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	Not For New Designs
Core Processor	CPU12
Core Size	16-Bit
Speed	8MHz
Connectivity	CANbus, MI Bus, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	68
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 16x8/10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc912d60acpve8

List of Figures

Address	Bit 7	6	5	4	3	2	1	Bit 0	Name
$0119	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0	CIDAR5
$011A	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0	CIDAR6
$011B	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0	CIDAR7
$011C	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0	CIDMR4
$011D	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0	CIDMR5
$011E	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0	CIDMR6
$011F	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0	CIDMR7
$0120–$013C	Unimplemented(4)								Reserved
$013D	0	0	0	0	0	0	PUPCAN	RDPCAN	PCTLCAN
$013E	PCAN7	PCAN6	PCAN5	PCAN4	PCAN3	PCAN2	TxCAN	RxCAN	PORTCAN
$013F	DDCAN7	DDCAN6	DDCAN5	DDCAN4	DDCAN3	DDCAN2	0	0	DDRCAN
$0140–$014F	RECEIVE BUFFER								RxFG
$0150–$015F	TRANSMIT BUFFER 0								Tx0
$0160–$016F	TRANSMIT BUFFER 1								Tx1
$0170–$017F	TRANSMIT BUFFER 2								Tx2
$0180–$01DF	Unimplemented(4)								Reserved
$01E0	Reserved								ATD1CTL0
$01E1	Reserved								ATD1CTL1
$01E2	ADPU	AFFC	ASWAI	DJM	R	R	ASCIE	ASCIF	ATD1CTL2
$01E3	0	0	0	0	S1C	FIFO	FRZ1	FRZ0	ATD1CTL3
$01E4	RES10	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0	ATD1CTL4
$01E5	0	S8C	SCAN	MULT	SC	CC	CB	CA	ATD1CTL5
$01E6	SCF	0	0	0	0	CC2	CC1	CC0	ATD1STAT0
$01E7	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0	ATD1STAT1
$01E8	SAR9	SAR8	SAR7	SAR6	SAR5	SAR4	SAR3	SAR2	ATD1TESTH
$01E9	SAR1	SAR0	RST	TSTOUT	TST3	TST2	TST1	TST0	ATD1TESTL
$01EA–$01EE	0	0	0	0	0	0	0	0	Reserved
$01EF	PAD17	PAD16	PAD15	PAD14	PAD13	PAD12	PAD11	PAD10	PORTAD1

= Reserved or unimplemented bits.

Table 4-1. MC68HC912D60A Register Map (Sheet 8 of 9)

Normal Single-Chip Mode — There are no external address and data buses in this mode. The MCU operates as a stand-alone device and all program and data resources are on-chip. External port pins normally associated with address and data buses can be used for general-purpose I/O.

Normal Expanded Wide Mode — This is a normal mode of operation in which the expanded bus is present with a 16-bit data bus. Ports A and B are used for the 16-bit multiplexed address/data bus.

Normal Expanded Narrow Mode — This is a normal mode of operation in which the expanded bus is present with an 8-bit data bus. Ports A and B are used for the16-bit address bus. Port A is used as the data bus, multiplexed with addresses. In this mode, 16-bit data is presented one byte at a time, the high byte followed by the low byte. The address is automatically incremented on the second cycle.

5.3.2 Special Operating Modes

There are three special operating modes that correspond to normal operating modes. These operating modes are commonly used in factory testing and system development. In addition, there is a special peripheral mode, in which an external master, such as an I.C. tester, can control the on-chip peripherals.

Special Single-Chip Mode — This mode can be used to force the MCU to active BDM mode to allow system debug through the BKGD pin. There are no external address and data buses in this mode. The MCU operates as a stand-alone device and all program and data space are on-chip. External port pins can be used for general-purpose I/O.

Special Expanded Wide Mode — This mode can be used for emulation of normal expanded wide mode and emulation of normal single-chip mode. Ports A and B are used for the 16-bit multiplexed address/data bus.

MMSWAI — Memory Mapping Interface Stop in Wait Control

This bit controls access to the memory mapping interface when in Wait mode.

Normal modes: write anytime; special modes: write never. Read anytime.

0 = Memory mapping interface continues to function during Wait mode.

