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	CPU12
Core Size	16-Bit
Speed	8MHz
Connectivity	SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	63
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	768 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/xc912bc32cfue8

Chapter 11
Pulse-Width Modulator (PWM)

Chapter 12
Standard Timer (TIM)

Chapter 1
General Description

1.1 Introduction

The MC68HC912B32, MC68HC12BE32 and MC68HC(9)12BC32, are 16-bit microcontroller units (MCUs) composed of standard on-chip peripherals. The multiplexed external bus can also operate in an 8-bit narrow mode for interfacing with single 8-bit wide memory in lower-cost systems. There is a slight feature set difference between the four pin-for-pin compatible devices as shown in Table 1-1.

Table 1-1. M68HC12B Series Feature Set Comparisons

Features	MC68HC912B32	MC68HC12BE32	MC68HC912BC32	MC68HC12BC32
CPU12	X	X	X	X
Multiplexed bus	X	X	X	X
32-Kbyte FLASH electrically erasable, programmable read-only memory (EEPROM)	X		X	
32-Kbyte read-only memory (ROM)		X		X
768-byte EEPROM	X	X	X	X
1-Kbyte random-access memory (RAM)	X	X	X	X
Analog-to-digital (A/D) converter	X	X	X	X
Standard timer module (TIM)	X		X	X
Enhanced capture timer (ECT)		X		
Pulse-width modulator (PWM)	X	X	X	X
Asynchronous serial communications interface (SCII)	X	X	X	X
Synchronous serial peripheral interface (SPI)	X	X	X	X
J1850 byte data link communication (BDLC)	X	X		
Controller area network module (CAN)			X	X
Computer operating properly (COP) watchdog timer	X	X	X	X
Slow mode clock divider	X	X	X	X
80-pin quad flat pack (QFP)	X	X	X	X
Single-wire background debug mode (BDM)	X	X	X	X

selectable (interrupt control register, INTCR). $\overline{\text{IRQ}}$ is always configured to level-sensitive triggering at reset. When the MCU is reset, the $\overline{\text{IRQ}}$ function is masked in the condition code register.

This pin is always an input and can always be read. In special modes, it can be used to apply external EEPROM V_{PP} in support of EEPROM testing. External V_{PP} is not needed for normal EEPROM program and erase cycles. Because the $\overline{\text{IRQ}}$ pin is also used as an EEPROM programming voltage pin, there is an internal resistive pullup on the pin.

1.6.3.5 $\overline{\text{XIRQ}}$

$\overline{\text{XIRQ}}$ is the non-maskable external interrupt pin. It provides a means of requesting a non-maskable interrupt after reset initialization. During reset, the X bit in the condition code register (CCR) is set and any interrupt is masked until MCU software enables it. Because the $\overline{\text{XIRQ}}$ input is level sensitive, it can be connected to a multiple-source wired-OR network. This pin is always an input and can always be read. There is an active pullup on this pin while in reset and immediately out of reset. The pullup can be turned off by clearing the PUPE bit in the pullup control register (PUCR). $\overline{\text{XIRQ}}$ is often used as a power loss detect interrupt.

When $\overline{\text{XIRQ}}$ or $\overline{\text{IRQ}}$ are used with multiple interrupt sources ($\overline{\text{IRQ}}$ must be configured for level-sensitive operation if there is more than one source of $\overline{\text{IRQ}}$ interrupt), each source must drive the interrupt input with an open-drain type of driver to avoid contention between outputs. There must also be an interlock mechanism at each interrupt source so that the source holds the interrupt line low until the MCU recognizes and acknowledges the interrupt request. If the interrupt line is held low, the MCU recognizes another interrupt as soon as the interrupt mask bit in the MCU is cleared, normally upon return from an interrupt.

1.6.3.6 SMODN, MODA, and MODB

SMODN, MODA, and MODB are the mode-select signals. Their state during reset determines the MCU operating mode. After reset, MODA and MODB can be configured as instruction queue tracking signals IPIPE0 and IPIPE1. MODA and MODB have active pulldowns during reset.

The SMODN pin can be used as BKGD or $\overline{\text{TAGHI}}$ after reset.

NOTE

To aid in mode selection, refer to Figure 1-8 and Figure 1-9. These schematics are provided as suggestive layouts only.

1.6.3.7 BKGD

BKGD is the single-wire background mode pin. It receives and transmits serial background debugging commands. A special self-timing protocol is used. The BKGD pin has an active pullup when configured as input; BKGD has no pullup control. Currently, the tool connection configuration shown in Figure 1-7 is used.

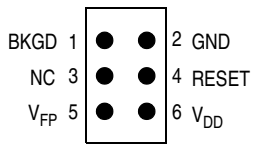


Figure 1-7. BDM Tool Connector

1.6.4.4 Port DLC

The MC68HC912B32 and MC68HC12BE32 contain the port DLC.

Byte data link communications (BDLC) pins can be configured as general-purpose I/O port DLC. When BDLC functions are not enabled, the port has seven general-purpose I/O pins, PDLC6–PDLC0. The port DLC control register (DLCSCR) controls port DLC function. The BDLC function, enabled with the BDLCEN bit, takes precedence over other port functions.

The port DLC data direction register (DDRDLC) determines whether each port DLC pin is an input or output. Setting a bit in DDRDLC makes the corresponding pin in port DLC an output; clearing a bit makes the corresponding pin an input. After reset, port DLC pins are configured as inputs.

When the PUPDLC bit in the DLCSCR register is set, all port DLC input pins are pulled up internally by an active pullup device.

Setting the RDPDLC bit in register DLCSCR causes all port DLC outputs to have reduced drive level. Levels are at normal drive capability after reset. RDPDLC can be written anytime after reset. Refer to Chapter 15 Byte Data Link Communications (BDLC).

1.6.4.5 Port CAN

The MC68HC(9)12BC32 contains the port CAN.

The port CAN has five general-purpose I/O pins, PCAN[6:2]. The msCAN12 receive pin, RxCAN, and transmit pin, TxCAN, cannot be configured as general-purpose I/O on port CAN.

The msCAN data direction register (DDRCAN) determines whether each port CAN pin PCAN[6:2] is an input or output. Setting a bit in DDRCAN makes the corresponding pin in port CAN an output; clearing a bit makes the corresponding pin an input. After reset, port CAN pins PCAN[6:2] are configured as inputs.

When a read to the port CAN is performed, the value read from the most significant bit (MSB) depends on the MSB, PCAN7, of the port CAN data register, PORTCAN, and the MSB of DDRCAN: it is 0 if DDRCAN7 = 0 and is PCAN7 if DDRCAN7 = 1.

