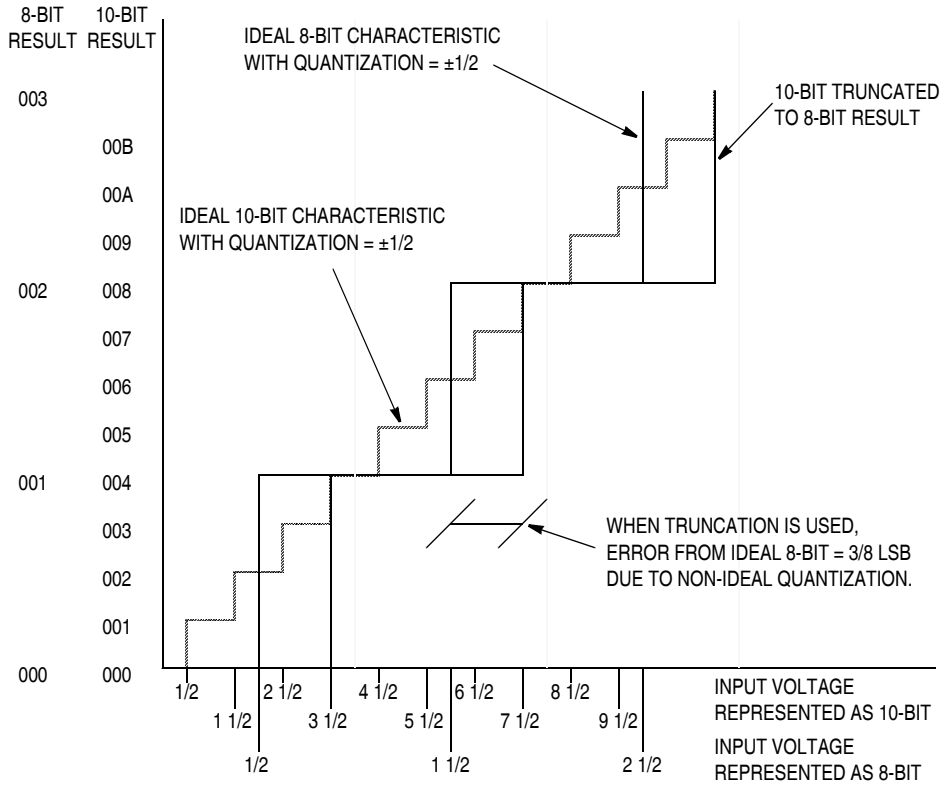


Figure 3-3. Bit Truncation Mode Error

3.4 Monotonicity

The conversion process is monotonic and has no missing codes.

3.5 Interrupts

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

3.6 Low-Power Modes

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

3.6.1 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

3.7.4 ADC Voltage Reference Low Pin (V_{REFL})

The ADC analog portion uses V_{REFL} as its lower voltage reference pin. By default, connect the V_{REFL} pin to the same voltage potential as V_{SS}. External filtering is often necessary to ensure a clean V_{REFL} for good results. Any noise present on this pin will be reflected and possibly magnified in A/D conversion values.

NOTE

For maximum noise immunity, route V_{REFL} carefully and, if not connected to V_{SS}, place bypass capacitors as close as possible to the package. Routing V_{REFH} close and parallel to V_{REFL} may improve common mode noise rejection.

V_{SSAD} and V_{REFL} are bonded internally.

3.7.5 ADC Voltage In (V_{ADIN})

V_{ADIN} is the input voltage signal from one of the 24 ADC channels to the ADC module.

3.8 I/O Registers

These I/O registers control and monitor ADC operation:

- ADC status and control register (ADSCR)
- ADC data register (ADRH and ADRL)
- ADC clock register (ADCLK)

3.8.1 ADC Status and Control Register

Function of the ADC status and control register (ADSCR) is described here.

Address: $003C

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

R = Reserved

Figure 3-4. ADC Status and Control Register (ADSCR)

COCO — Conversions Complete Bit

In non-interrupt mode (AIEN = 0), COCO is a read-only bit that is set at the end of each conversion. COCO will stay set until cleared by a read of the ADC data register. Reset clears this bit.

In interrupt mode (AIEN = 1), COCO is a read-only bit that is not set at the end of a conversion. It always reads as a 0.

1 = Conversion completed (AIEN = 0)

0 = Conversion not completed (AIEN = 0) or CPU interrupt enabled (AIEN = 1)

NOTE

The write function of the COCO bit is reserved. When writing to the ADSCR register, always have a 0 in the COCO bit position.

