

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	Active
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	LINbus, SCI, SPI
Peripherals	POR, PWM
Number of I/O	24
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 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908ey16amfje

5.1	Introduction	63
5.2	Functional Description	63

Chapter 6 Computer Operating Properly

6.1	Introduction	69
6.2	Functional Description	69
6.3	I/O Signals	70
6.3.1	CGMXCLK	70
6.3.2	STOP Instruction	70
6.3.3	COPCTL Write	70
6.3.4	Power-On Reset.	70
6.3.5	Internal Reset.	70
6.3.6	Reset Vector Fetch	70
6.3.7	COPD.	70
6.3.8	COPRS	71
6.4	COP Control Register	71
6.5	Interrupts	71
6.6	Monitor Mode	71
6.7	Low-Power Modes	71
6.7.1	Wait Mode	71
6.7.2	Stop Mode	71
6.8	COP Module During Break Interrupts	71

Chapter 7 Central Processor Unit (CPU)

7.1	Introduction	73
7.2	Features.	73
7.3	CPU Registers	73
7.3.1	Accumulator	74
7.3.2	Index Register	74
7.3.3	Stack Pointer	75
7.3.4	Program Counter	75
7.3.5	Condition Code Register	76
7.4	Arithmetic/Logic Unit (ALU)	77
7.5	Low-Power Modes	77
7.5.1	Wait Mode	77
7.5.2	Stop Mode	77
7.6	CPU During Break Interrupts	77
7.7	Instruction Set Summary	78
7.8	Opcode Map	83

Chapter 8 Internal Clock Generator (ICG) Module

8.1	Introduction	85
8.2	Features.	85

16.4	Interrupts	199
16.5	TBM Interrupt Rate	200
16.6	Low-Power Modes	201
16.6.1	Wait Mode	201
16.6.2	Stop Mode	201
16.7	Timebase Control Register	202

Chapter 17

Timer Interface A (TIMA) Module

17.1	Introduction	203
17.2	Features	203
17.3	Functional Description	203
17.3.1	TIMA Counter Prescaler	203
17.3.2	Input Capture	205
17.3.3	Output Compare	206
17.3.3.1	Unbuffered Output Compare	206
17.3.3.2	Buffered Output Compare	206
17.3.4	Pulse Width Modulation (PWM)	207
17.3.4.1	Unbuffered PWM Signal Generation	208
17.3.4.2	Buffered PWM Signal Generation	208
17.3.4.3	PWM Initialization	209
17.4	Interrupts	209
17.5	Low-Power Modes	210
17.5.1	Wait Mode	210
17.5.2	Stop Mode	210
17.6	TIMA During Break Interrupts	210
17.7	I/O Signals	210
17.7.1	TIMA Channel I/O Pins (PTD0/TACH0, PTD1/TACH1)	210
17.8	I/O Registers	211
17.8.1	TIMA Status and Control Register	211
17.8.2	TIMA Counter Registers	213
17.8.3	TIMA Counter Modulo Registers	213
17.8.4	TIMA Channel Status and Control Registers	214
17.8.5	TIMA Channel Registers	217