1 = Memory mapping interface access is shut down during Wait mode.

5.5.2 RAM Mapping

The MC68HC912D60A has 2K byte of fully static RAM that is used for storing instructions, variables, and temporary data during program execution. After reset, RAM addressing begins at location $0000 but can be assigned to any 2K byte boundary within the standard 64K byte address space. Mapping of internal RAM is controlled by five bits in the INITRM register.

After reset, the first 512 bytes of RAM have their access inhibited by the presence of the register address space. After initial MCU configuration, it is recommended to map the register space at location $0800.

	Bit 7	6	5	4	3	2	1	Bit 0
	RAM15	RAM14	RAM13	RAM12	RAM11	0	0	0
RESET:	0	0	0	0	0	0	0	0

INITRM — Initialization of Internal RAM Position Register **$0010**

RAM[15:11] — Internal RAM map position

These bits specify the upper five bits of the 16-bit RAM address.

Normal modes: write once; special modes: write anytime. Read anytime.

program/erase voltage. Programming voltage is derived from the internal V_{DD} supply with an internal charge pump.

8.3 EEPROM Selective Write More Zeros

The EEPROM can be programmed such that one or multiple bits are programmed (written to a logic “0”) at a time. However, the user should never program any bit more than once before erasing the entire byte. In other words, the user is not allowed to over write a logic “0” with another “0’.

For some applications it may be advantageous to track more than 10k events with a single byte of EEPROM by programming one bit at a time. For that purpose, a special selective bit programming technique is available. An example is shown here.

Original state of byte = binary 1111:1111 (erased)

First event is recorded by programming bit position 0
Program write = binary 1111:1110; Result = binary 1111:1110

Second event is recorded by programming bit position 1
Program write = binary 1111:1101; Result = binary 1111:1100

Third event is recorded by programming bit position 2
Program write = binary 1111:1011; Result = binary 1111:1000

Fourth event is recorded by programming bit position 3
Program write = binary 1111:0111; Result = binary 1111:0000

Events five through eight are recorded in a similar fashion.

Note that none of the bit locations are actually programmed more than once although the byte was programmed eight times.

When this technique is utilized, a program / erase cycle is defined as multiple writes (up to eight) to a unique location followed by a single erase sequence.

required to complete the instruction. Some of the longer instructions can be interrupted and will resume normally after servicing the interrupt.

When the CPU begins to service an interrupt, the instruction queue is cleared, the return address is calculated, and then it and the contents of the CPU registers are stacked as shown in Table 9-2.

Table 9-2. Stacking Order on Entry to Interrupts

Memory Location	CPU Registers
SP – 2	RTN_H : RTN_L
SP – 4	Y_H : Y_L
SP – 6	X_H : X_L
SP – 8	B : A
SP – 9	CCR

After the CCR is stacked, the I bit (and the X bit, if an $\overline{\text{XIRQ}}$ interrupt service request is pending) is set to prevent other interrupts from disrupting the interrupt service routine. The interrupt vector for the highest priority source that was pending at the beginning of the interrupt sequence is fetched, and execution continues at the referenced location. At the end of the interrupt service routine, an RTI instruction restores the content of all registers from information on the stack, and normal program execution resumes.

If another interrupt is pending at the end of an interrupt service routine, the register unstacking and restacking is bypassed and the vector of the interrupt is fetched.

9.9 Customer Information

Before disabling an interrupt using a local interrupt control bit, set the I mask bit in the CCR. Failing to do so may cause an SWI interrupt to be fetched instead of the vector for the interrupt source that was disabled.

VCO clock at its minimum frequency, f_{VCOMIN}, is provided as the system clock, allowing the MCU to continue operating.

The MCU is said to be operating in **“limp-home” mode** with the forced VCO clock as the system clock. PLLON and BCSP (‘bus clock select PLL’) signals are forced high and the MCS (‘module clock select’) signal is forced low. The LHOME flag in the PLLFLG register is set to indicate that the MCU is running in limp-home mode. A change of this flag sets the limp-home interrupt flag, LHIF, and if enabled by the LHIE bit, the limp-home mode interrupt is requested. The Clock Monitor is enabled irrespective of CME and FCME bit settings. Module clocks to the RTI & COP (XCLK), BDM (BCLK) and ECT & SCI (MCLK) are forced to be PCLK (at f_{VCOMIN}) and ECLK is also equal to f_{VCOMIN}. MSCAN clock select is unaffected.