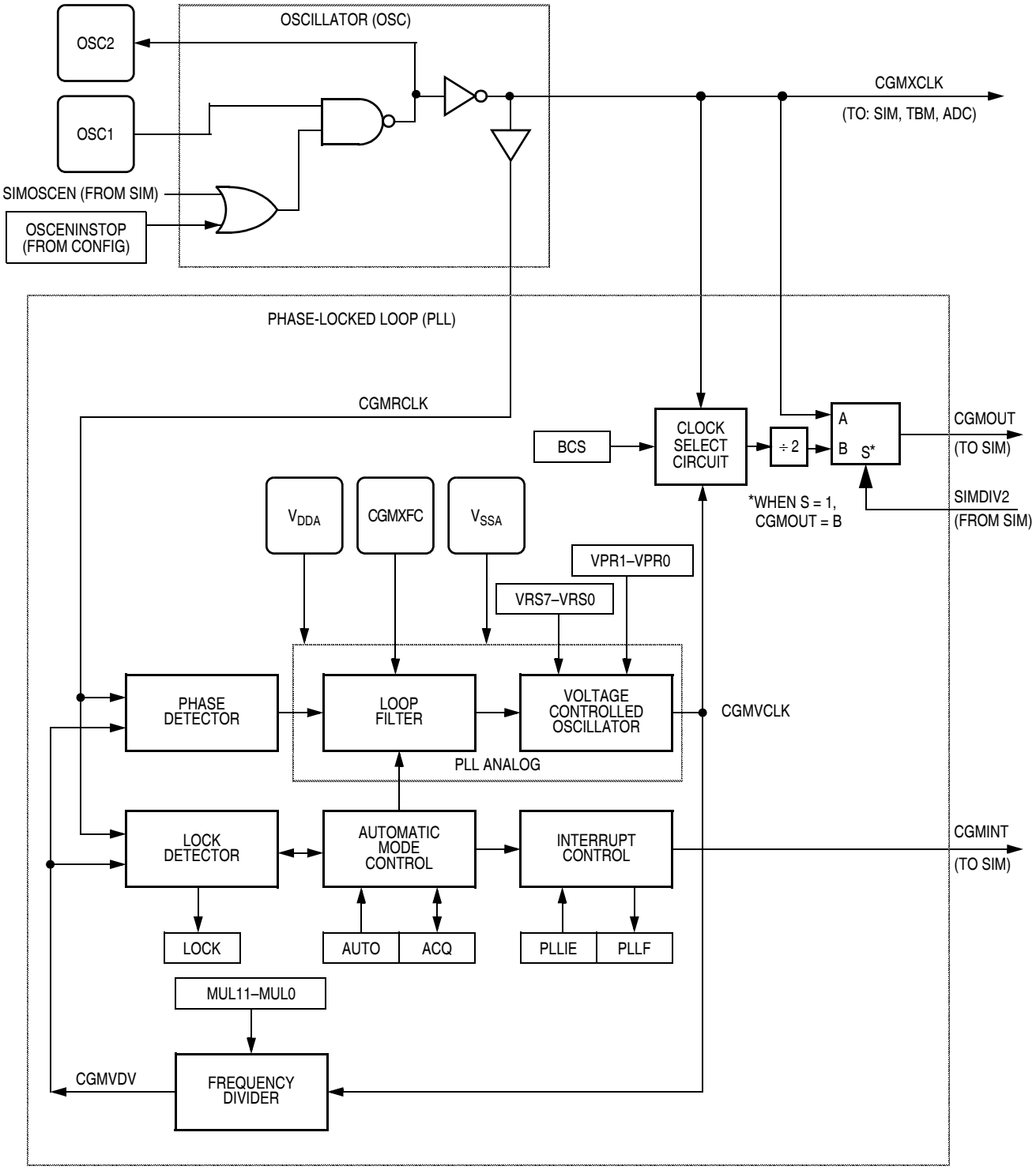


Figure 4-1. CGM Block Diagram

LOCK — Lock Indicator Bit

When the AUTO bit is set, LOCK is a read-only bit that becomes set when the VCO clock, CGMVCLK, is locked (running at the programmed frequency). When the AUTO bit is clear, LOCK reads as 0 and has no meaning. The write one function of this bit is reserved for test, so this bit must ***always*** be written as a 0. Reset clears the LOCK bit.

1 = VCO frequency correct or locked
0 = VCO frequency incorrect or unlocked

$\overline{\text{ACQ}}$ — Acquisition Mode Bit

When the AUTO bit is set, $\overline{\text{ACQ}}$ is a read-only bit that indicates whether the PLL is in acquisition mode or tracking mode. When the AUTO bit is clear, $\overline{\text{ACQ}}$ is a read/write bit that controls whether the PLL is in acquisition or tracking mode.

In automatic bandwidth control mode (AUTO = 1), the last-written value from manual operation is stored in a temporary location and is recovered when manual operation resumes. Reset clears this bit, enabling acquisition mode.

1 = Tracking mode
0 = Acquisition mode

4.5.3 PLL Multiplier Select Register High

The PLL multiplier select register high (PMSH) contains the programming information for the high byte of the modulo feedback divider.

Address: $0038

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	MUL11	MUL10	MUL9	MUL8
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 4-6. PLL Multiplier Select Register High (PMSH)

MUL11–MUL8 — Multiplier Select Bits

These read/write bits control the high byte of the modulo feedback divider that selects the VCO frequency multiplier N. (See 4.3.3 PLL Circuits and 4.3.6 Programming the PLL.) A value of $0000 in the multiplier select registers configures the modulo feedback divider the same as a value of $0001. Reset initializes the registers to $0040 for a default multiply value of 64.

NOTE

The multiplier select bits have built-in protection such that they cannot be written when the PLL is on (PLLON = 1).

PMSH[7:4] — Unimplemented Bits

These bits have no function and always read as 0s.

Chapter 6
Computer Operating Properly (COP) Module

6.1 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 CONFIG register.

6.2 Functional Description

Figure 6-1 shows the structure of the COP module.