Chapter 18

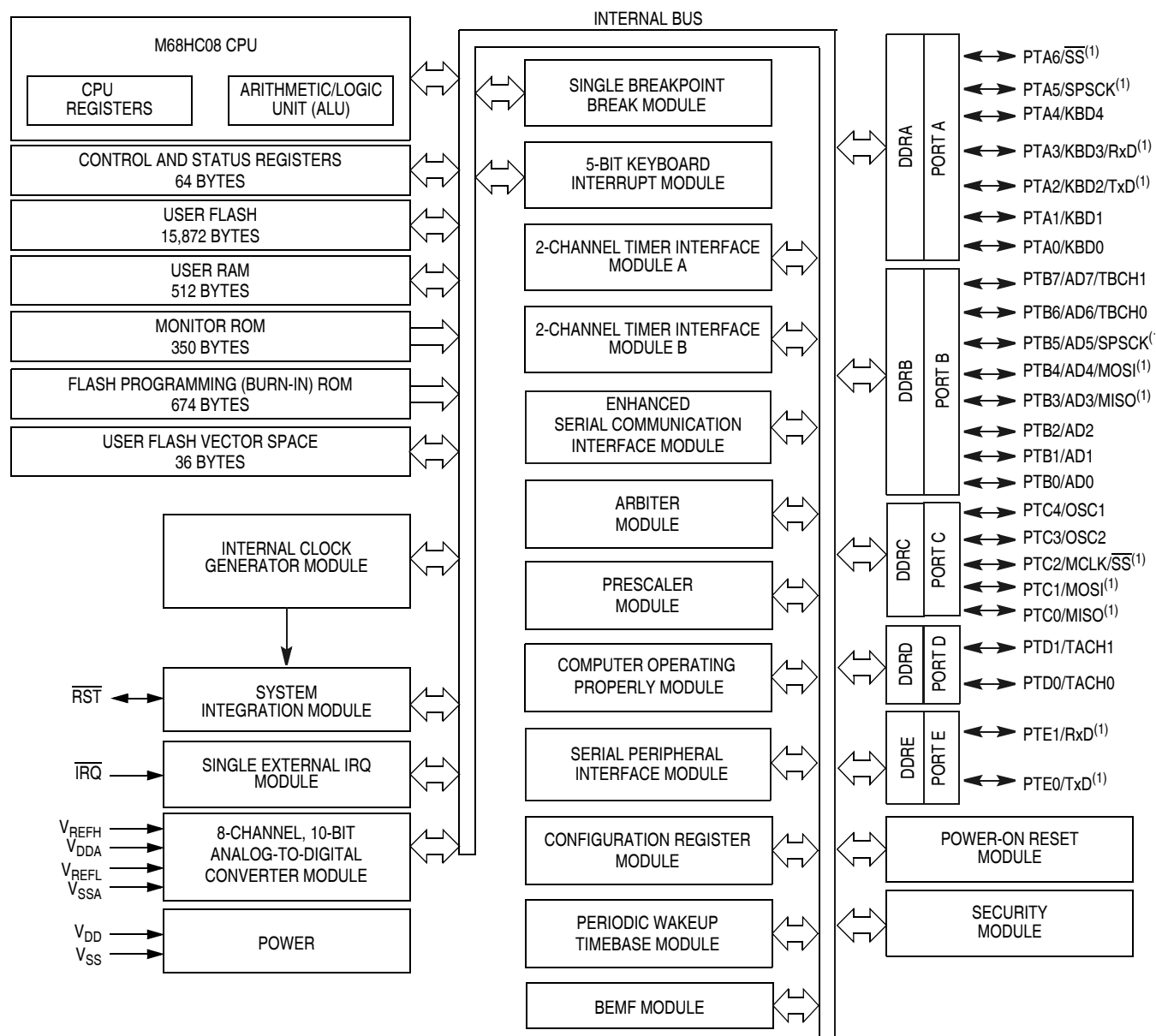
Timer Interface B (TIMB) Module

18.1	Introduction	219
18.2	Features	219
18.3	Functional Description	219
18.3.1	TIMB Counter Prescaler	219
18.3.2	Input Capture	221
18.3.3	Output Compare	222
18.3.3.1	Unbuffered Output Compare	222
18.3.3.2	Buffered Output Compare	223
18.3.4	Pulse Width Modulation (PWM)	223
18.3.4.1	Unbuffered PWM Signal Generation	224

18.3.4.2	Buffered PWM Signal Generation	224
18.3.4.3	PWM Initialization	225
18.4	Interrupts	225
18.5	Low-Power Modes	226
18.5.1	Wait Mode	226
18.5.2	Stop Mode	226
18.6	TIMB During Break Interrupts	226
18.7	I/O Signals	226
18.7.1	TIMB Channel I/O Pins (PTB7/AD7/TBCH1–PTB6/AD6/TBCH0)	226
18.8	I/O Registers	227
18.8.1	TIMB Status and Control Register	227
18.8.2	TIMB Counter Registers	229
18.8.3	TIMB Counter Modulo Registers	229
18.8.4	TIMB Channel Status and Control Registers	230
18.8.5	TIMB Channel Registers	233

Chapter 19 Development Support

19.1	Introduction	235
19.2	Break Module (BRK)	235
19.2.1	Functional Description	235
19.2.1.1	Flag Protection During Break Interrupts	237
19.2.1.2	TIM During Break Interrupts	237
19.2.1.3	COP During Break Interrupts	237
19.2.2	Break Module Registers	237
19.2.2.1	Break Status and Control Register	238
19.2.2.2	Break Address Registers	238
19.2.2.3	Break Status Register	239
19.2.2.4	Break Flag Control Register	239
19.2.3	Low-Power Modes	239
19.3	Monitor Module (MON)	240
19.3.1	Functional Description	240
19.3.1.1	Normal Monitor Mode	243
19.3.1.2	Forced Monitor Mode	244
19.3.1.3	Monitor Vectors	245
19.3.1.4	Data Format	245
19.3.1.5	Break Signal	245
19.3.1.6	Baud Rate	246
19.3.1.7	Commands	246
19.3.2	Security	249
19.3.3	Extended Security	250
19.4	Routines Supported in ROM	250
19.4.1	Variables Used in the Routines	251
19.4.2	How to Use the Routines	251
19.4.2.1	GetByte	253
19.4.2.2	PutByte	254
19.4.2.3	Verify	254



NOTE:

1. The locations of the ESCI and SPI pins are user selectable using CONFIG3 option bits.

Figure 1-1. MCU Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0	
\$0000	Port A Data Register (PTA) See page 123.	Read:	0	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0	
		Write:									
		Reset:	Unaffected by reset								
		\$0001	Port B Data Register (PTB) See page 125.	Read:		PTB6	PTB5	PTB4	PTB3	PTB2	PTB1
Write:	PTB7										
Reset:	Unaffected by reset										
\$0002	Port C Data Register (PTC) See page 126.			Read:	0	0	0	PTC4	PTC3	PTC2	PTC1
		Write:									
		Reset:	Unaffected by reset								
		\$0003	Port D Data Register (PTD) See page 128.	Read:	0	0	0	0	0	0	PTD1
Write:											
Reset:	Unaffected by reset										
\$0004	Data Direction Register A (DDRA) See page 123.			Read:	0	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1
		Write:									
		Reset:	0	0	0	0	0	0	0	0	
		\$0005	Data Direction Register B (DDRB) See page 125.	Read:		DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1
Write:	DDRB7										
Reset:	0			0	0	0	0	0	0	0	
\$0006	Data Direction Register C (DDRC) See page 127.			Read:	MCLKEN	0	0	DDRC4	DDRC3	DDRC2	DDRC1
		Write:									
		Reset:	0	0	0	0	0	0	0	0	
		\$0007	Data Direction Register D (DDRD) See page 128.	Read:	0	0	0	0	0	0	DDRD1
Write:											
Reset:	0			0	0	0	0	0	0	0	
\$0008	Port E Data Register (PTE) See page 130.			Read:	0	0	0	0	0	0	PTE1
		Write:									
		Reset:	Unaffected by reset								
		\$0009	Configuration Register 3 (CONFIG3) See page 67.	Read:		RNGSEL	ESCISRE	SPISRE	MCLKSRE	PORTSRE	ESCISEL
Write:											
Reset:	0			1	0	0	0	0	0	0	
\$000A	Data Direction Register E (DDRE) See page 130.			Read:	0	0	0	0	0	0	DDRE1
		Write:									
		Reset:	0	0	0	0	0	0	0	0	
		\$000B	BEMF Register (BEMF) See page 61.	Read:	BEMF7	BEMF6	BEMF5	BEMF4	BEMF3	BEMF2	BEMF1
Write:											
Reset:	0			0	0	0	0	0	0	0	
				= Unimplemented			R = Reserved		U = Unaffected		

Figure 2-2. Control, Status, and Data Registers (Sheet 1 of 7)

Table 7-1. Instruction Set Summary (Sheet 6 of 6)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	–	–	1	–	–	–	INH	83		9
TAP	Transfer A to CCR	CCR ← (A)	↑	↑	↑	↑	↑	↑	INH	84		2
TAX	Transfer A to X	X ← (A)	–	–	–	–	–	–	INH	97		1
TPA	Transfer CCR to A	A ← (CCR)	–	–	–	–	–	–	INH	85		1
TST <i>opr</i> TSTA TSTX TST <i>opr</i> ,X TST ,X TST <i>opr</i> ,SP	Test for Negative or Zero	(A) – \$00 or (X) – \$00 or (M) – \$00	0	–	–	↑	↑	–	DIR INH INH IX1 IX SP1	3D 4D 5D 6D 7D 9E6D	dd ff ff	3 1 1 3 2 4
TSX	Transfer SP to H:X	H:X ← (SP) + 1	–	–	–	–	–	–	INH	95		2
TXA	Transfer X to A	A ← (X)	–	–	–	–	–	–	INH	9F		1
TXS	Transfer H:X to SP	(SP) ← (H:X) – 1	–	–	–	–	–	–	INH	94		2
WAIT	Enable Interrupts; Wait for Interrupt	I bit ← 0; Inhibit CPU clocking until interrupted	–	–	0	–	–	–	INH	8F		1