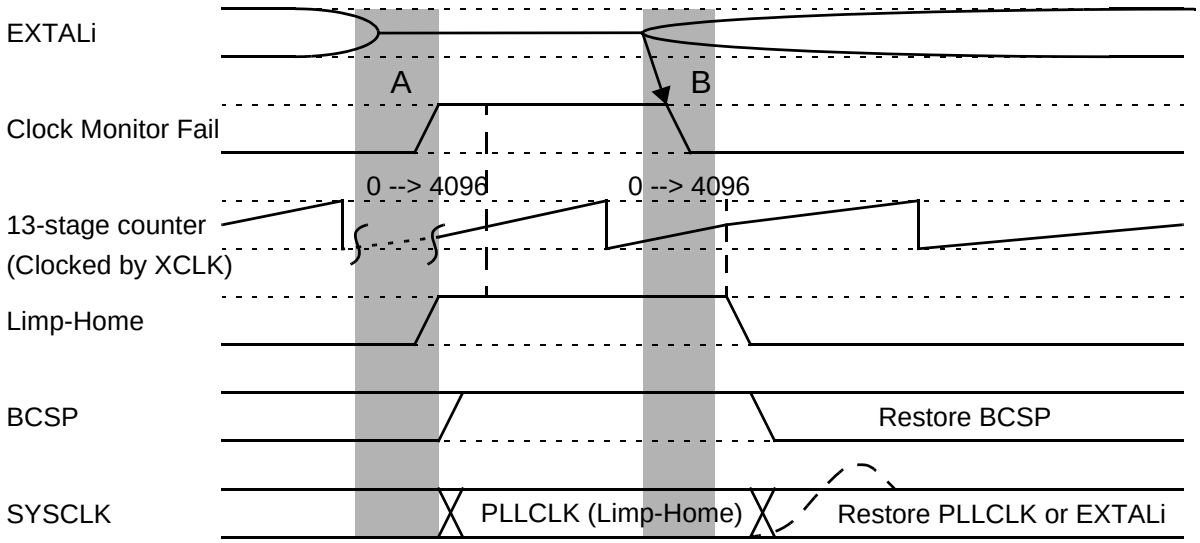

Figure 11-3. Clock Loss during Normal Operation

The clock monitor is polled each time the 13-stage free running counter reaches a count of 4096 XCLK cycles i.e. mid-count, hence the clock status gets checked once every 8192 XCLK cycles. When the presence of an external clock is detected, the MCU exits limp-home mode, clearing the LHOME flag and setting the limp-home interrupt flag. Upon leaving limp-home mode, BCSP and MCS signals are restored to their