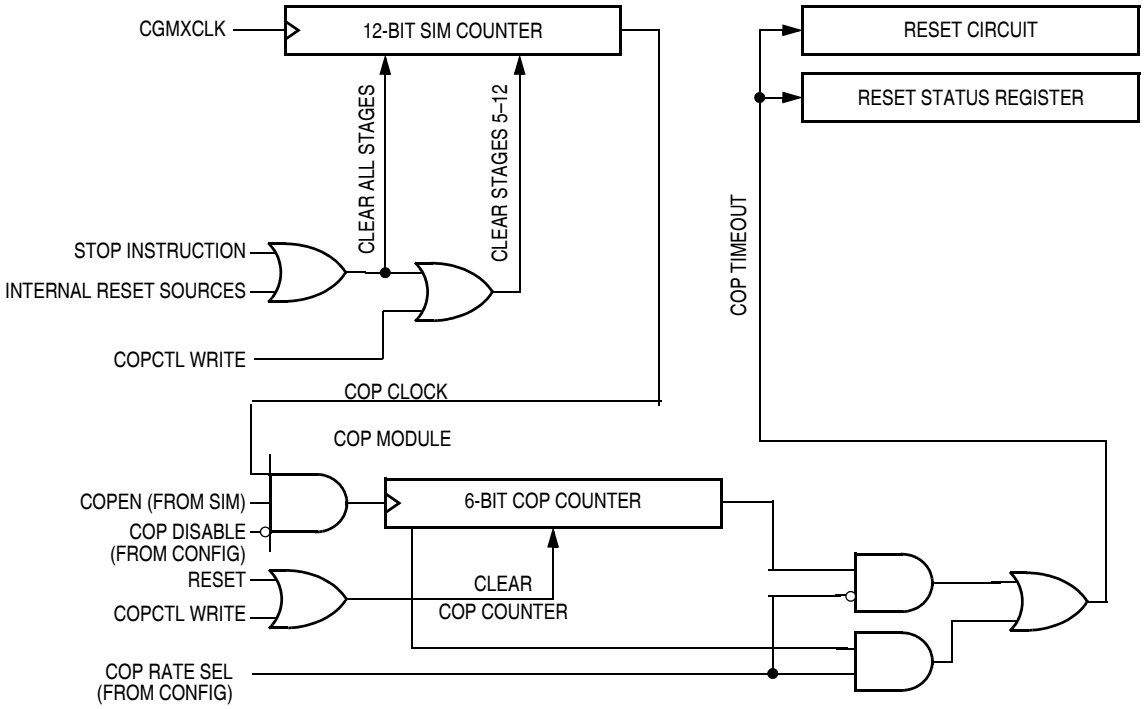
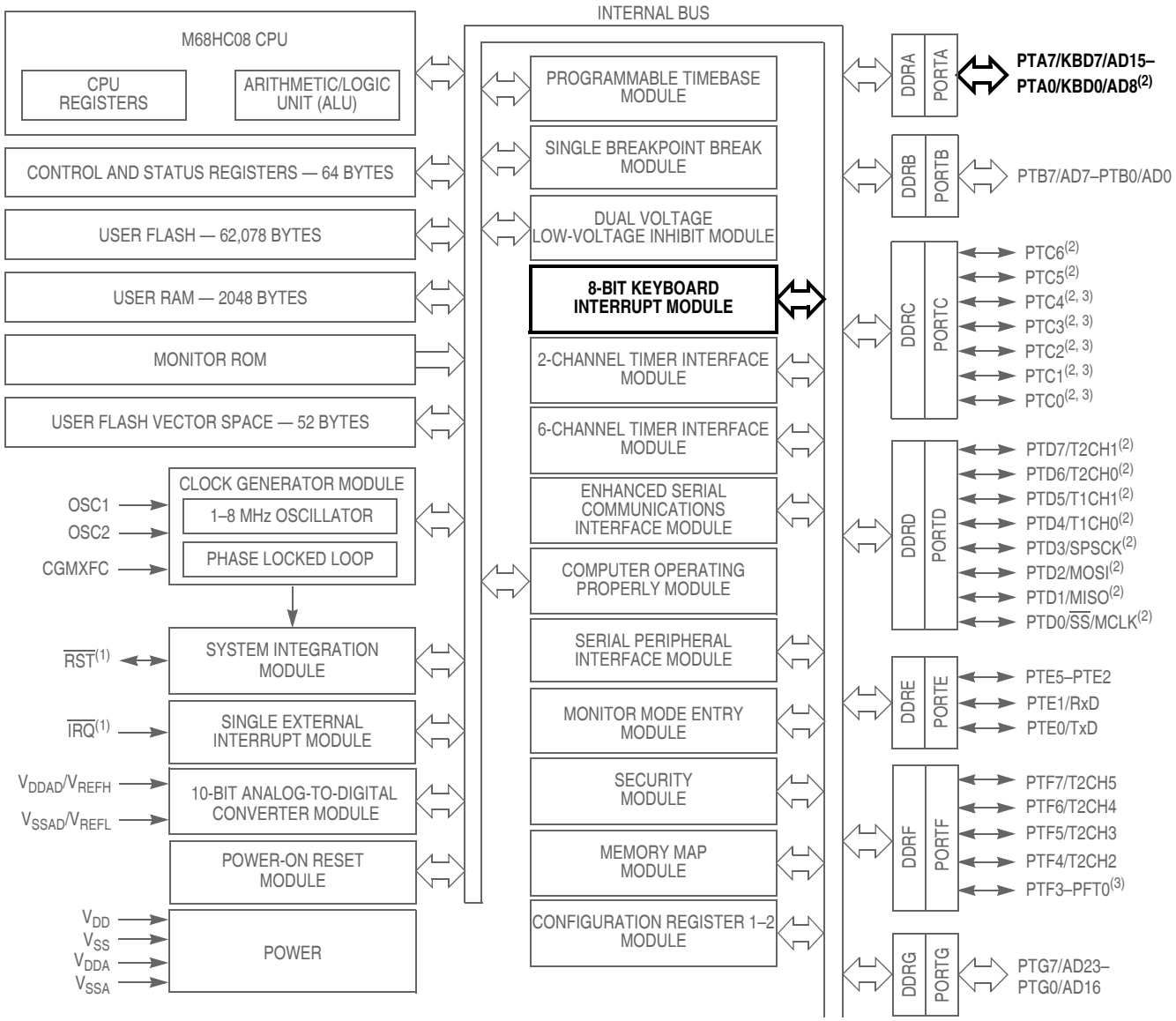


Figure 6-1. COP Block Diagram



1. Pin contains integrated pullup device.
2. Ports are software configurable with pullup device if input port or pullup/pulldown device for keyboard input.
3. Higher current drive port pins

Figure 9-1. Block Diagram Highlighting KBI Block and Pins

actual stop bit. Then a noise error occurs. If more than one of the samples is outside the stop bit, a framing error occurs. In most applications, the baud rate tolerance is much more than the degree of misalignment that is likely to occur.

As the receiver samples an incoming character, it resynchronizes the RT clock on any valid falling edge within the character. Resynchronization within characters corrects misalignments between transmitter bit times and receiver bit times.

Slow Data Tolerance

Figure 13-8 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.

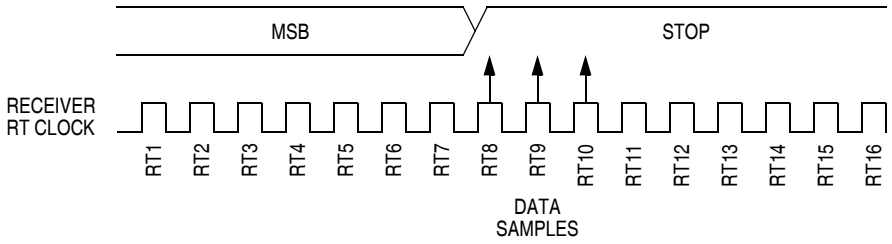


Figure 13-8. Slow Data

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 13-8, 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 13-8, 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 13-9 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.