A	Accumulator	<i>n</i>	Any bit
C	Carry/borrow bit	<i>opr</i>	Operand (one or two bytes)
CCR	Condition code register	PC	Program counter
dd	Direct address of operand	PCH	Program counter high byte
dd rr	Direct address of operand and relative offset of branch instruction	PCL	Program counter low byte
DD	Direct to direct addressing mode	REL	Relative addressing mode
DIR	Direct addressing mode	<i>rel</i>	Relative program counter offset byte
DIX+	Direct to indexed with post increment addressing mode	rr	Relative program counter offset byte
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	SP1	Stack pointer, 8-bit offset addressing mode
EXT	Extended addressing mode	SP2	Stack pointer 16-bit offset addressing mode
ff	Offset byte in indexed, 8-bit offset addressing	SP	Stack pointer
H	Half-carry bit	U	Undefined
H	Index register high byte	V	Overflow bit
hh ll	High and low bytes of operand address in extended addressing	X	Index register low byte
I	Interrupt mask	Z	Zero bit
ii	Immediate operand byte	&	Logical AND
IMD	Immediate source to direct destination addressing mode		Logical OR
IMM	Immediate addressing mode	⊕	Logical EXCLUSIVE OR
INH	Inherent addressing mode	()	Contents of
IX	Indexed, no offset addressing mode	–()	Negation (two's complement)
IX+	Indexed, no offset, post increment addressing mode	#	Immediate value
IX+D	Indexed with post increment to direct addressing mode	«	Sign extend
IX1	Indexed, 8-bit offset addressing mode	←	Loaded with
IX1+	Indexed, 8-bit offset, post increment addressing mode	?	If
IX2	Indexed, 16-bit offset addressing mode	:	Concatenated with
M	Memory location	↑	Set or cleared
N	Negative bit	—	Not affected

7.8 Opcode Map

See [Table 7-2](#).

8.3.3.2 External Clock Input Path

The external clock input path is the means by which the microcontroller uses an external clock source. The input to the path is the PTC4/OSC1 pin and the output is the external clock (ECLK). The path, which contains input buffering, is enabled when the external clock generator enable signal (ECGEN) is set. When not enabled, the PTC4/OSC1 pin reverts to its port function.

8.3.4 Clock Monitor Circuit

The ICG contains a clock monitor circuit which, when enabled, will continuously monitor both the external clock (ECLK) and the internal clock (ICLK) to determine if either clock source has been corrupted. The clock monitor circuit, shown in Figure 8-5, contains these blocks:

- Clock monitor reference generator
- Internal clock activity detector
- External clock activity detector