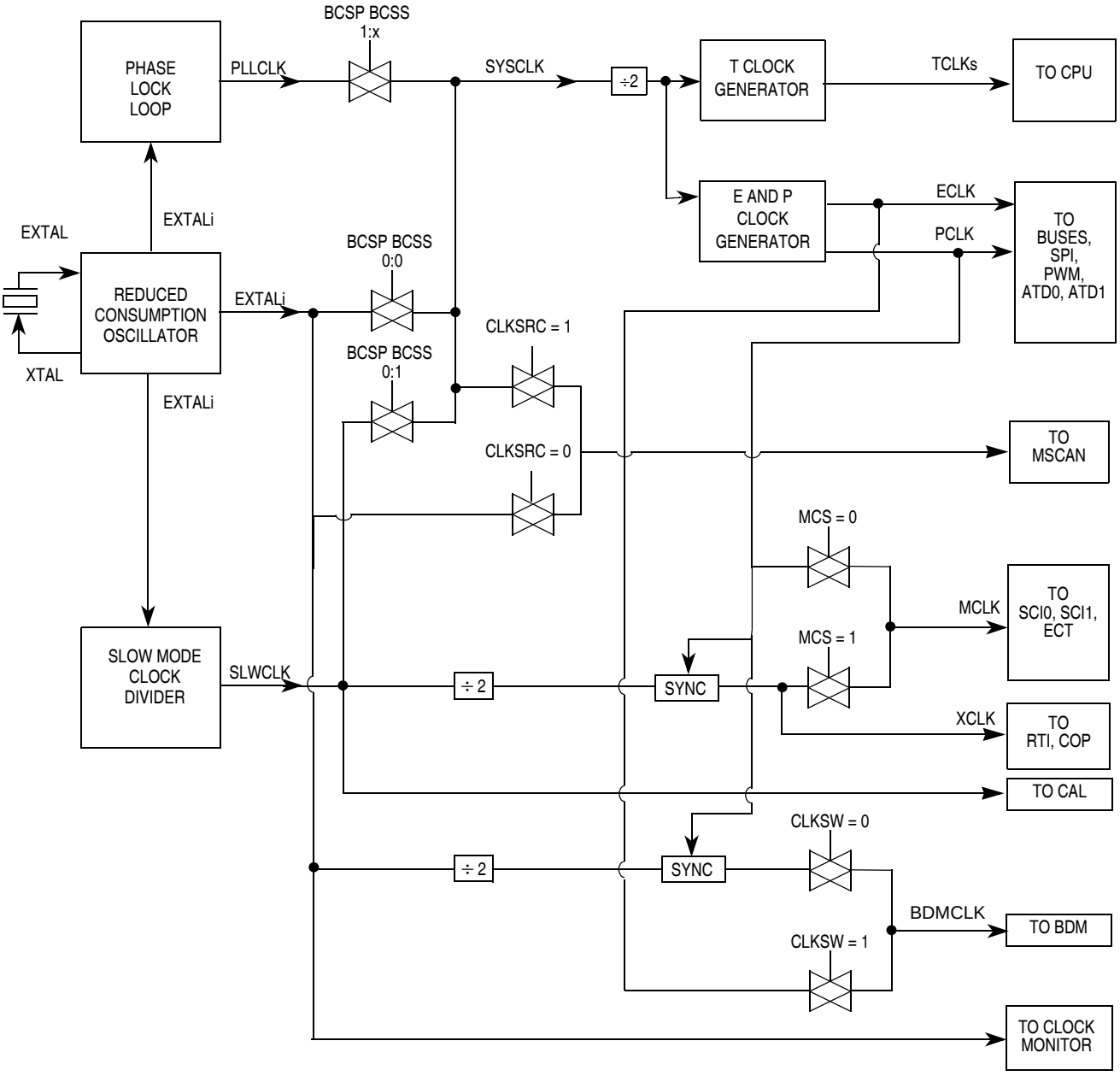

Figure 11-6. Clock Generation Chain

Bus clock select bits BCSP and BCSS in the clock select register (CLKSEL) determine which clock drives SYSCLK for the main system including the CPU and buses. BCSS has no effect if BCSP is set. During