When the PEUCAN bit in the port CAN control register (PCTLCAN) is set, port CAN input pins PCAN[6:2] are pulled up internally by an active pullup device.

Setting the RDRCAN bit in register PCTLCAN causes the port CAN outputs PCAN[6:2} to have reduced drive level. Levels are at normal drive capability after reset. RDRCAN can be written anytime after reset. Refer to Chapter 16 msCAN12 Controller.

1.6.4.6 Port AD

Port AD provides input to the analog-to-digital subsystem and general-purpose input. When analog-to-digital functions are not enabled, the port has eight general-purpose input pins, PAD7–PAD0. The ADPU bit in the ATD control register 2 (ATDCTL2) enables the A/D function.

Port AD pins are inputs; no data direction register is associated with this port. The port has no resistive input loads and no reduced drive controls. Refer to Chapter 17 Analog-to-Digital Converter (ATD).

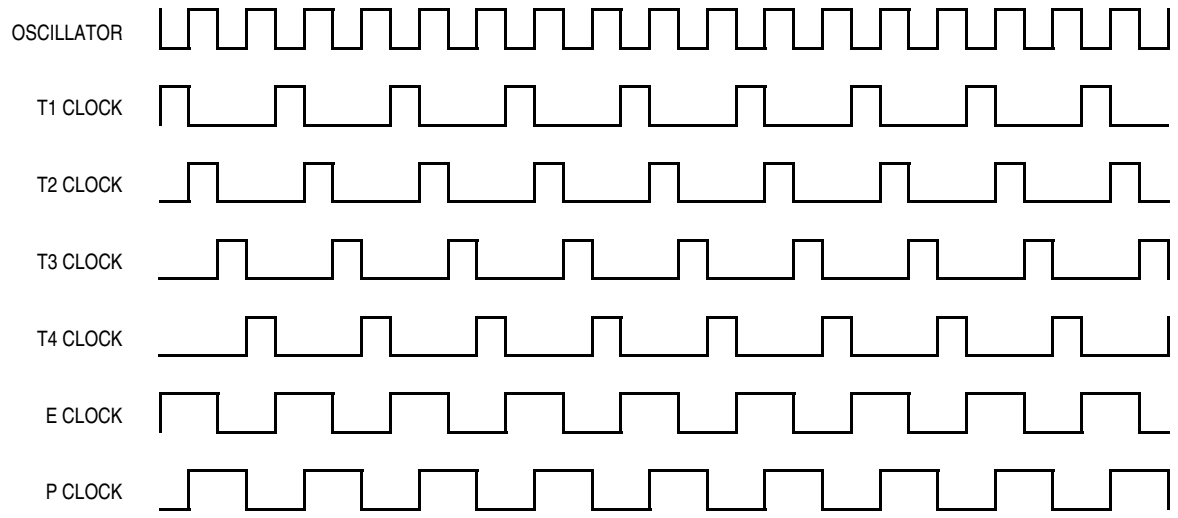
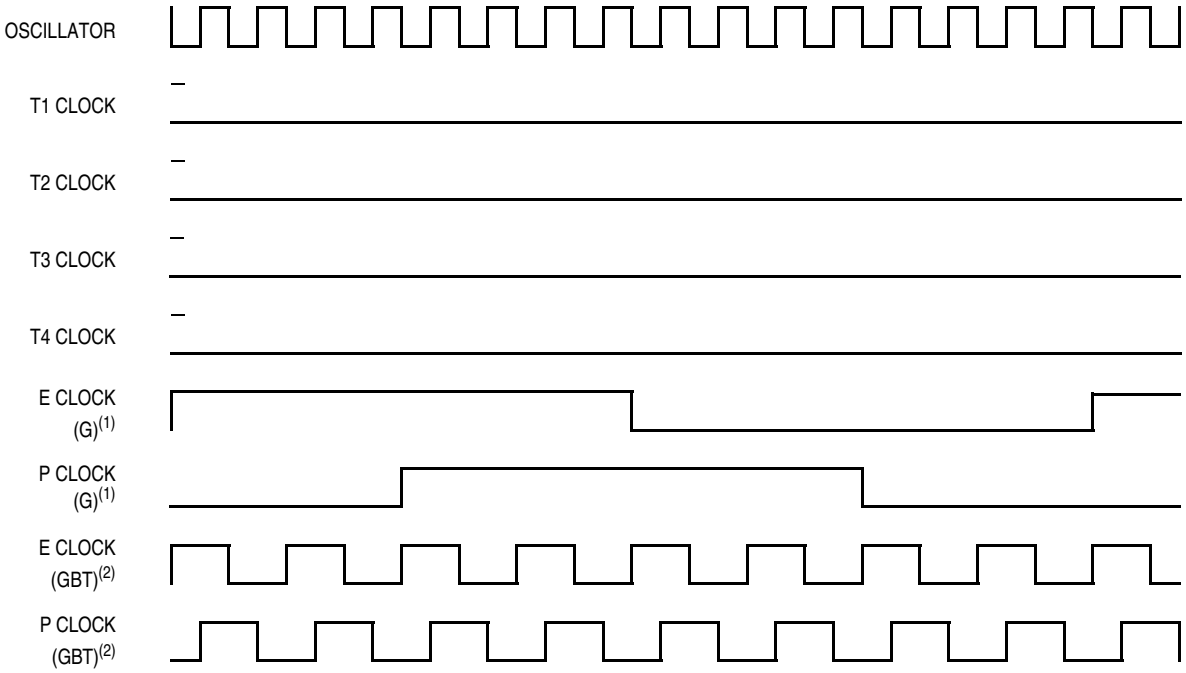


Figure 10-3. Internal Clock Relationships in Normal Run Modes



Notes:
1. Driven by slow clock divider in wait mode. Drives on-chip peripherals except BDLC and timer.
2. Remains at oscillator divided by 2 rate in wait mode. Drives BDLC and timer.

Figure 10-4. Internal Clock Relationships in Wait Mode

10.7.2 Real-Time Interrupt Control Register

Address: $0014

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	RTIE	RSWAI	RSBCK	0	RTBYP	RTR2	RTR1	RTR0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-6. Real-Time Interrupt Control Register (RTICTL)

Read: Anytime

Write: Varies on a bit-by-bit basis

RTIE — Real-Time Interrupt Enable Bit

Write anytime.

0 = Interrupt requests from RTI are disabled.

1 = Interrupt is requested when RTI is set.