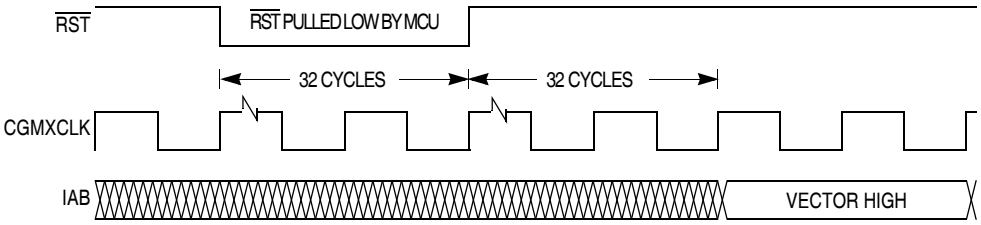


Figure 14-5. Internal Reset Timing

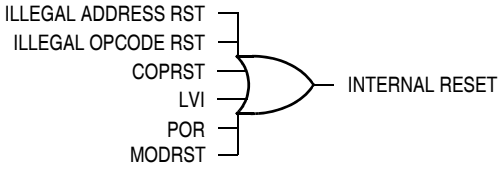


Figure 14-6. Sources of Internal Reset

Table 14-2. Reset Recovery

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

14.3.2.1 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 ($\overline{RST}$) 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.

At power-on, these events occur:

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

14.3.2.2 Computer Operating Properly (COP) Reset

An input to the SIM is reserved for the COP reset signal. The overflow of the COP counter causes an internal reset and sets the COP bit in the SIM reset status register (SRSR) if the COPD bit in the CONFIG1 register is cleared. The SIM actively pulls down the $\overline{RST}$ pin for all internal reset sources.

The COP module is disabled if the $\overline{RST}$ pin or the $\overline{IRQ}$ pin is held at V_{TST} while the MCU is in monitor mode. During a break state, V_{TST} on the $\overline{RST}$ pin disables the COP module.

general-purpose I/O not affecting the SPI. (See 15.6.2 Mode Fault Error.) When CPHA = 0, the first SPSCK edge is the MSB capture strobe. Therefore, the slave must begin driving its data before the first SPSCK edge, and a falling edge on the $\overline{SS}$ pin is used to start the slave data transmission. The slave's $\overline{SS}$ pin must be toggled back to high and then low again between each byte transmitted as shown in Figure 15-6.

When CPHA = 0 for a slave, the falling edge of $\overline{SS}$ indicates the beginning of the transmission. This causes the SPI to leave its idle state and begin driving the MISO pin with the MSB of its data. Once the transmission begins, no new data is allowed into the shift register from the transmit data register. Therefore, the SPI data register of the slave must be loaded with transmit data before the falling edge of $\overline{SS}$. Any data written after the falling edge is stored in the transmit data register and transferred to the shift register after the current transmission.

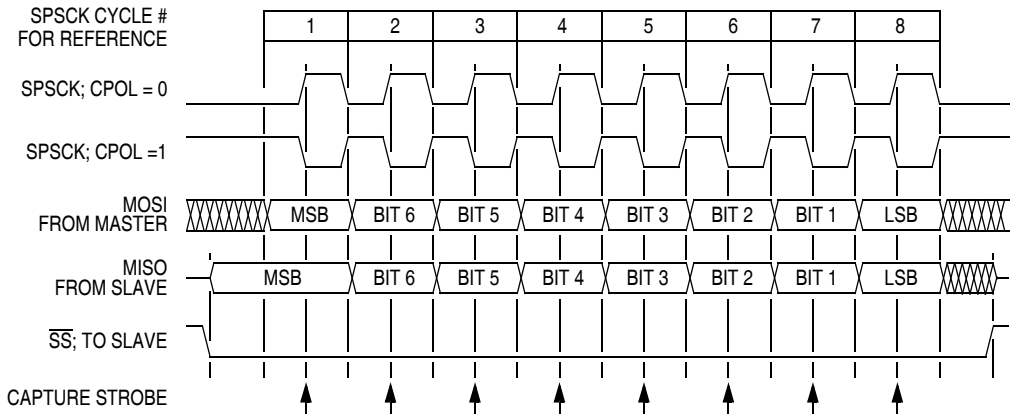


Figure 15-5. Transmission Format (CPHA = 0)

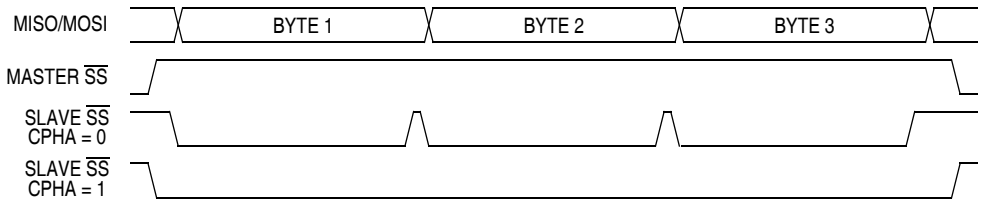


Figure 15-6. CPHA/$\overline{SS}$ Timing

15.4.3 Transmission Format When CPHA = 1

Figure 15-7 shows an SPI transmission in which CPHA = 1. The figure should not be used as a replacement for data sheet parametric information. Two waveforms are shown for SPSCK: one for CPOL = 0 and another for CPOL = 1. The diagram may be interpreted as a master or slave timing diagram since the serial clock (SPSCK), master in/slave out (MISO), and master out/slave in (MOSI) pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the slave select input to the slave. The slave SPI drives its MISO output only when its slave select input ($\overline{SS}$) is low, so that only the selected slave drives to the master. The $\overline{SS}$ pin of the master is not shown but is assumed to be inactive. The $\overline{SS}$ pin of the master must be high or must be reconfigured as general-purpose I/O not affecting the SPI. (See 15.6.2 Mode Fault Error.) When CPHA = 1, the master begins driving its MOSI pin on the first SPSCK

15.6 Error Conditions

The following flags signal SPI error conditions:

- Overflow (OVRF) — Failing to read the SPI data register before the next full byte enters the shift register sets the OVRF bit. The new byte does not transfer to the receive data register, and the unread byte still can be read. OVRF is in the SPI status and control register.
- Mode fault error (MODF) — The MODF bit indicates that the voltage on the slave select pin ($\overline{SS}$) is inconsistent with the mode of the SPI. MODF is in the SPI status and control register.