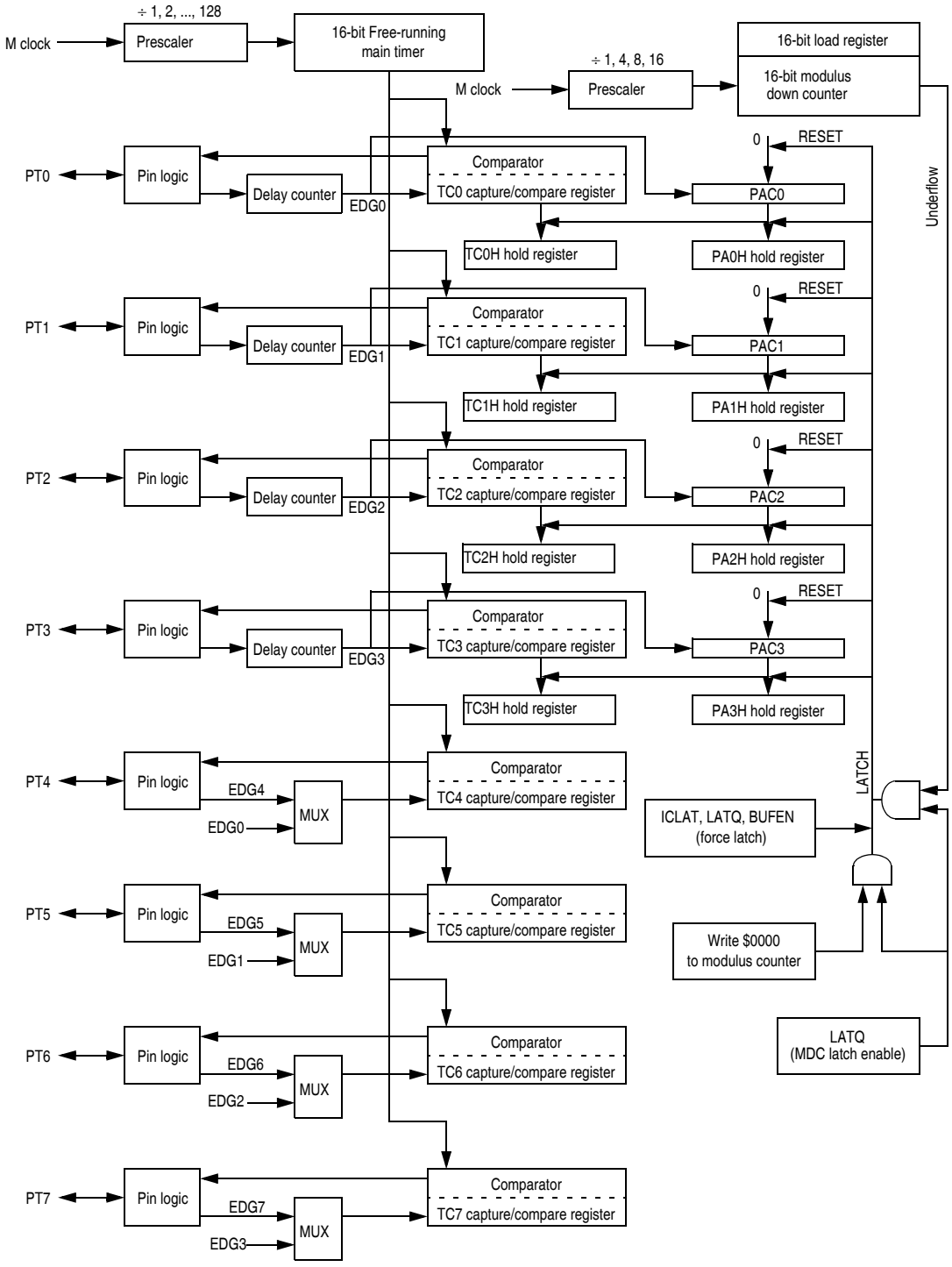

Figure 14-1. Timer Block Diagram in Latch Mode

Table 14-1. Compare Result Output Action

OMn	OLn	Action
0	0	Timer disconnected from output pin logic
0	1	Toggle OCn output line
1	0	Clear OCn output line to zero
1	1	Set OCn output line to one

To operate the 16-bit pulse accumulators A and B (PACA and PACB) independently of input capture or output compare 7 and 0 respectively the user must set the corresponding bits IOSn = 1, OMn = 0 and OLn = 0. OC7M7 or OC7M0 in the OC7M register must also be cleared.

	Bit 7	6	5	4	3	2	1	Bit 0
	EDG7B	EDG7A	EDG6B	EDG6A	EDG5B	EDG5A	EDG4B	EDG4A
RESET:	0	0	0	0	0	0	0	0

TCTL3 — Timer Control Register 3 **$008A**

	Bit 7	6	5	4	3	2	1	Bit 0
	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
RESET:	0	0	0	0	0	0	0	0

TCTL4 — Timer Control Register 4 **$008B**

Read or write anytime.

EDGnB, EDGnA — Input Capture Edge Control

These eight pairs of control bits configure the input capture edge detector circuits.

Table 14-2. Edge Detector Circuit Configuration

EDGnB	EDGnA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge (rising or falling)

	BIT 7	6	5	4	3	2	1	BIT 0
	0	0	0	0	0	0	DLY1	DLY0
RESET:	0	0	0	0	0	0	0	0

DLYCT — Delay Counter Control Register **$00A9**

Read: any time
Write: any time

If enabled, after detection of a valid edge on input capture pin, the delay counter counts the pre-selected number of M clock (module clock) cycles, then it will generate a pulse on its output. The pulse is generated only if the level of input signal, after the preset delay, is the opposite of the level before the transition.This will avoid reaction to narrow input pulses.