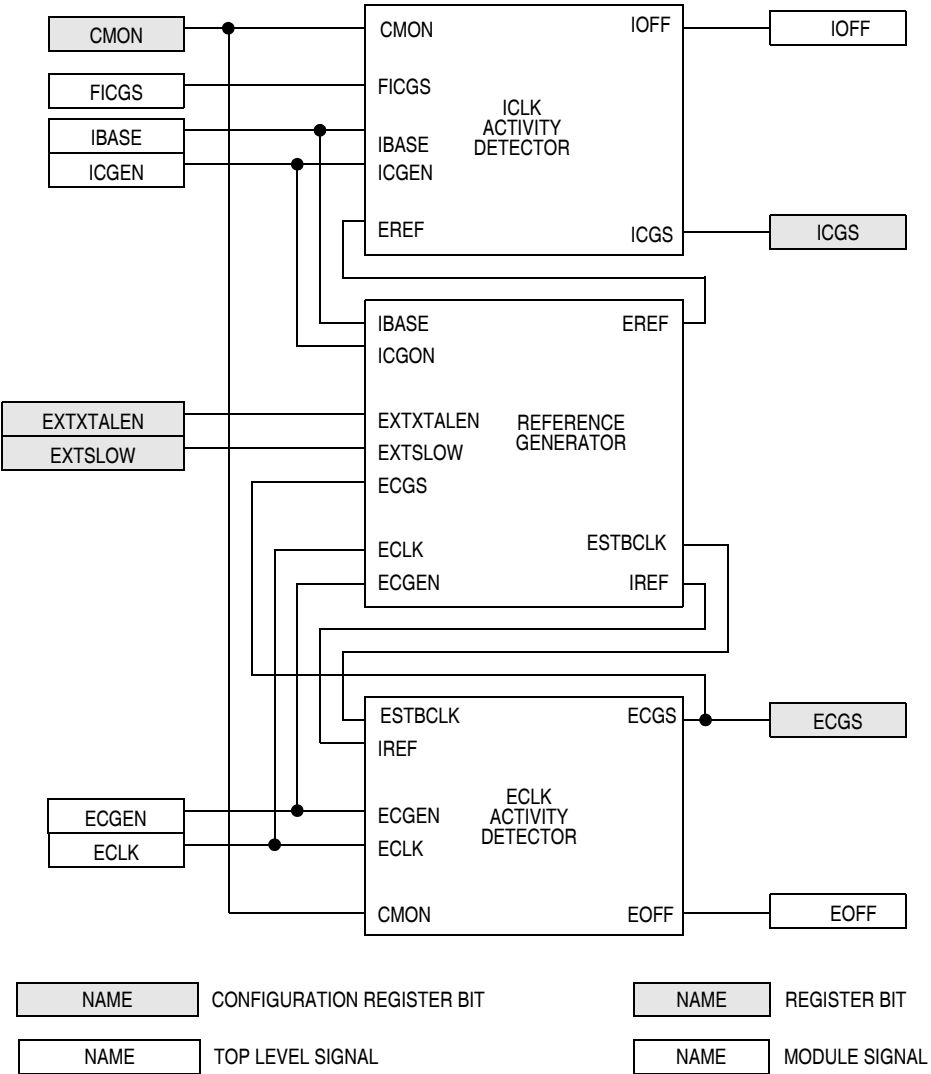


Figure 8-5. Clock Monitor Block Diagram

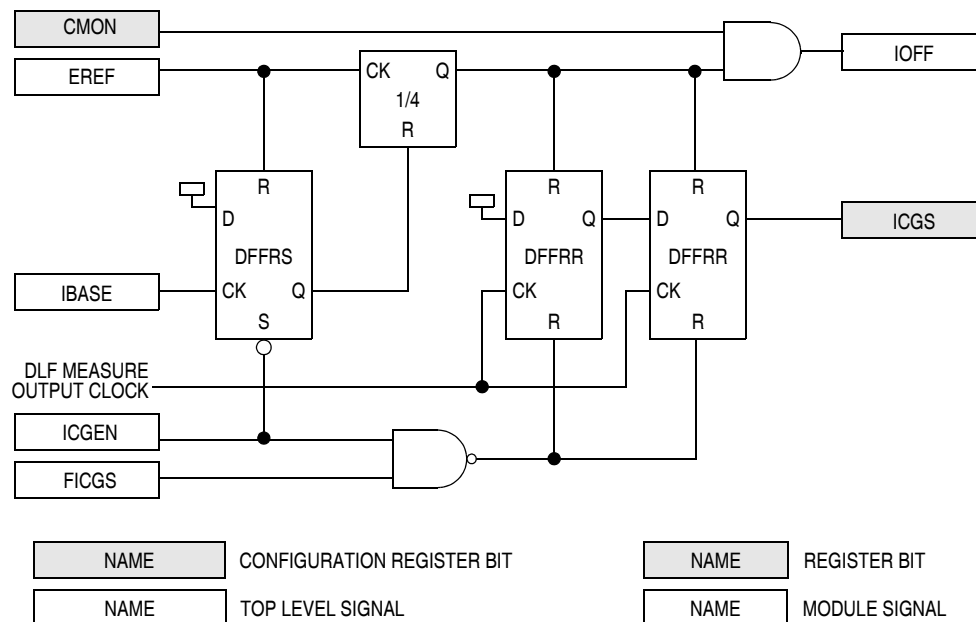


Figure 8-6. Internal Clock Activity Detector

The external clock stable bit (ECGS) is also generated in the external clock activity detector. ECGS is set on a falling edge of the external stabilization clock (ESTBCLK). This will be 4096 ECLK cycles after the external clock generator on bit is set, or the MCU exits stop mode (ECGEN = 1) if the external crystal enable (EXTXTALEN) in the CONFIG is set, or 16 cycles when EXTXTALEN is clear. ECGS is cleared when the external clock generator is turned off or in stop mode (ECGEN is clear) or when EOFF is set.

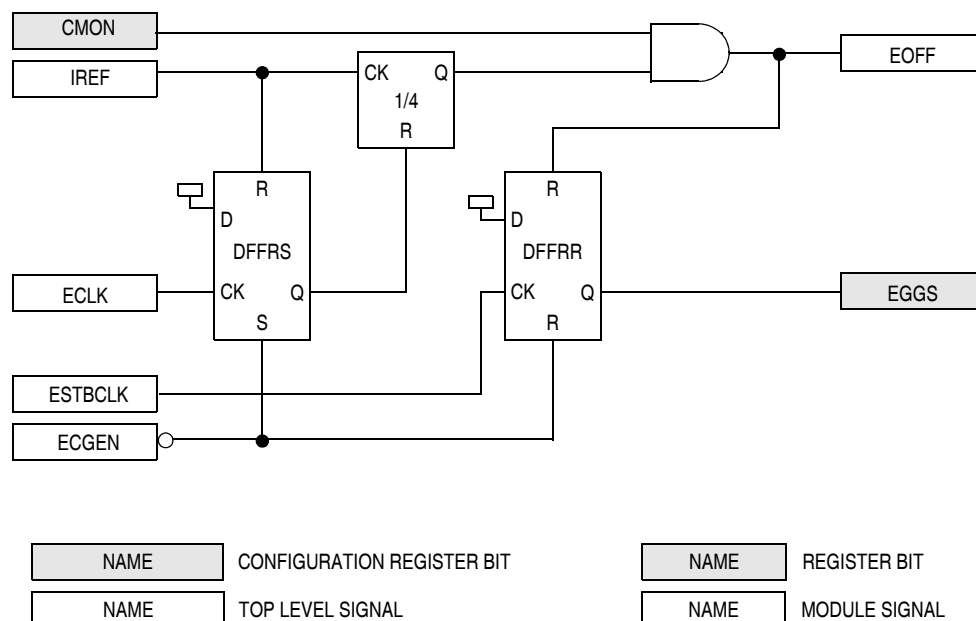


Figure 8-7. External Clock Activity Detector

Chapter 11

Low-Voltage Inhibit (LVI) Module

11.1 Introduction

This section describes the low-voltage inhibit (LVI) module, which monitors the voltage on the V_{DD} pin and can force a reset when the V_{DD} voltage falls to the LVI trip voltage.

11.2 Features

Features include:

- Programmable LVI reset
- Programmable power consumption
- 3 V or 5 V selectable trip point

11.3 Functional Description

Figure 11-1 shows the structure of the LVI module. The LVI is enabled out of reset. The following bits located in the configuration register can alter the default conditions:

- Setting the LVI power disable bit, LVIPWRD, disables the LVI
- Setting the LVI reset disable bit, LVIRSTD, prevents the LVI module from generating a reset.
- Setting the LVI enable in stop mode bit, LVISTOP, enables the LVI to continue monitoring the voltage level on V_{DD} while in stop mode.

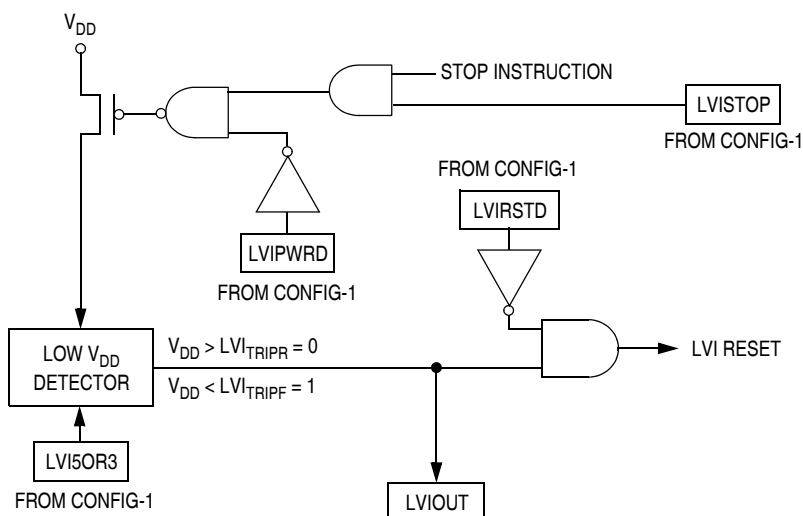


Figure 11-1. LVI Module Block Diagram

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 13-7 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.