RSWAI — RTI and COP Stop While in Wait Bit

Write once in normal modes, anytime in special modes.

0 = Allows the RTI and COP to continue running in wait

1 = Disables both the RTI and COP when the part goes into wait

RSBCK — RTI and COP Stop While in Background Debug Mode Bit

Write once in normal modes, anytime in special modes.

0 = Allows the RTI and COP to continue running while in background mode

1 = Disables RTI and COP when the part is in background mode (useful for emulation)

RTBYP — Real-Time Interrupt Divider Chain Bypass Bit

Write is not allowed in normal modes, anytime in special modes.

0 = Divider chain functions normally.

1 = Divider chain is bypassed, allows faster testing. The divider chain is normally P divided by 2^{13}, when bypass becomes P divided by 4.

RTR2, RTR1, and RTR0 — Real-Time Interrupt Rate Select Bits

Write anytime.

Rate select for real-time interrupt. The E clock is used for this module.

Table 10-3. Real-Time Interrupt Rates

RTR2	RTR1	RTR0	Divide E By:	Timeout Period E = 4.0 MHz	Timeout Period E = 8.0 MHz
0	0	0	OFF	OFF	OFF
0	0	1	2^{13}	2.048 ms	1.024 ms
0	1	0	2^{14}	4.096 ms	2.048 ms
0	1	1	2^{15}	8.196 ms	4.096 ms
1	0	0	2^{16}	16.384 ms	8.196 ms
1	0	1	2^{17}	32.768 ms	16.384 ms
1	1	0	2^{18}	65.536 ms	32.768 ms
1	1	1	2^{19}	131.72 ms	65.536 ms

10.8 Clock Divider Chains

Figure 10-10, Figure 10-11, Figure 10-12, and Figure 10-13 summarize the clock divider chains for these peripherals:

- SCI — Serial peripheral interface
- BDLC — Byte data link communications
- RTI — Real-time interrupt
- COP — Computer operating properly
- TIM — Standard timer module
- ECT — Enhanced capture timer
- SPI — Serial peripheral interface
- ATD — Analog-to-digital converter
- BDM — Background debug mode

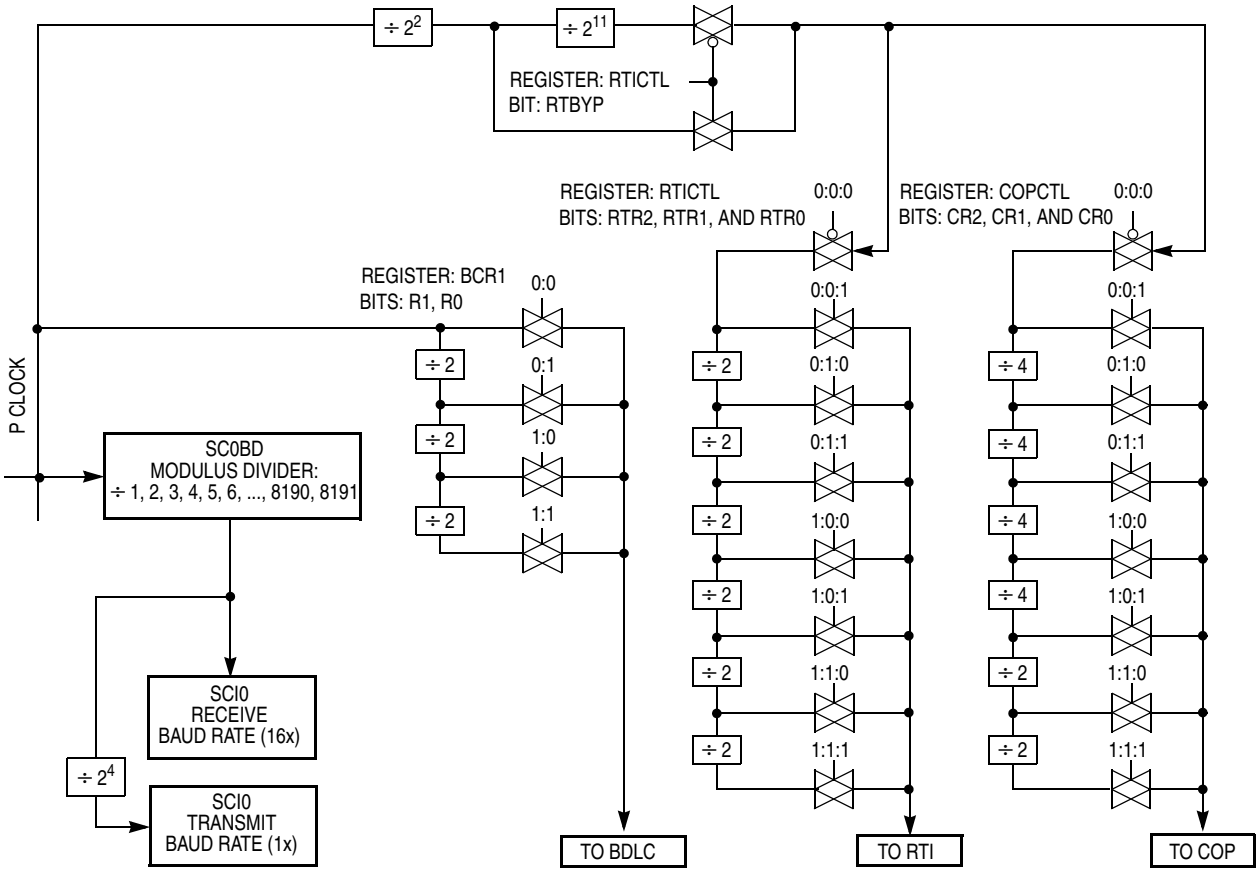


Figure 10-10. Clock Chain for SCI, BDLC, RTI, and COP

C7I–C0I — Input Capture/Output Compare x Interrupt Enable Bits

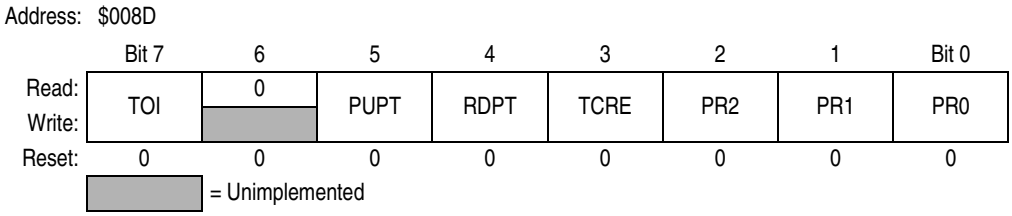


Figure 12-13. Timer Interrupt Mask 2 Register (TMSK2)

Read: Anytime

Write: Anytime

TOI — Timer Overflow Interrupt Enable Bit

0 = Interrupt inhibited
1 = Hardware interrupt requested when TOF flag set

PUPT — Timer Pullup Resistor Enable Bit

This enable bit controls pullup resistors on the timer port pins when the pins are configured as inputs.