After counting, the counter will be cleared automatically.

Delay between two active edges of the input signal period should be longer than the selected counter delay.

DLYx — Delay Counter Select

DLY1	DLY0	Delay
0	0	Disabled (bypassed)
0	1	256 M clock cycles
1	0	512 M clock cycles
1	1	1024 M clock cycles

	BIT 7	6	5	4	3	2	1	BIT 0
	NOVW7	NOVW6	NOVW5	NOVW4	NOVW3	NOVW2	NOVW1	NOVW0
RESET:	0	0	0	0	0	0	0	0

ICOVW — Input Control Overwrite Register **$00AA**

Read: any time
Write: any time

1 = Pulse Accumulator B system enabled. The two 8-bit pulse accumulators PAC1 and PAC0 are cascaded to form the PACB 16-bit pulse accumulator. When PACB in enabled, the PACN1 and PACN0 registers contents are respectively the high and low byte of the PACB.
PA1EN and PA0EN control bits in ICPACR ($A8) have no effect.

PBEN is independent from TEN. With timer disabled, the pulse accumulator can still function unless pulse accumulator is disabled.

PBOVI — Pulse Accumulator B Overflow Interrupt enable
0 = interrupt inhibited
1 = interrupt requested if PBOVF is set

	BIT 7	6	5	4	3	2	1	BIT 0
	0	0	0	0	0	0	PBOVF	0
RESET:	0	0	0	0	0	0	0	0

PBFLG — Pulse Accumulator B Flag Register **$00B1**

Read: any time

Write: any time

PBOVF — Pulse Accumulator B Overflow Flag

This bit is set when the 16-bit pulse accumulator B overflows from $FFFF to $0000, or when 8-bit pulse accumulator 1 (PAC1) overflows from $FF to $00.

This bit is cleared by a write to the PBFLG register with bit 1 set.

Any access to the PACN1 and PACN0 registers will clear the PBOVF flag in this register when TFFCA bit in register TSCR($86) is set.

	Bit 7	6	5	4	3	2	1	Bit 0
	0	RDPS2	RDPS1	RDPS0	0	PUPS2	PUPS1	PUPS0
RESET:	0	0	0	0	0	0	0	0

PURDS — Pull-Up Register for Port S **$00D9**

RDPS2 — Reduce Drive of Port S[7:4]
0 = Port S[7:4] output drivers operate normally
1 = Port S[7:4] output pins have reduced drive capability for lower power and less noise

RDPS1 — Reduce Drive of Port S[3:2]
0 = Port S[3:2] output drivers operate normally
1 = Port S[3:2] output pins have reduced drive capability for lower power and less noise

RDPS0 — Reduce Drive of Port S[1:0]
0 = Port S[1:0] output drivers operate normally
1 = Port S[1:0] output pins have reduced drive capability for lower power and less noise

PUPS2 — Pull-up Port S[7:4] Enable
0 = No internal pull-ups on port S[7:4]
1 = Port S[7:4] input pins have an active pull-up device. If a pin is programmed as output, the pull-up device becomes inactive.

PUPS1 — Pull-up Port S[3:2] Enable
0 = No internal pull-ups on port S[3:2]
1 = Port S[3:2] input pins have an active pull-up device. If a pin is programmed as output, the pull-up device becomes inactive.

PUPS0 — Pull-up Port S[1:0] Enable
0 = No internal pull-ups on port S[1:0]
1 = Port S[1:0] input pins have an active pull-up device. If a pin is programmed as output, the pull-up device becomes inactive.