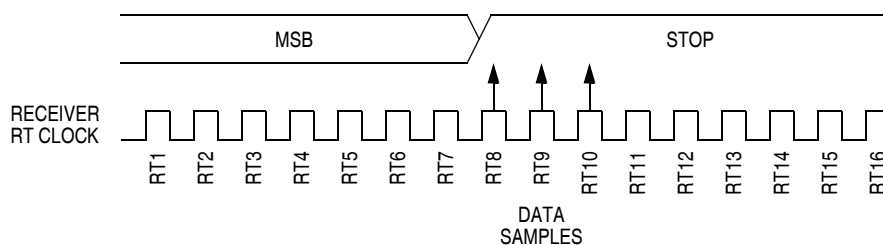


Figure 13-7. Slow Data

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

With the misaligned character shown in Figure 13-7, the receiver counts 154 RT cycles at the point when the count of the transmitting device is $9 \text{ bit times} \times 16 \text{ RT cycles} + 3 \text{ RT cycles} = 147 \text{ RT cycles}$.

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
 $10 \text{ bit times} \times 16 \text{ RT cycles} + 10 \text{ RT cycles} = 170 \text{ RT cycles}$.

With the misaligned character shown in Figure 13-7, the receiver counts 170 RT cycles at the point when the count of the transmitting device is

$10 \text{ bit times} \times 16 \text{ RT cycles} + 3 \text{ RT cycles} = 163 \text{ RT cycles}$.

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

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

14.6.6 Status Flag Protection in Break Mode

The SIM controls whether status flags contained in other modules can be cleared during break mode. The user can select whether flags are protected from being cleared by properly initializing the break clear flag enable bit (BCFE) in the SIM break flag control register (SBFCR).

Protecting flags in break mode ensures that set flags will not be cleared while in break mode. This protection allows registers to be freely read and written during break mode without losing status flag information.

Setting the BCFE bit enables the clearing mechanisms. Once cleared in break mode, a flag remains cleared even when break mode is exited. Status flags with a two-step clearing mechanism — for example, a read of one register followed by the read or write of another — are protected, even when the first step is accomplished prior to entering break mode. Upon leaving break mode, execution of the second step will clear the flag as normal.

14.7 Low-Power Modes

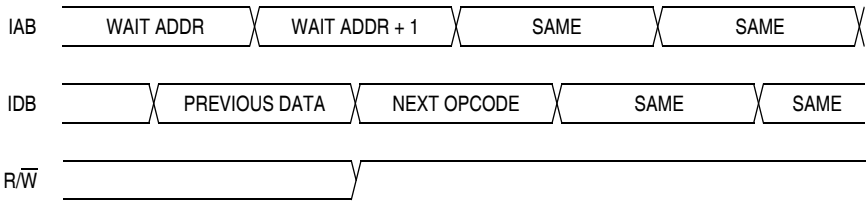
Executing the WAIT or STOP instruction puts the MCU in a low power- consumption mode for standby situations. The SIM holds the CPU in a non-clocked state. Both STOP and WAIT clear the interrupt mask (I) in the condition code register, allowing interrupts to occur. Low-power modes are exited via an interrupt or reset.

14.7.1 Wait Mode

In wait mode, the CPU clocks are inactive while one set of peripheral clocks continues to run. [Figure 14-14](#) shows the timing for wait mode entry.

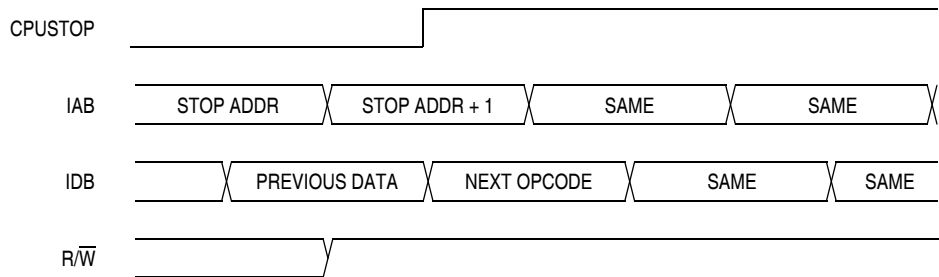
A module that is active during wait mode can wake up the CPU with an interrupt if the interrupt is enabled. Stacking for the interrupt begins one cycle after the WAIT instruction during which the interrupt occurred. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

Wait mode can also be exited by a reset. If the COP disable bit, COPD, in the configuration register is 0, 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 14-14. Wait Mode Entry Timing



Note: Previous data can be operand data or the STOP opcode, depending on the last instruction.

Figure 14-17. Stop Mode Entry Timing

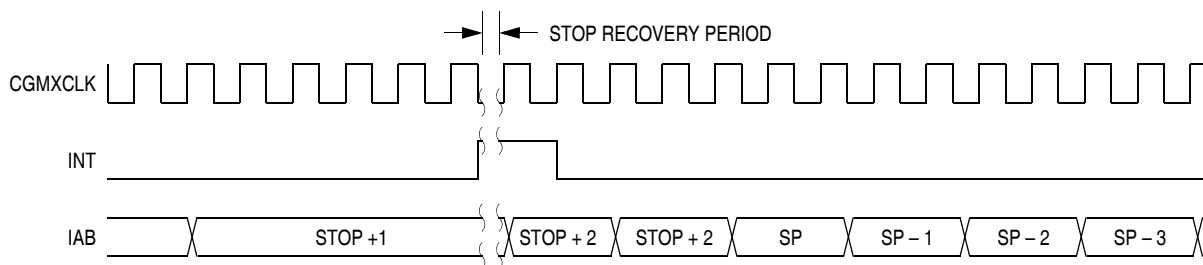


Figure 14-18. Stop Mode Recovery from Interrupt

14.8 SIM Registers

The SIM has three memory mapped registers. [Table 14-4](#) shows the mapping of these registers.