0 = Disable pullup resistor function
1 = Enable pullup resistor function

RDPT — Timer Drive Reduction Bit

This bit reduces the effective output driver size which can reduce power supply current and generated noise depending upon pin loading.

0 = Normal output drive capability
1 = Enable output drive reduction function

TCRE — Timer Counter Reset Enable Bit

This bit allows the timer counter to be reset by a successful output compare 7 event.

0 = Counter reset inhibited and counter free runs
1 = Counter reset by a successful output compare 7

If TC7 = $0000 and TCRE = 1, TCNT stays at $0000 continuously. If TC7 = $FFFF and TCRE = 1, TOF never gets set even though TCNT counts from $0000 through $FFFF.

PR2, PR1, and PR0 — Timer Prescaler Select Bits

These three bits specify the number of ÷2 stages that are to be inserted between the module clock and the timer counter. See Table 12-3.

Table 12-3. Prescaler Selection

PR2	PR1	PR0	Prescale Factor
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	Reserved
1	1	1	Reserved

The newly selected prescale factor will not take effect until the next synchronized edge where all prescale counter stages equal 0.

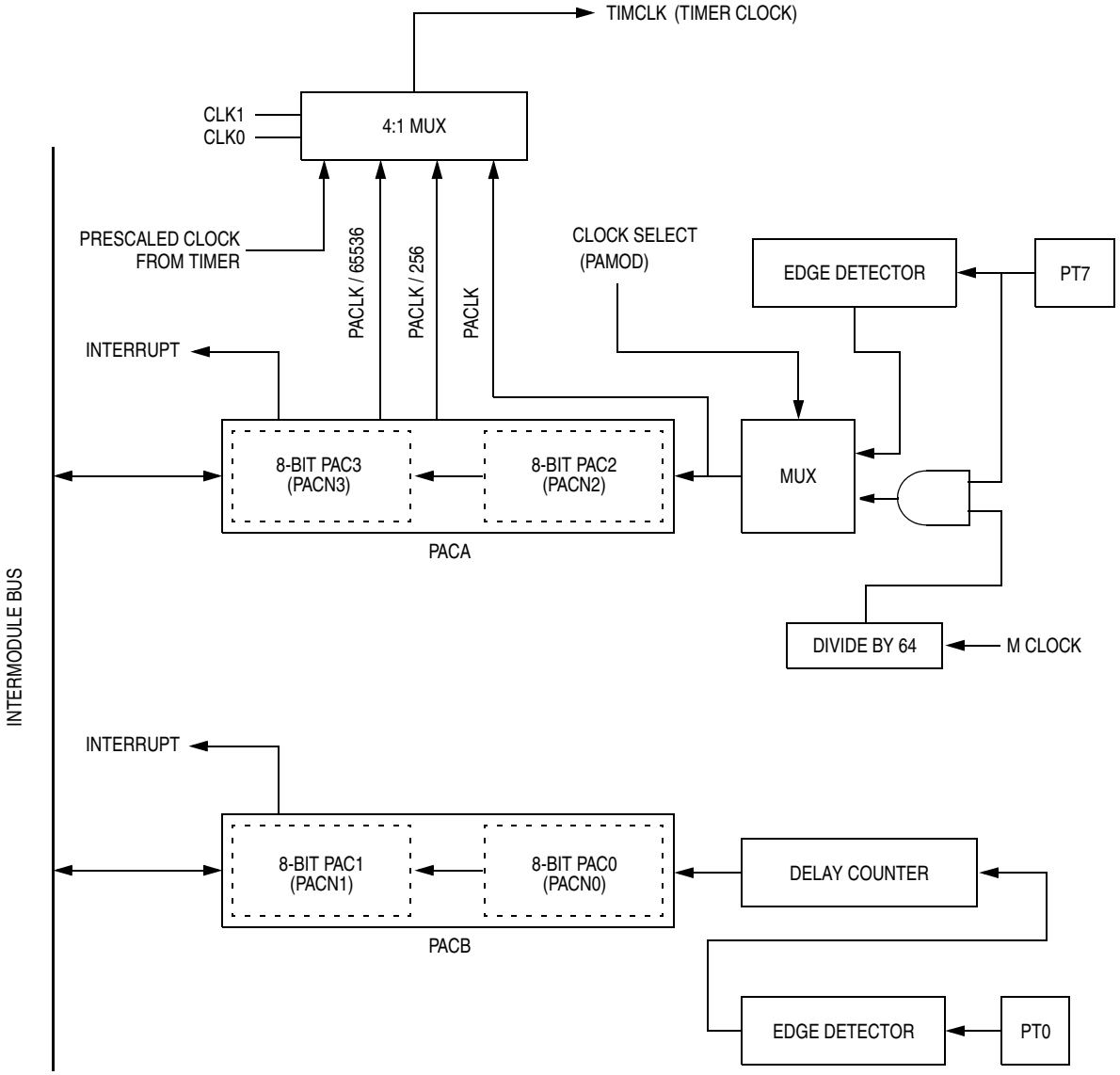


Figure 13-4. 16-Bit Pulse Accumulators Block Diagram

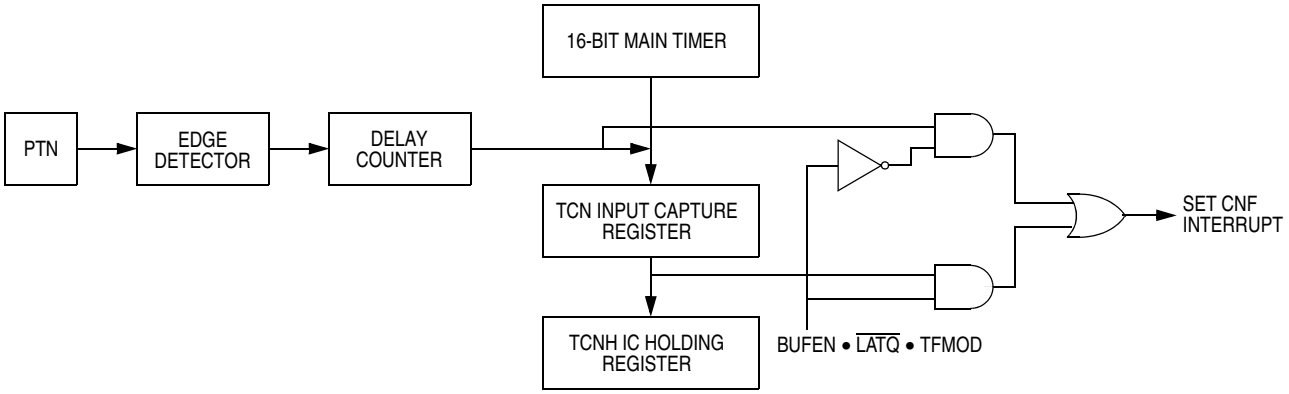


Figure 13-19. C3F–C0F Interrupt Flag Setting

Address: $008F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TOF	0	0	0	0	0	0	0
Reset:	0	0	0	0	0	0	0	0