15.6.1 Overflow Error

The overflow flag (OVRF) becomes set if the receive data register still has unread data from a previous transmission when the capture strobe of bit 1 of the next transmission occurs. The bit 1 capture strobe occurs in the middle of SPSCK cycle 7 (see Figure 15-5 and Figure 15-7.) If an overflow occurs, all data received after the overflow and before the OVRF bit is cleared does not transfer to the receive data register and does not set the SPI receiver full bit (SPRF). The unread data that transferred to the receive data register before the overflow occurred can still be read. Therefore, an overflow error always indicates the loss of data. Clear the overflow flag by reading the SPI status and control register and then reading the SPI data register.

OVRF generates a receiver/error CPU interrupt request if the error interrupt enable bit (ERRIE) is also set. The SPRF, MODF, and OVRF interrupts share the same CPU interrupt vector (see Figure 15-12.) It is not possible to enable MODF or OVRF individually to generate a receiver/error CPU interrupt request. However, leaving MODFEN low prevents MODF from being set.

If the CPU SPRF interrupt is enabled and the OVRF interrupt is not, watch for an overflow condition. Figure 15-10 shows how it is possible to miss an overflow. The first part of Figure 15-10 shows how it is possible to read the SPSCR and SPDR to clear the SPRF without problems. However, as illustrated by the second transmission example, the OVRF bit can be set in between the time that SPSCR and SPDR are read.

In this case, an overflow can be missed easily. Since no more SPRF interrupts can be generated until this OVRF is serviced, it is not obvious that bytes are being lost as more transmissions are completed. To prevent this, either enable the OVRF interrupt or do another read of the SPSCR following the read of the SPDR. This ensures that the OVRF was not set before the SPRF was cleared and that future transmissions can set the SPRF bit. Figure 15-11 illustrates this process. Generally, to avoid this second SPSCR read, enable the OVRF to the CPU by setting the ERRIE bit.

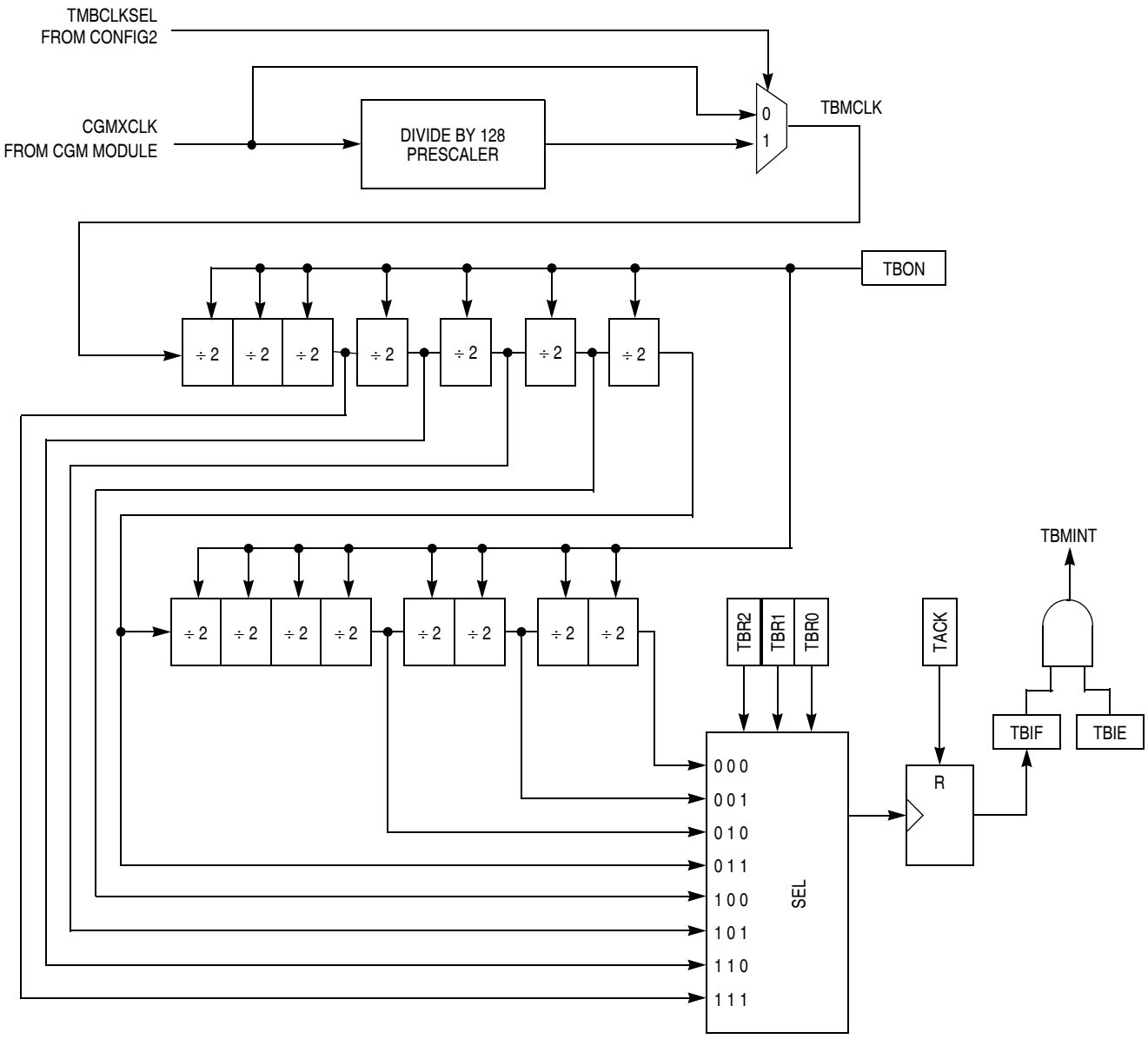


Figure 16-1. Timebase Block Diagram

16.5 TBM Interrupt Rate

The interrupt rate is determined by the equation:

$$t_{TBMRATE} = \frac{Divider}{f_{CGMXCLK}}$$

where:

$f_{CGMXCLK}$ = Frequency supplied from the clock generator (CGM) module

Divider = Divider value as determined by TBR2–TBR0 settings and TMBCLKSEL, see Table 16-1

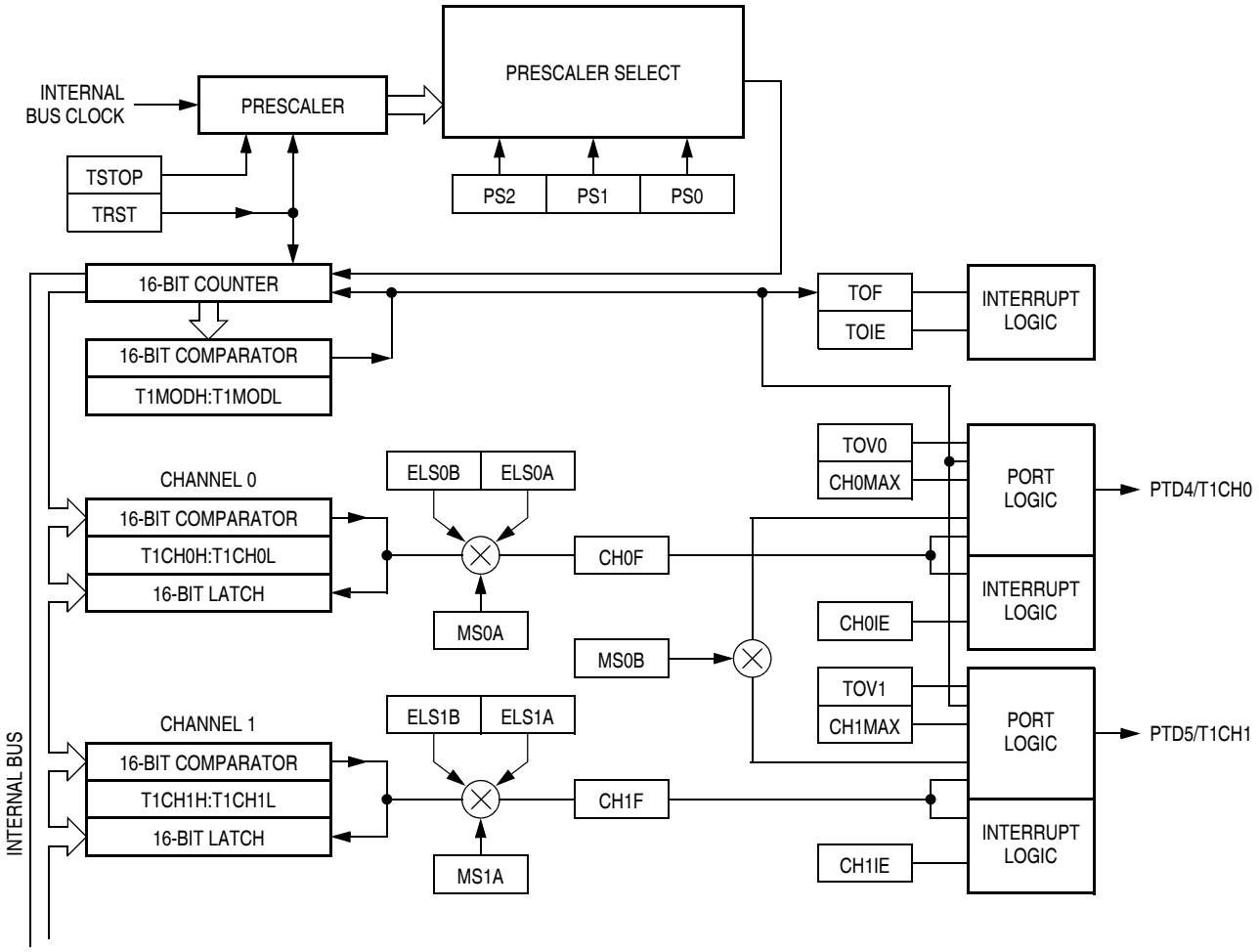


Figure 17-2. TIM1 Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0020	TIM1 Status and Control Register (T1SC) See page 234.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
$0021	TIM1 Counter Register High (T1CNTH) See page 235.	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
$0022	TIM1 Counter Register Low (T1CNTL) See page 235.	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

= Unimplemented

Figure 17-3. TIM1 I/O Register Summary

17.3.3 Output Compare

With the output compare function, the TIM1 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 TIM1 can set, clear, or toggle the channel pin. Output compares can generate TIM1 CPU interrupt requests.

17.3.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in 17.3.3 Output Compare. The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIM1 channel registers.

An unsynchronized write to the TIM1 channel registers to change an output compare value could cause incorrect operation for up to two counter overflow periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that counter overflow period. Also, using a TIM1 overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIM1 may pass the new value before it is written.

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

- When changing to a smaller value, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current output compare pulse. The interrupt routine has until the end of the counter overflow period to write the new value.
- When changing to a larger output compare value, enable TIM1 overflow interrupts and write the new value in the TIM1 overflow interrupt routine. The TIM1 overflow interrupt occurs at the end of the current counter overflow period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same counter overflow period.

17.3.3.2 Buffered Output Compare

Channels 0 and 1 can be linked to form a buffered output compare channel whose output appears on the T1CH0 pin. The TIM1 channel registers of the linked pair alternately control the output.

Setting the MS0B bit in TIM1 channel 0 status and control register (TSC0) links channel 0 and channel 1. The output compare value in the TIM1 channel 0 registers initially controls the output on the T1CH0 pin. Writing to the TIM1 channel 1 registers enables the TIM1 channel 1 registers to synchronously control the output after the TIM1 overflows. At each subsequent overflow, the TIM1 channel registers (0 or 1) that control the output are the ones written to last. T1SC0 controls and monitors the buffered output compare function, and TIM1 channel 1 status and control register (T1SC1) is unused. While the MS0B bit is set, the channel 1 pin, T1CH1, is available as a general-purpose I/O pin.