17.7 Protocol Violation Protection

The msCAN12 will protect the user from accidentally violating the CAN protocol through programming errors. The protection logic implements the following features:

- The receive and transmit error counters cannot be written or otherwise manipulated.
- All registers which control the configuration of the msCAN12 cannot be modified while the msCAN12 is on-line. The SFTRES bit in CMCR0 (see msCAN12 Module Control Register 0 (CMCR0)) serves as a lock to protect the following registers:
  - msCAN12 module control register 1 (CMCR1)
  - msCAN12 bus timing register 0 and 1 (CBTR0, CBTR1)
  - msCAN12 identifier acceptance control register (CIDAC)
  - msCAN12 identifier acceptance registers (CIDAR0–7)
  - msCAN12 identifier mask registers (CIDMR0–7)
- The TxCAN pin is forced to recessive when the msCAN12 is in any of the low power modes.

17.8 Low Power Modes

In addition to normal mode, the msCAN12 has three modes with reduced power consumption: SLEEP, SOFT_RESET and POWER_DOWN mode. In SLEEP and SOFT_RESET modes, power consumption is reduced by stopping all clocks except those to access the registers. In POWER_DOWN mode, all clocks are stopped and no power is consumed.

The WAI and STOP instructions put the MCU in low power consumption stand-by modes. Table 17-2 summarizes the combinations of msCAN12 and CPU modes. A particular combination of modes is entered for the given settings of the bits CSWAI, SLPAK, and SFTRES. For all modes, an msCAN wake-up interrupt can occur only if SLPAK=WUPIE=1. While the CPU is in Wait Mode, the msCAN12 can be operated in Normal

18.3 Modes of Operation

Analog to digital conversions are performed in a variety of different programmable sequences referred to as conversion modes. Each conversion mode is defined by:

- How many A/D conversions (one, four or eight) are performed in a sequence
- Which analog input channels are examined during a sequence
- The sample time length
- Whether sequences are performed continuously or not
- Result register assignments

The modes are defined by the settings of three control bits (in ATDCTL5)

- MULT controls whether the sequence examines a single analog input channel or scans a number of different channels
- SCAN determines if sequences are performed continuously
- SC determines if we are performing a special conversion i.e. converting V_{RL}, V_{RH}, $(V_{RL}+V_{RH})/2$ (usually used for test purposes).

and three control values

- CC/CB/CA (in ATDCTL5) define the input channel(s) to be examined
- S8C/S1C (in ATDCTL3/5) define the number of conversions in a sequence
- SMP0/SMP1 (in ATDCTL4) define the length of the sample time.

Sequences are initiated or halted by writing to control registers ATDCTL4 and ATDCTL5.

For the continuous sequence modes, conversions will not stop until

- Another non-continuous conversion sequence is initiated and finishes
- The ATD is powered down (ADPU control bit)
- The ATD is reset

This bit provides program on/off control over the ATD module allowing reduced MCU power consumption when the ATD is not being used. When reset to zero, the ADPU bit aborts any conversion sequence in progress. Because the analog electronics is turned off when powered down, the ATD requires a recovery time period when ADPU bit is enabled.

AFFC — ATD Fast Conversion Complete Flag Clear

0 = ATD flag clearing operates normally (read the status register before reading the result register to clear the associated CCF bit).

1 = Changes all ATD conversion complete flags to a fast clear sequence. Any access to a result register (ATD0–7) will cause the associated CCF flag to clear automatically if it was set at the time.

Operating normally means that the status register must be read after the conversion complete flag has been set before that flag can be reset. After the status register read, a read to the associated result register causes its conversion complete flag in the status register to be cleared. The SCF flag is cleared when a new conversion sequence is begun by writing to control register ATDCTL4/5. In applications where the ATD module is polled to determine if an ATD conversion is complete, this feature provides a convenient way of clearing the status register conversion complete flag.

In applications where ATD interrupts are used to signal conversion completion, the precondition of reading the status register can be eliminated using fast conversion complete flag clear mode. In this mode, any access to a result register will cause its associated conversion complete flag in the status register to be cleared. The SCF flag is cleared after the first (any) result register is read.