Figure 13-20. Main Timer Interrupt Flag 2 (TFLG2)

Read: Anytime

Write: Used in clearing mechanism (set bits cause corresponding bits to be cleared). Any access to TCNT will clear the TFLG2 register, if the TFFCA bit in the TSCR register is set.

TFLG2 indicates when interrupt conditions have occurred. To clear a bit in the flag register, set the bit to 1.

TOF — Timer Overflow Flag

TOF is set when the 16-bit free-running timer overflows from $FFFF to $0000. This bit is cleared automatically by a write to the TFLG2 register with bit 7 set. See the explanation of the TCRE control bit in 13.4.8 Timer Interrupt Mask Registers.)

13.4.10 Timer Input Capture/Output Compare Registers

Address: $0090–$0091

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	0	0	0	0	0	0	0	0
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	0	0	0	0	0	0	0	0

Figure 13-21. Timer Input Capture/Output Compare Register 0 (TC0)

RSRC — Receiver Source Bit

When LOOPS = 1, the RSRC bit determines the internal feedback path for the receiver.

0 = Receiver input connected to the transmitter internally
(not TXD pin)
1 = Receiver input connected to the TXD pin

M — Mode Bit (select character format)

0 = One start, eight data, one stop bit
1 = One start, eight data, ninth data, one stop bit

WAKE — Wakeup by Address Mark/Idle Bit

0 = Wakeup by IDLE line recognition
1 = Wakeup by address mark (last data bit set)

ILT — Idle Line Type Bit

This bit determines which of two types of idle line detection is used by the SCI receiver.

0 = Short idle line mode enabled
1 = Long idle line mode detected

In short mode, the SCI circuitry begins counting 1s in the search for the idle line condition immediately after the start bit. This means that the stop bit and any bits that were 1s before the stop bit could be counted in that string of 1s, resulting in earlier recognition of an idle line.

In long mode, the SCI circuitry does not begin counting 1s in the search for the idle line condition until a stop bit is received. Therefore, the last byte's stop bit and preceding 1 bits do not affect how quickly an idle line condition can be detected.

PE — Parity Enable Bit

0 = Parity disabled
1 = Parity enabled

PT — Parity Type Bit

If parity is enabled, this bit determines even or odd parity for both the receiver and the transmitter. An even number of 1s in the data character causes the parity bit to be 0 and an odd number of 1s causes the parity bit to be 1.

0 = Even parity selected
1 = Odd parity selected

OR — Overrun Error Flag

New byte is ready to be transferred from the receive shift register to the receive data register and the receive data register is already full (RDRF bit is set). Data transfer is inhibited until this bit is cleared.

0 = No overrun
1 = Overrun detected

NF — Noise Error Flag

Set during the same cycle as the RDRF bit but not set in the case of an overrun (OR).

0 = Unanimous decision
1 = Noise on a valid start bit, any of the data bits, or on the stop bit

FE — Framing Error Flag

Set when a 0 is detected where a stop bit was expected. Clear the FE flag by reading SC0SR1 with FE set and then reading SC0DR.

0 = Stop bit detected
1 = Zero detected rather than a stop bit

PF — Parity Error Flag

Indicates if received data's parity matches parity bit. This feature is active only when parity is enabled. The type of parity tested for is determined by the PT (parity type) bit in SC0CR1.

0 = Parity correct
1 = Incorrect parity detected

14.2.3.5 SCI Status Register 2

Address: $00C5

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	RAF
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 14-8. SCI Status Register 2 (SC0SR2)

Read: Anytime

Write: Has no meaning or effect

RAF — Receiver Active Flag

This bit is controlled by the receiver front end. It is set during the RT1 time period of the start bit search. It is cleared when an idle state is detected or when the receiver circuitry detects a false start bit (generally due to noise or baud rate mismatch).

0 = Character is not being received.
1 = Character is being received.

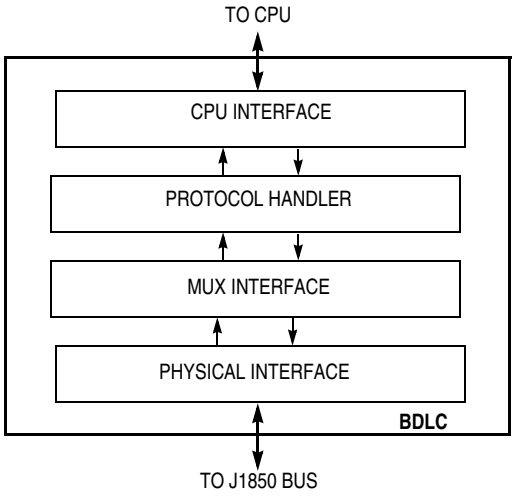


Figure 15-1. BDLC Block Diagram

15.4 BDLC Operating Modes

The BDLC has five main modes of operation which interact with the power supplies, pins, and rest of the MCU as shown in Figure 15-2.