Table 14-4. SIM Registers

Address	Register	Access Mode
\$FE00	SBSR	User
\$FE01	SRSR	User
\$FE03	SBFCR	User

14.8.1 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from stop or wait mode.

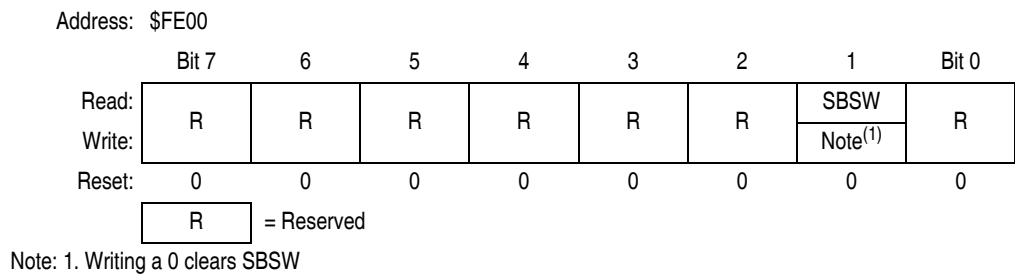


Figure 14-19. SIM Break Status Register (SBSR)

16.7 Timebase Control Register

The timebase has one register, the timebase control register (TBCR), which is used to enable the timebase interrupts and set the rate.

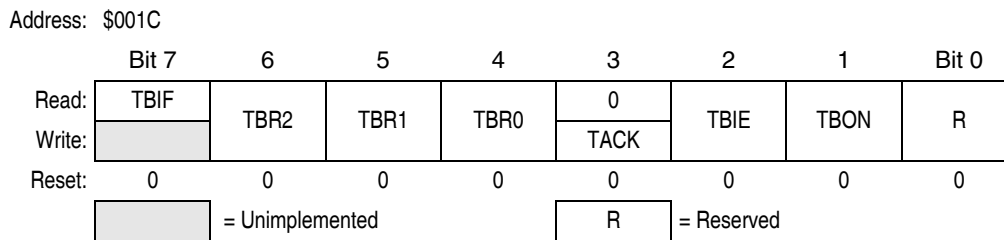


Figure 16-2. Timebase Control Register (TBCR)

TBIF — Timebase Interrupt Flag

This read-only flag bit is set when the timebase counter has rolled over.

1 = Timebase interrupt pending

0 = Timebase interrupt not pending

TBR2–TBR0 — Timebase Divider Selection Bits

These read/write bits select the tap in the counter to be used for timebase interrupts as shown in [Table 16-1](#).

NOTE

Do not change TBR2–TBR0 bits while the timebase is enabled (TBON = 1).

TACK— Timebase ACKnowledge Bit

The TACK bit is a write-only bit and always reads as 0. Writing a 1 to this bit clears TBIF, the timebase interrupt flag bit. Writing a 0 to this bit has no effect.

1 = Clear timebase interrupt flag

0 = No effect

TBIE — Timebase Interrupt Enabled Bit

This read/write bit enables the timebase interrupt when the TBIF bit becomes set. Reset clears the TBIE bit.

1 = Timebase interrupt is enabled.

0 = Timebase interrupt is disabled.

TBON — Timebase Enabled Bit

This read/write bit enables the timebase. Timebase may be turned off to reduce power consumption when its function is not necessary. The counter can be initialized by clearing and then setting this bit. Reset clears the TBON bit.

1 = Timebase is enabled.

0 = Timebase is disabled and the counter initialized to 0s.

NOTE

Clearing TBON has no effect on the TBIF flag.

captured. To measure a pulse width, two alternate polarity edges are captured. Software should track the overflows at the 16-bit module counter to extend its range.

Another use for the input capture function is to establish a time reference. In this case, an input capture function is used in conjunction with an output compare function. For example, to activate an output signal a specified number of clock cycles after detecting an input event (edge), use the input capture function to record the time at which the edge occurred. A number corresponding to the desired delay is added to this captured value and stored to an output compare register (see [17.8.5 TIMA Channel Registers](#)). Because both input captures and output compares are referenced to the same 16-bit modulo counter, the delay can be controlled to the resolution of the counter independent of software latencies.

Reset does not affect the contents of the input capture channel register (TACHxH–TACHxL).

17.3.3 Output Compare

With the output compare function, the TIMA 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 TIMA can set, clear, or toggle the channel pin. Output compares can generate TIMA 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 TIMA channel registers.

An unsynchronized write to the TIMA 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 TIMA overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIMA may pass the new value before it is written.

Use these 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 TIMA overflow interrupts and write the new value in the TIMA overflow interrupt routine. The TIMA 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 PTD0/TACH0 pin. The TIMA channel registers of the linked pair alternately control the output.

Setting the MS0B bit in TIMA channel 0 status and control register (TASC0) links channel 0 and channel 1. The output compare value in the TIMA channel 0 registers initially controls the output on the PTD0/TACH0 pin. Writing to the TIMA channel 1 registers enables the TIMA channel 1 registers to

17.3.4.3 PWM Initialization

To ensure correct operation when generating unbuffered or buffered PWM signals, use this initialization procedure:

1. In the TIMA status and control register (TASC):
 - a. Stop the TIMA counter by setting the TIMA stop bit, TSTOP.
 - b. Reset the TIMA counter prescaler by setting the TIMA reset bit, TRST.
2. In the TIMA counter modulo registers (TAMODH–TAMODL), write the value for the required PWM period.
3. In the TIMA channel x registers (TACHxH–TACHxL), write the value for the required pulse width.
4. In TIMA channel x status and control register (TASCx):
 - a. Write 0:1 (for unbuffered output compare or PWM signals) or 1:0 (for buffered output compare or PWM signals) to the mode select bits, MSxB–MSxA. See [Table 17-2](#).
 - b. Write 1 to the toggle-on-overflow bit, TOVx.
 - c. Write 1:0 (to clear output on compare) or 1:1 (to set output on compare) to the edge/level select bits, ELSxB–ELSxA. The output action on compare must force the output to the complement of the pulse width level. See [Table 17-2](#).

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 can also cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.