ASWAI — ATD Stop In Wait Mode

0 = ATD continues to run when the MCU is in wait mode

1 = ATD stops to save power when the MCU is in wait mode

The wait function allows the MCU to selectively halt and power down the ATD module. If the ASWAI bit is set and the MCU, then the ATD module immediately halts operation and powers down. When WAIT is

Table 20-8. ATD Maximum Ratings

Characteristic	Symbol	Value	Units
ATD reference voltage $V_{RH} \leq V_{DDA}$ $V_{RL} \geq V_{SSA}$	 V_{RH} V_{RL}	 –0.3 to +6.5 –0.3 to +6.5	 V V
V_{SS} differential voltage	$\lvert V_{SS}–V_{SSA} \rvert$	0.1	V
V_{DD} differential voltage	$V_{DD}–V_{DDA}$ $V_{DDA}–V_{DD}$	6.5 0.3	V V
Reference to supply differential voltage	$V_{DDA}–V_{RH}$ $V_{RH}–V_{DDA}$ $V_{DDA}–V_{RL}$ $V_{RL}–V_{DDA}$	6.5 0.3 6.5 0.3	V
Analog input differential voltage	$V_{DDA}–V_{INDC}$ $V_{INDC}–V_{DDA}$	6.5 0.3	V

Table 20-9. EEPROM Characteristics

V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted

Characteristic	Symbol	Min	Max	Unit
Minimum programming clock frequency	f_{PROG}	250K		hz
Programming time	t_{PROG}		10	ms
Clock recovery time, following STOP, to continue programming	t_{CRSTOP}		t_{PROG}+ 1	ms
Erase time	t_{ERASE}		10	ms
Write/erase endurance		10,000		cycles
Data retention		10(1)		years
EEPROM Programming Maximum Time to ‘AUTO’ Bit Set	—	—	500	µs
EEPROM Erasing Maximum Time to ‘AUTO’ Bit Set	—	—	10	ms

1. Based on the average life time operating temperature of 70°C.

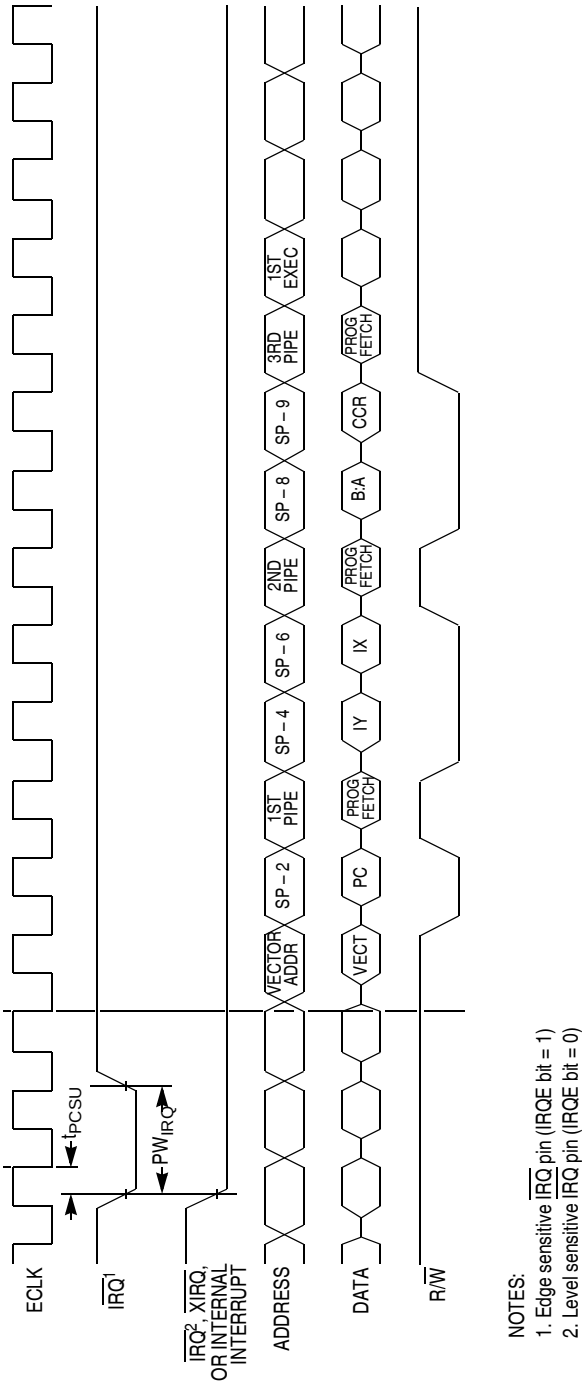

Figure 20-5. Interrupt Timing Diagram