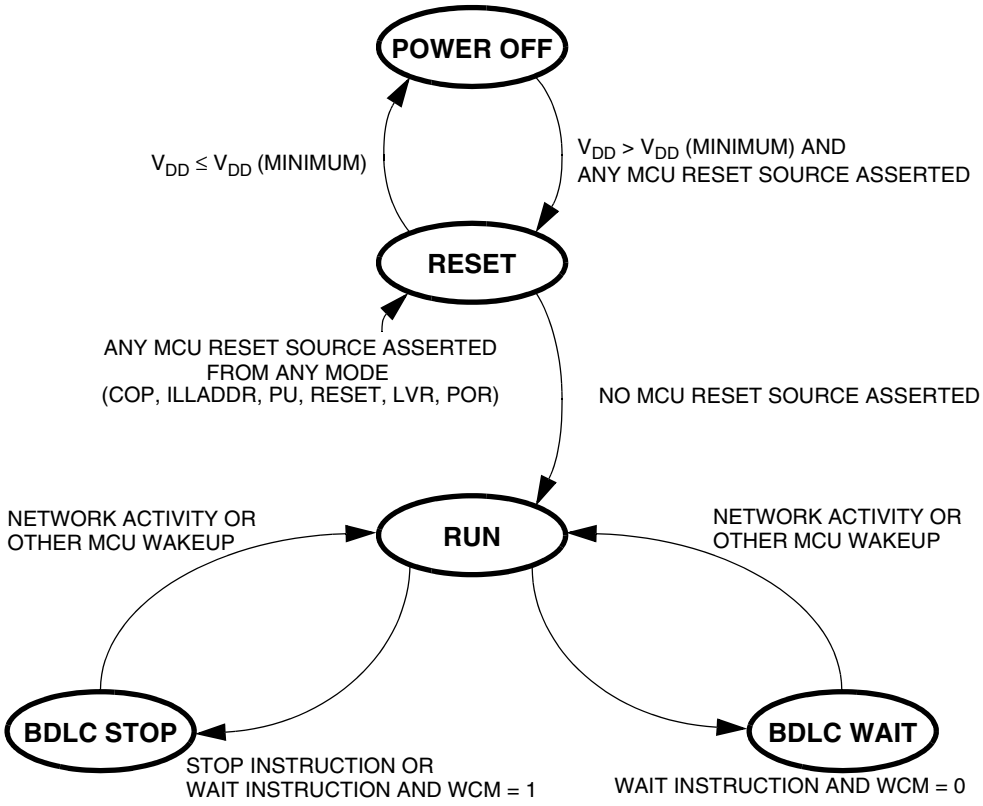


Figure 15-2. BDLC Operating Modes State Diagram

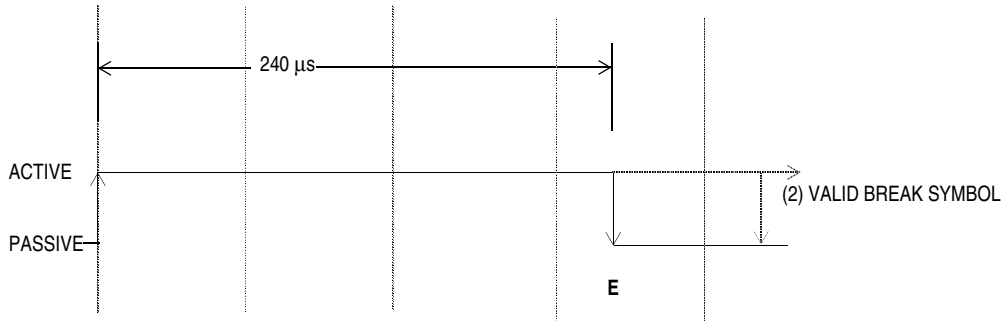


Figure 15-9. J1850 VPW Received BREAK Symbol Times

15.7.4.11 Valid BREAK Symbol

In Figure 15-9, if the next active-to-passive received transition does not occur until after **E**, the current symbol is considered a valid BREAK symbol. A BREAK symbol should be followed by a start-of-frame (SOF) symbol beginning the next message to be transmitted onto the J1850 bus. See 15.7.2 J1850 Frame Format for BDLC response to BREAK symbols.

15.7.5 Message Arbitration

Message arbitration on the J1850 bus is accomplished in a non-destructive manner, allowing the message with the highest priority to be transmitted, while any transmitters which lose arbitration simply stop transmitting and wait for an idle bus to begin transmitting again.

If the BDLC wants to transmit onto the J1850 bus, but detects that another message is in progress, it waits until the bus is idle. However, if multiple nodes begin to transmit in the same synchronization window, message arbitration occurs beginning with the first bit after the SOF symbol and continues with each bit thereafter. If a write to the BDR (for instance, to initiate transmission) occurred on or before $104 \bullet t_{BDLC}$ from the received rising edge, then the BDLC transmits and arbitrates for the bus. If a CPU write to the BDR occurred after $104 \bullet t_{BDLC}$ from the detection of the rising edge, then the BDLC does not transmit, but waits for the next IFS period to expire before attempting to transmit the byte.

The variable pulse-width modulation (VPW) symbols and J1850 bus electrical characteristics are chosen carefully so that a logic 0 (active or passive type) always dominates over a logic 1 (active or passive type) simultaneously transmitted. Hence, logic 0s are said to be dominant and logic 1s are said to be recessive.

When a node detects a dominant bit on BDRxD when it transmitted a recessive bit, it loses arbitration and immediately stops transmitting. This is known as bitwise arbitration (see Figure 15-10).

Since a logic 0 dominates a logic 1, the message with the lowest value has the highest priority and always wins arbitration. For instance, a message with priority 000 wins arbitration over a message with priority 011.

This method of arbitration works no matter how many bits of priority encoding are contained in the message.

During arbitration, or even throughout the transmitting message, when an opposite bit is detected, transmission is stopped immediately unless it occurs on the eighth bit of a byte. In this case, the BDLC automatically appends up to two extra logic 1 bits and then stops transmitting. These two extra bits are arbitrated normally and thus do not interfere with another message. The second logic 1 bit is not sent if

2. Four identifier acceptance filters, each to be applied to:
   a. 11 bits of the identifier and the RTR bit of CAN 2.0A messages, or
   b. 14 most significant bits of the identifier of CAN 2.0B messages

   Figure 16-4 shows how the first 32-bit filter bank (CIDAR0–CIDAR3, CIDMR0–CIDMR3) produces filter 0 and 1 hit. Similarly, the second filter bank (CIDAR4–CUIDAR7, CIDMR4–CIDMR7) produces filter 2 and three hits.