5. In the TIMA status control register (TASC), clear the TIMA stop bit, TSTOP.

Setting MS0B links channels 0 and 1 and configures them for buffered PWM operation. The TIMA channel 0 registers (TACH0H–TACH0L) initially control the buffered PWM output. TIMA status control register 0 (TASC0) controls and monitors the PWM signal from the linked channels. MS0B takes priority over MS0A.

Clearing the toggle-on-overflow bit, TOVx, inhibits output toggles on TIMA overflows. Subsequent output compares try to force the output to a state it is already in and have no effect. The result is a 0% duty cycle output.

Setting the channel x maximum duty cycle bit (CHxMAX) and setting the TOVx bit generates a 100% duty cycle output. See [17.8.4 TIMA Channel Status and Control Registers](#).

17.4 Interrupts

These TIMA sources can generate interrupt requests:

- TIMA overflow flag (TOF) — The TOF bit is set when the TIM counter reaches the modulo value programmed in the TIMA counter modulo registers. The TIMA overflow interrupt enable bit, TOIE, enables TIMA overflow CPU interrupt requests. TOF and TOIE are in the TIMA status and control register.
- TIMA channel flags (CH1F–CH0F) — The CHxF bit is set when an input capture or output compare occurs on channel x. Channel x TIMA CPU interrupt requests are controlled by the channel x interrupt enable bit, CHxIE.

18.8 I/O Registers

These I/O registers control and monitor TIMB operation:

- TIMB status and control register, TBSC
- TIMB control registers, TBCNTH–TBCNTL
- TIMB counter modulo registers, TBMODH–TBMODL
- TIMB channel status and control registers, TBSC0 and TBSC1
- TIMB channel registers, TBCH0H–TBCH0L and TBCH1H–TBCH1L

18.8.1 TIMB Status and Control Register

The TIMB status and control register:

- Enables TIMB overflow interrupts
- Flags TIMB overflows
- Stops the TIMB counter
- Resets the TIMB counter
- Prescales the TIMB counter clock

Address: \$002B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOF	TOIE	TSTOP	0	R	PS2	PS1	PS0
Write:	0			TRST				
Reset:	0	0	1	0	0	0	0	0

R = Reserved

Figure 18-4. TIMB Status and Control Register (TBSC)

TOF — TIMB Overflow Flag Bit

This read/write flag is set when the TIMB counter reaches the modulo value programmed in the TIMB counter modulo registers. Clear TOF by reading the TIMB status and control register when TOF is set and then writing a 0 to TOF. If another TIMB overflow occurs before the clearing sequence is complete, then writing 0 to TOF has no effect. Therefore, a TOF interrupt request cannot be lost due to inadvertent clearing of TOF. Reset clears the TOF bit. Writing a 1 to TOF has no effect.

1 = TIMB counter has reached modulo value

0 = TIMB counter has not reached modulo value

TOIE — TIMB Overflow Interrupt Enable Bit

This read/write bit enables TIMB overflow interrupts when the TOF bit becomes set. Reset clears the TOIE bit.

1 = TIMB overflow interrupts enabled

0 = TIMB overflow interrupts disabled

MSxB — Mode Select Bit B

This read/write bit selects buffered output compare/PWM operation. MSxB exists only in the TIMB channel 0.

Setting MS0B disables the channel 1 status and control register and reverts TBCH1 to general-purpose I/O.

Reset clears the MSxB bit.

- 1 = Buffered output compare/PWM operation enabled
- 0 = Buffered output compare/PWM operation disabled

MSxA — Mode Select Bit A

When ELSxB:A ≠ 00, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation. See [Table 18-2](#).

- 1 = Unbuffered output compare/PWM operation
- 0 = Input capture operation

When ELSxB:A = 00, this read/write bit selects the initial output level of the TCHx pin once PWM, input capture, or output compare operation is enabled. See [Table 18-2](#). Reset clears the MSxA bit.

- 1 = Initial output level low
- 0 = Initial output level high

NOTE

Before changing a channel function by writing to the MSxB or MSxA bit, set the TSTOP and TRST bits in the TIMB status and control register (TBSC).

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 B, and pin PTBx/TBCHx is available as a general-purpose I/O pin. However, channel x is at a state determined by these bits and becomes transparent to the respective pin when PWM, input capture, or output compare mode is enabled. [Table 18-2](#) shows how ELSxB and ELSxA work. Reset clears the ELSxB and ELSxA bits.