NOTE

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

Chapter 18
Timer Interface Module (TIM2)

18.1 Introduction

This section describes the timer interface module (TIM2). The TIM2 is a 6-channel timer that provides a timing reference with input capture, output compare, and pulse-width-modulation functions. Figure 18-2 is a block diagram of the TIM2.

18.2 Features

Features of the TIM2 include:

- Six input capture/output compare channels:
  - Rising-edge, falling-edge, or any-edge input capture trigger
  - Set, clear, or toggle output compare action
- Buffered and unbuffered pulse width modulation (PWM) signal generation
- Programmable TIM2 clock input
  - 7-frequency internal bus clock prescaler selection
  - External TIM2 clock input (4-MHz maximum frequency)
- Free-running or modulo up-count operation
- Toggle any channel pin on overflow
- TIM2 counter stop and reset bits

18.3 Functional Description

Figure 18-2 shows the TIM2 structure. The central component of the TIM2 is the 16-bit TIM2 counter that can operate as a free-running counter or a modulo up-counter. The TIM2 counter provides the timing reference for the input capture and output compare functions. The TIM2 counter modulo registers, T2MODH:T2MODL, control the modulo value of the TIM2 counter. Software can read the TIM2 counter value at any time without affecting the counting sequence.

The six TIM2 channels are programmable independently as input capture or output compare channels.

Table 18-2. Mode, Edge, and Level Selection

MSxB	MSxA	ELSxB	ELSxA	Mode	Configuration
X	0	0	0	Output preset	Pin under port control; initial output level high
X	1	0	0		Pin under port control; initial output level low
0	0	0	1	Input capture	Capture on rising edge only
0	0	1	0		Capture on falling edge only
0	0	1	1		Capture on rising or falling edge
0	1	0	0	Output compare or PWM	Software compare only
0	1	0	1		Toggle output on compare
0	1	1	0		Clear output on compare
0	1	1	1		Set output on compare
1	X	0	1	Buffered output compare or buffered PWM	Toggle output on compare
1	X	1	0		Clear output on compare
1	X	1	1		Set output on compare

ELSxB and ELSxA — Edge/Level Select Bits

When channel x is an input capture channel, these read/write bits control the active edge-sensing logic on channel x.

When channel x is an output compare channel, ELSxB and ELSxA control the channel x output behavior when an output compare occurs.

When ELSxB and ELSxA are both clear, channel x is not connected to port D or port F, and pin PTDx/T2CHx or pin PTFx/T2CHx is available as a general- purpose I/O pin. Table 18-2 shows how ELSxB and ELSxA work. Reset clears the ELSxB and ELSxA bits.

NOTE

After initially enabling a TIM2 channel register for input capture operation and selecting the edge sensitivity, clear CHxF to ignore any erroneous edge detection flags.

TOVx — Toggle-On-Overflow Bit

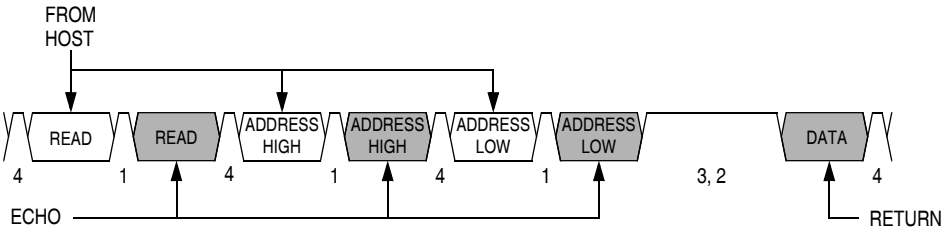
When channel x is an output compare channel, this read/write bit controls the behavior of the channel x output when the TIM2 counter overflows. When channel x is an input capture channel, TOVx has no effect. Reset clears the TOVx bit.

1 = Channel x pin toggles on TIM2 counter overflow.

0 = Channel x pin does not toggle on TIM2 counter overflow.

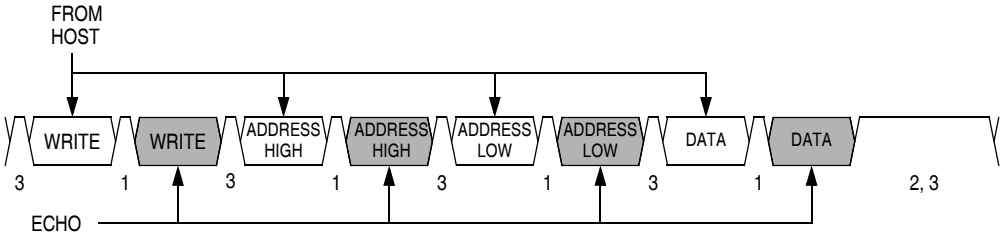
NOTE

When TOVx is set, a TIM2 counter overflow takes precedence over a channel x output compare if both occur at the same time.



Notes:
1 = Echo delay, approximately 2 bit times
2 = Data return delay, approximately 2 bit times
3 = Cancel command delay, 11 bit times
4 = Wait 1 bit time before sending next byte.

Figure 19-14. Read Transaction



Notes:
1 = Echo delay, approximately 2 bit times
2 = Cancel command delay, 11 bit times
3 = Wait 1 bit time before sending next byte.

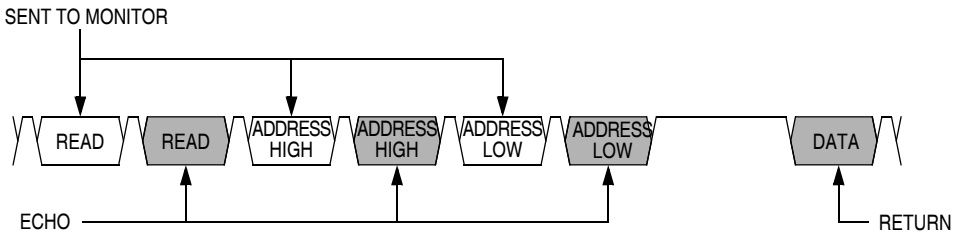
Figure 19-15. Write Transaction

A brief description of each monitor mode command is given in Table 19-3 through Table 19-8.