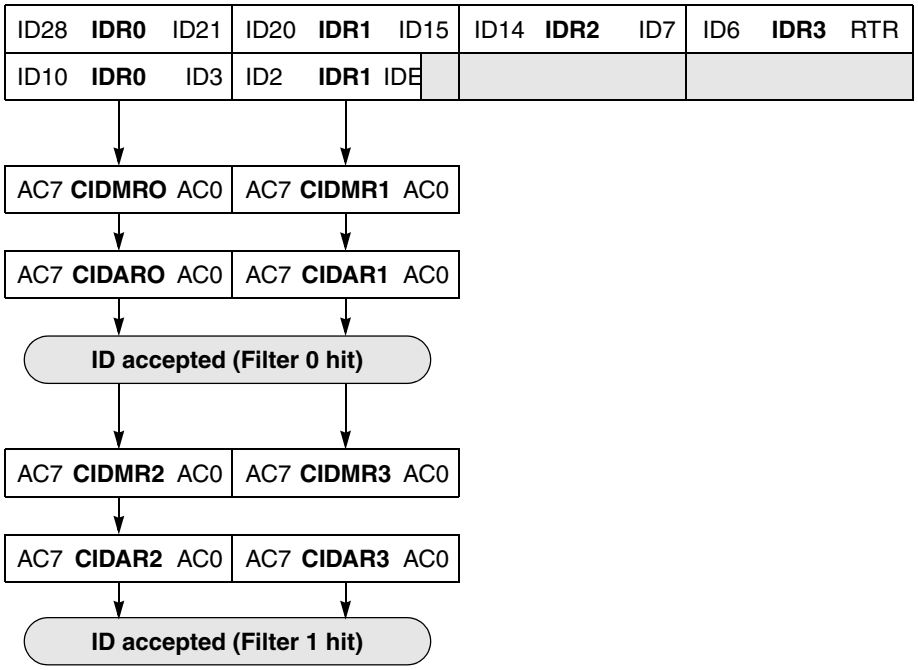


Figure 16-4. 16-Bit Maskable Acceptance Filters

3. Eight identifier acceptance filters, each to be applied to the first eight bits of the identifier. This mode implements eight independent filters for the first eight bits of a CAN 2.0A compliant standard identifier or of a CAN 2.0B compliant extended identifier. Figure 16-5 shows how the first 32-bit filter bank (CIDAR0–CIDAR3, CIDMR0–CIDMR3) produces filter 0 to three hits. Similarly, the second filter bank (CIDAR4–CUIDAR7, CIDMR4–CIDMR7) produces filter 4 to seven hits.
4. Closed filter. No CAN message will be copied into the foreground buffer RxFG, and the RXF flag will never be set.

16.10 Memory Map

The msCAN12 occupies 128 bytes in the CPU12 memory space. The background receive buffer can be read only in test mode.

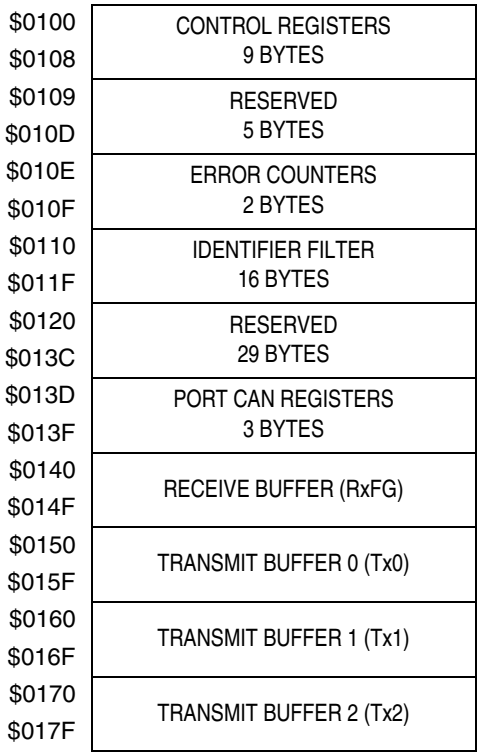


Figure 16-9. msCAN12 Memory Map

16.11 Programmer's Model of Message Storage

This subsection details the organization of the receive and transmit message buffers and the associated control registers.

16.11.1 Message Buffer Organization

Figure 16-10 shows the organization of a single message buffer. For reasons of programmer interface simplification, the receive and transmit message buffers have the same register organization. Each message buffer allocates 16 bytes in the memory map containing:

- 13-byte data structure which includes an identifier section (IDRn), a data section (DSRn), and the data length register (DLR)
- Transmit buffer priority register (TBPR) which is only applicable for transmit buffers. See 16.11.5 Transmit Buffer Priority Register
- Two unused bytes

All bits of the 13-byte data structure are undefined out of reset.

NOTE

The receive buffer can be read anytime but cannot be written. The transmit buffers can be read or written anytime.

Address: $0114

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset:	Unaffected by reset							

Address: $0115

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset:	Unaffected by reset							

Address: $0116

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset:	Unaffected by reset							

Address: $0117

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset:	Unaffected by reset							

Figure 16-29. First Bank msCAN12 Identifier Mask Registers (CIDMR0–CIDMR3)

Address: $011C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset:	Unaffected by reset							

Address: $011D

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset:	Unaffected by reset							

Address: $011E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset:	Unaffected by reset							

Address: $011F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset:	Unaffected by reset							

Figure 16-30. Second Bank msCAN12 Identifier Mask Registers (CIDMR4–CIDMR7)

Clearing these bits causes the prescale value to default to 1 which results in a divide-by-two prescale factor. This signal is then fed into the divide-by-two logic. The reset state divides the P clock by a total of four and is appropriate for nominal operation at a bus rate of between 2 MHz and 8 MHz. Table 17-3 shows the divide-by operation and the appropriate range of system clock frequencies.

Table 17-3. Clock Prescaler Values

Prescale Value	Total Divisor	Max P Clock(1)	Min P Clock(2)
00000	2	4 MHz	1 MHz
00001	4	8 MHz	2 MHz
00010	6	8 MHz	3 MHz
00011	8	8 MHz	4 MHz
00100	10	8 MHz	5 MHz
00101	12	8 MHz	6 MHz
00110	14	8 MHz	7 MHz
00111	16	8 MHz	8 MHz
01xxx	Do not use		
1xxxx			

1. Maximum conversion frequency is 2 MHz. Maximum P clock divisor value becomes maximum conversion rate that can be used on this ATD module.
2. Minimum conversion frequency is 500 kHz. Minimum P clock divisor value becomes minimum conversion rate that this ATD can perform.

17.3.6 ATD Control Register 5

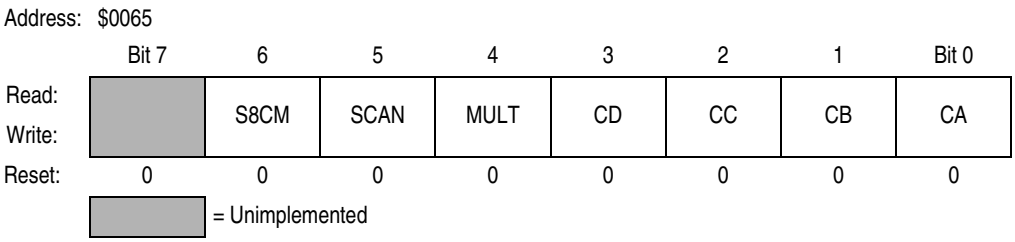


Figure 17-7. ATD Control Register 5 (ATDCTL5)

Read: Anytime

Write: Anytime

The ATD control register 5 is used to select the conversion modes, the conversion channel(s), and initiate conversions.