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

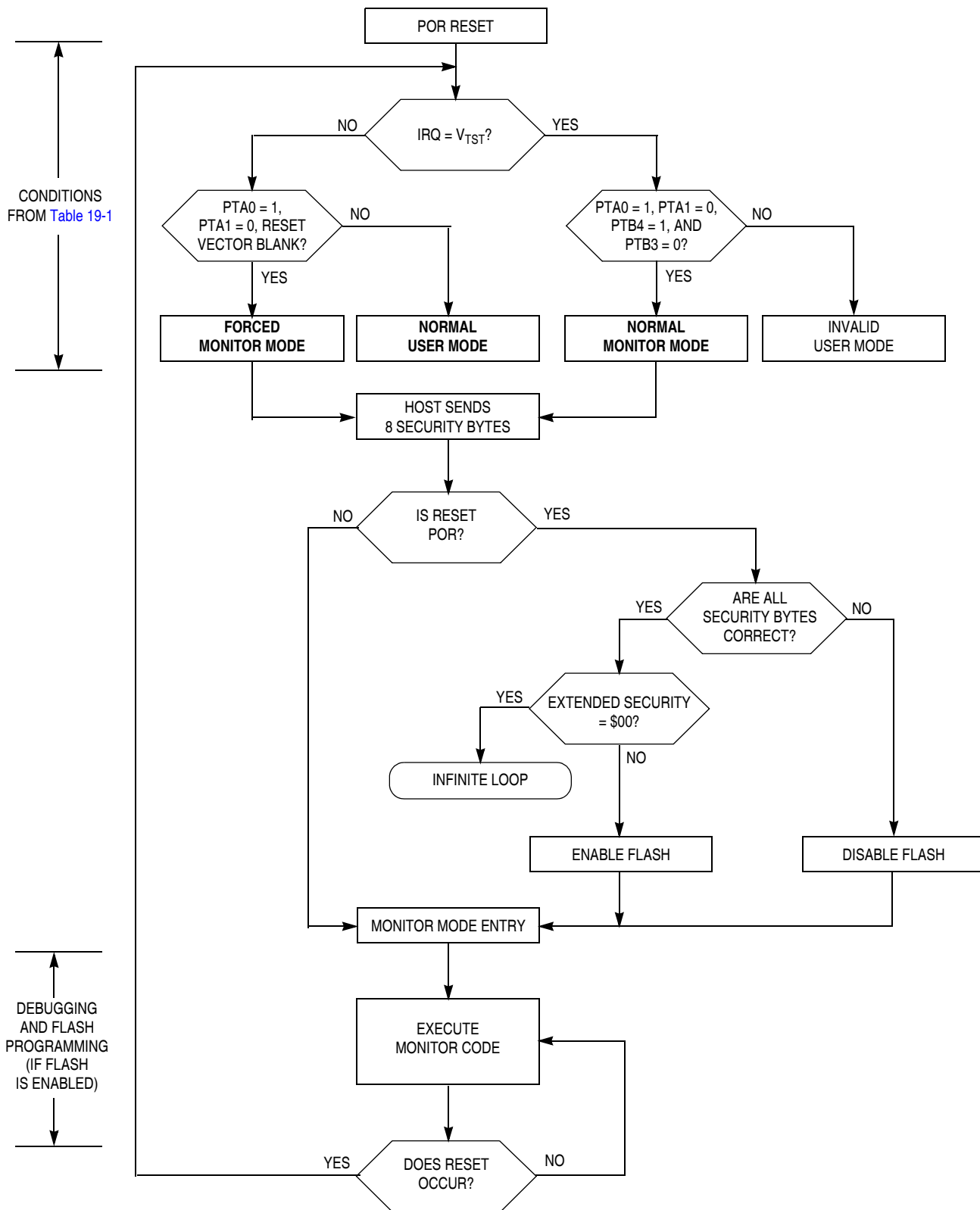


Figure 19-8. Simplified Monitor Mode Entry Flowchart

20.6 5V Control Timing

Characteristic ⁽¹⁾	Symbol	Min	Max	Unit
Frequency of operation ⁽²⁾ Crystal option (EXTSLOW = 1, RNGSEL = 0) Crystal option (EXTSLOW = 0, RNGSEL = 1) Crystal option (EXTSLOW = 0, RNGSEL = 0) External clock option (EXTSLOW = 1, RNGSEL = 0) ⁽³⁾ External clock option (EXTSLOW = 0, RNGSEL = 1) External clock option (EXTSLOW = 0, RNGSEL = 0)	f_{OSC}	32 1 8 dc ⁽⁴⁾ 1 8	100 8 32 100 8 32	kHz MHz MHz kHz MHz MHz
Internal operating frequency		f_{OP}	—	8 MHz
Internal clock period (1/ f_{OP})		t_{cyc}	125	— ns
\overline{RST} input pulse width low ⁽⁵⁾		t_{IRL}	50	— ns
\overline{IRQ} interrupt pulse width low ⁽⁶⁾ (edge-triggered)		t_{ILIH}	50	— ns
\overline{IRQ} interrupt pulse period		t_{ILIL}	Note 8	— t_{cyc}
16-bit timer ⁽⁷⁾ Input capture pulse width Input capture period	t_{TH}, t_{TL} t_{TLTL}	Note 8	— —	ns t_{cyc}

1. $V_{SS} = 0$ Vdc; timing shown with respect to 20% V_{DD} and 70% V_{SS} unless otherwise noted.

2. See [Chapter 8 Internal Clock Generator \(ICG\) Module](#) for more information.

3. No more than 10% duty cycle deviation from 50%

4. Some modules may require a minimum frequency greater than dc for proper operation. See appropriate table for this information.

5. Minimum pulse width reset is guaranteed to be recognized. It is possible for a smaller pulse width to cause a reset.

6. Minimum pulse width is for guaranteed interrupt. It is possible for a smaller pulse width to be recognized.

7. Minimum pulse width is for guaranteed interrupt. It is possible for a smaller pulse width to be recognized.

8. The minimum period, t_{ILIL} or t_{TLTL} , should not be less than the number of cycles it takes to execute the interrupt service routine plus t_{cyc} .

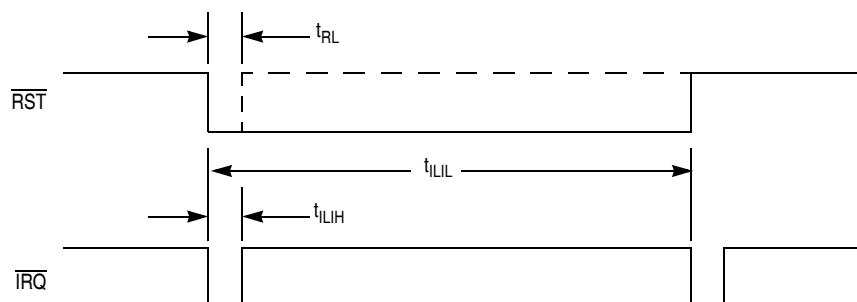


Figure 20-1. \overline{RST} and \overline{IRQ} Timing

B.4.2 Program

The existing 908EY16 routine uses the row programming method to program any range of addresses (starting address in H:X and ending address in RAM at LADDR, data to be programmed is in RAM starting after LADDR). This routine is interrupted every 6 bytes to service the COP.

The 908EY16A routine uses the same variables, so the call setup would be the same. In the 908EY16A, the COPD bit is checked. If COP servicing is required, then a byte-by-byte algorithm is used to program the range. If no COP servicing is required, then a combination of byte-by-byte and row programming is used for a faster algorithm. (In fact, the fastest programming would occur if the start address is the start of a row and the end address is the end of the same row. Then only the faster row programming method would be used.) To this point, the routines are compatible. However, there is a range limitation on this second algorithm. If a range is passed that crosses an 0xFF to 0x00 boundary, then it will fail. The byte-by-byte algorithm does not have this limitation.

Table B-1. Programming Routine Comparison

Routine	908EY16	908EY16A
Page erase	I bit set on exit	I bit restored to value at time of call
Mass erase	Selected with CTRBYT	Selected with ADDR = FLBPR
Program	No restrictions on range to be programmed	If COPD = 1, range cannot include 0xFF/0x00 boundary

The maximum number of bytes required for the stack for all FLASH programming routines is now 13 bytes. In the 908EY16, the maximum number of stack locations was 12 bytes. The user should check to verify that enough stack space is set aside to accommodate this one byte increase.