Table 19-3. READ (Read Memory) Command

Description	Read byte from memory
Operand	2-byte address in high-byte:low-byte order
Data Returned	Returns contents of specified address
Opcode	$4A

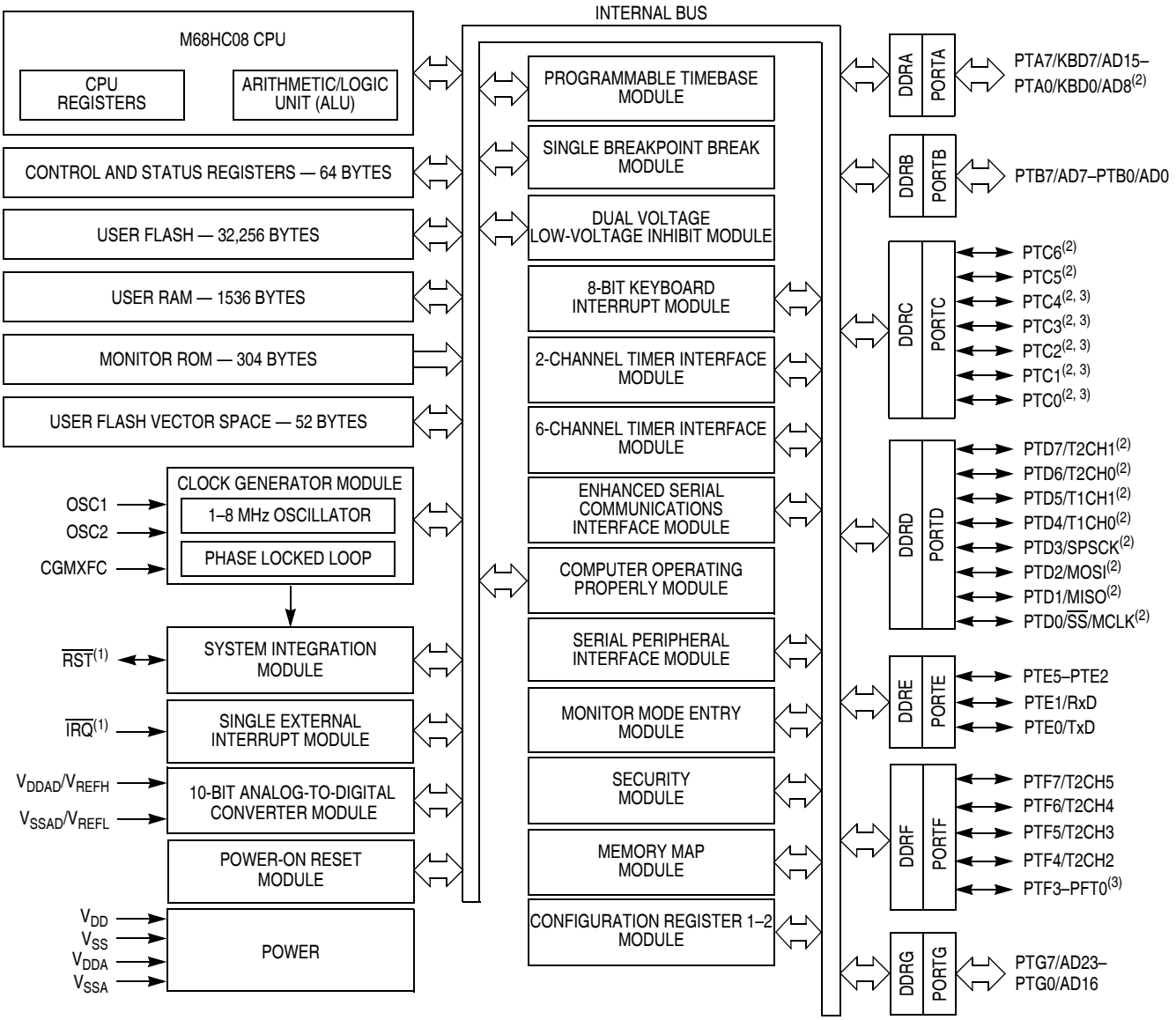
Command Sequence



20.11 3.3-Volt ADC Characteristics

Characteristic[1]	Symbol	Min	Max	Unit	Comments
Supply voltage	V_{DDAD}	3.0	3.6	V	V_{DDAD} should be tied to the same potential as V_{DD} via separate traces.
Input voltages	V_{ADIN}	0	V_{DDAD}	V	V_{ADIN} <= V_{DDAD}
Resolution	B_{AD}	10	10	Bits	
Absolute accuracy	A_{AD}	–6	+6	LSB	Includes quantization
ADC internal clock	f_{ADIC}	500 k	1.048 M	Hz	$t_{AIC} = 1/f_{ADIC}$
Conversion range	R_{AD}	V_{SSAD}	V_{DDAD}	V	
Power-up time	t_{ADPU}	16	—	t_{AIC} cycles	
Conversion time	t_{ADC}	16	17	t_{AIC} cycles	
Sample time	t_{ADS}	5	—	t_{AIC} cycles	
Monotonicity	M_{AD}	Guaranteed			
Zero input reading	Z_{ADI}	000	005	Hex	$V_{ADIN} = V_{SSA}$
Full-scale reading	F_{ADI}	3FA	3FF	Hex	$V_{ADIN} = V_{DDA}$
Input capacitance	C_{ADI}	—	30	pF	Not tested
V_{DDAD}/V_{REFH} current	I_{VREF}	—	1.2	mA	
Absolute accuracy (8-bit truncation mode)	A_{AD}	–1	+1	LSB	Includes quantization
Quantization error (8-bit truncation mode)	—	–1/8	+7/8	LSB	

1. V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, V_{DDAD}/V_{REFH} = 3.3 Vdc ± 10%, V_{SSAD}/V_{REFL} = 0 Vdc



1. Pin contains integrated pullup device.
2. Ports are software configurable with pullup device if input port or pullup/pulldown device for keyboard input.
3. Higher current drive port pins

Figure B-1. MC68HC908GR32A Block Diagram