A write to ATDCTL5 initiates a new conversion sequence. If a conversion sequence is in progress when a write occurs, that sequence is aborted and the SCF and CCF bits are reset.

S8CM — Select 8 Channel Mode Bit

0 = Conversion sequence consists of four conversions.
1 = Conversion sequence consists of eight conversions.

19.10 ATD AC Operating Characteristics (Operating)

Characteristic[1]	Symbol	Min	Max	Unit
MCU clock frequency (p-clock)	f_{PCLK}	2.0	8.0	MHz
ATD operating clock frequency	f_{ATDCLK}	0.5	2.0	MHz
ATD 8-bit conversion period ATD clock cycles[2] ATD conversion time[3]	 n_{CONV8} t_{CONV8}	 18 9	 32 16	 Cycles µs
ATD 10-bit conversion period ATD clock cycles[2] ATD conversion time[3]	 n_{CONV10} t_{CONV10}	 20 10.0	 34 17	 Cycles µs
Stop and ATD power-up recovery time[4] V_{DDA} = 5.0 V	t_{SR}	—	10	µs

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, ATD clock = 2 MHz, unless otherwise noted
2. The minimum time assumes a final sample period of 2 ATD clock cycles while the maximum time assumes a final sample period of 16 ATD clocks.
3. This assumes an ATD clock frequency of 2.0 MHz.
4. From the time ADPU is asserted until the time an ATD conversion can begin

19.11 EEPROM Characteristics

Characteristic[1]	Symbol	Min	Typ	Max	Unit
Minimum programming clock frequency[2]	f_{PROG}	1.0	—	—	MHz
Programming time	t_{PROG}	10.0	—	10.5	ms
Clock recovery time, following STOP, to continue programming	t_{CRSTOP}	—	—	t_{PROG}+ 1	ms
Erase time	t_{ERASE}	10.0	—	10.5	ms
Write/erase endurance	—	10,000	—	—	Cycles
Data retention	—	10	—	—	Years

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted
2. RC oscillator must be enabled if programming is desired and $f_{SYS} < f_{PROG}$.

19.16 Multiplexed Expansion Bus Timing

NOTE

Use of the multiplexed expansion bus at 8 MHz is discouraged due to TAD delay factors.

Num	Characteristic[1], [2], [3], [4], [5]	Delay	Symbol	8 MHz		2 MHz		Unit
				Min	Max	Min	Max	
—	Frequency of operation (E-clock frequency)	—	f_o	dc	8.0	dc	8.0	MHz
1	Cycle time $t_{cyc} = 1/f_o$	—	t_{cyc}	125	—	500	—	ns
2	Pulse width, E low $PW_{EL} = t_{cyc}/2 + delay$	–4	PW_{EL}	59	—	246	—	ns
3	Pulse width, E high[6] $PW_{EH} = t_{cyc}/2 + delay$	–2	PW_{EH}	59	—	248	—	ns
5	Address delay time $t_{AD} = t_{cyc}/4 + delay$	27	t_{AD}	—	67.5	—	152	ns
7	Address valid time to ECLK rise $t_{AV} = PW_{EL} – t_{AD}$	—	t_{AV}	–6.2	—	94	—	ns
8	Multiplexed address hold time $t_{MAH} = t_{cyc}/4 + delay$	–18	t_{MAH}	13	—	107	—	ns
9	Address hold to data valid	—	t_{AHDS}	30	—	20	—	ns
10	Data hold to high impedance $t_{DHZ} = t_{AD} – 20$	—	t_{DHZ}	—	45.2	—	132	ns
11	Read data setup time	—	t_{DSR}	31.2	—	25	—	ns
12	Read data hold time	—	t_{DHR}	0	—	0	—	ns
13	Write data delay time	—	t_{DDW}	—	62.5	—	165	ns
14	Write data hold time	—	t_{DHW}	25	—	20	—	ns
15	Write data setup time[6] $t_{DSW} = PW_{EH} – t_{DDW}$	—	t_{DSW}	5.8	—	83	—	ns
16	Read/write delay time $t_{RWD} = t_{cyc}/4 + delay$	18	t_{RWD}	—	57.5	—	143	ns
17	Read/write valid time to E rise $t_{RWV} = PW_{EL} – t_{RWD}$	—	t_{RWV}	3.8	—	103	—	ns
18	Read/write hold time	—	t_{RWH}	25	—	20	—	ns
19	Low strobe[7] delay time $t_{LSD} = t_{cyc}/4 + delay$	18	t_{LSD}	—	57.5	—	143	ns
20	Low strobe[7] valid time to E rise $t_{LSV} = PW_{EL} – t_{LSD}$	—	t_{LSV}	3.8	—	103	—	ns
21	Low strobe[7] hold time	—	t_{LSH}	25	—	20	—	ns
22	Address access time[6] $t_{ACCA} = t_{cyc} – t_{AD} – t_{DSR}$	—	t_{ACCA}	—	27.6	—	323	ns
23	Access time from E rise[6] $t_{ACCE} = PW_{EH} – t_{DSR}$	—	t_{ACCE}	—	27.8	—	223	ns
24	$\overline{DBE}$ delay from ECLK rise[6] $t_{DBED} = t_{cyc}/4 + delay$	8	t_{DBED}	—	57.5	—	133	ns
25	$\overline{DBE}$ valid time $t_{DBE} = PW_{EH} – t_{DBED}$	—	t_{DBE}	11.8	—	115	—	ns
26	$\overline{DBE}$ hold time from ECLK fall	—	t_{DBEH}	–3	10	–3	10	ns

1. $V_{DD} = 5.0$ Vdc ± 10%, $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H, unless otherwise noted
2. All timings are calculated for normal port drives.
3. Crystal input is required to be within 45% to 55% duty.
4. Reduced drive must be off to meet these timings.
5. Unequalled loading of pins will affect relative timing numbers.
6. This characteristic is affected by clock stretch.
   Add N × t_{cyc} where N = 0, 1, 2, or 3, depending on the number of clock stretches.
7. Without TAG enabled