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	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 ~ 125°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/mchc912b32mfue8

Chapter 2
Register Block

Chapter 3
Central Processor Unit (CPU)

Chapter 4
Resets and Interrupts

- Serial interfaces:
  - Asynchronous serial communications interface (SCI)
  - Synchronous serial peripheral interface (SPI)
  - J1850 byte data link communication (BDLC), MC68HC912B32 and MC68HC12BE32 only
  - Controller area network (CAN), MC68HC(9)12BC32 only
- Computer operating properly (COP) watchdog timer, clock monitor, and periodic interrupt timer
- Slow-mode clock divider
- 80-pin quad flat pack (QFP)
- Up to 63 general-purpose input/output (I/O) lines
- Single-wire background debug mode (BDM)
- On-chip hardware breakpoints

1.3 Slow-Mode Clock Divider Advisory

Current versions of the M68HC12B-series devices include a slow-mode clock divider feature. This feature is fully described in Chapter 10 Clock Generation Module (CGM). The register that controls this feature is located at $00E0. Older device mask sets do not support the slow-mode clock divider feature. This register address is reserved in older devices and provides no function.

Mask sets that do not have the slow-mode clock divider feature on the MC68HC912B32 include: G96P, G86W, and H91F.

Mask sets that do not have the slow-mode clock divider feature on the MC68HC12BE32 include: H54T and J38M.

Mask sets that do not have the slow-mode clock divider feature on the MC68HC(9)12BC32 include: J15G.

Table 1-3. Signal Description Summary

Pin Name	Pin Number	Description
PW3–PW0	3–6	Pulse-width modulator channel outputs
ADDR7–ADDR0 DATA7–DATA0	25–18	External bus pins share function with general-purpose I/O ports A and B. In single-chip modes, the pins can be used for I/O. In expanded modes, the pins are used for the external buses.
ADDR15–ADDR8 DATA15–DATA8	46–39	
IOC7–IOC0	16–12, 9–7	Pins used for input capture and output compare in the timer and pulse accumulator subsystem
PAI	16	Pulse accumulator input
AN7–AN0	58–51	Analog inputs for the analog-to-digital conversion module
$\overline{\text{DBE}}$	26	Data bus control and, in expanded mode, enables the drive control of external buses during external reads
MODB, MODA	27, 28	State of mode select pins during reset determines the initial operating mode of the MCU. After reset, MODB and MODA can be configured as instruction queue tracking signals IPIPE1 and IPIPE0 or as general-purpose I/O pins.
IPIPE1, IPIPE0	27, 28	
ECLK	29	E-clock is the output connection for the external bus clock. ECLK is used as a timing reference and for address demultiplexing.
$\overline{\text{RESET}}$	32	An active low bidirectional control signal, $\overline{\text{RESET}}$ acts as an input to initialize the MCU to a known startup state and an output when COP or clock monitor causes a reset.
EXTAL	33	Crystal driver and external clock input pins. On reset all the device clocks are derived from the EXTAL input frequency. XTAL is the crystal output.
XTAL	34	
$\overline{\text{LSTRB}}$	35	Low byte strobe (0 = low byte valid), in all modes this pin can be used as I/O. The low strobe function is the exclusive-NOR of A0 and the internal $\overline{\text{SZ8}}$ signal. The $\overline{\text{SZ8}}$ internal signal indicates the size 16/8 access.
$\overline{\text{TAGLO}}$	35	Pin used in instruction tagging
R/$\overline{\text{W}}$	36	Indicates direction of data on expansion bus; shares function with general-purpose I/O; read/write in expanded modes
$\overline{\text{IRQ}}$	37	Maskable interrupt request input provides a means of applying asynchronous interrupt requests to the MCU. Either falling edge-sensitive triggering or level-sensitive triggering is program selectable (INTCR register).
$\overline{\text{XIRQ}}$	38	Provides a means of requesting asynchronous non-maskable interrupt requests after reset initialization
BKGD	17	Single-wire background interface pin is dedicated to the background debug function. During reset, this pin determines special or normal operating mode.
TAGHI	17	Pin used in instruction tagging
DLCRx/RxCAN(1)	76	BDLC receive pin
DLCTx/TxCAN(1)	75	BDLC transmit pin
$\overline{\text{CS}}$/$\overline{\text{SS}}$	68	Slave-select output for SPI master mode; input for slave mode or master mode
SCK	67	Serial clock for SPI system
SDO/MOSI	66	Master out/slave in pin for serial peripheral interface
SDI/MISO	65	Master in/slave out pin for serial peripheral interface
TxD0	62	SCI transmit pin
RxD0	61	SCI receive pin

1. The RxCAN and TxCAN designations are for the MC68HC(9)12BC32 only.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$00B3	8-Bit Pulse Accumulator Holding Register 2 (PA2H) See page 186.	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
$00B4	8-Bit Pulse Accumulator Holding Register 1 (PA1H) See page 186.	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
$00B5	8-Bit Pulse Accumulator Holding Register 0 (PA0H) See page 186.	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
$00B6	Modulus Down-Counter Count Register (MCCNT) See page 187.	Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
$00B7	Modulus Down-Counter Count Register (MCCNT) See page 187.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$00B8	Timer Input Capture Holding Register 0 (TC0H) See page 187.	Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	0	0	1	0	0	0	0	0
$00B9	Timer Input Capture Holding Register 0 (TC0H) See page 188.	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
$00BA	Timer Input Capture Holding Register 1 (TC1H) See page 188.	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
$00BB	Timer Input Capture Holding Register 1 (TC1H) See page 188.	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
$00BC	Timer Input Capture Holding Register 2 (TC2H) See page 188.	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
$00BD	Timer Input Capture Holding Register 2 (TC2H) See page 188.	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
$00BE	Timer Input Capture Holding Register 3 (TC3H) See page 188.	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

= Unimplemented R = Reserved U = Unaffected

Notes:
1. Available only on MC68HC912B32 and MC68HC912BC32 devices.
2. Available only on MC68HC912B32 and MC68HC12BE32 devices.
3. Available only on MC68HC(9)12BC32 devices.

Figure 2-1. Register Map (Sheet 12 of 19)

3.3 CPU Registers

This section describes the CPU registers.

3.3.1 Accumulators A and B

Accumulators A and B are general-purpose 8-bit accumulators that contain operands and results of arithmetic calculations or data manipulations.

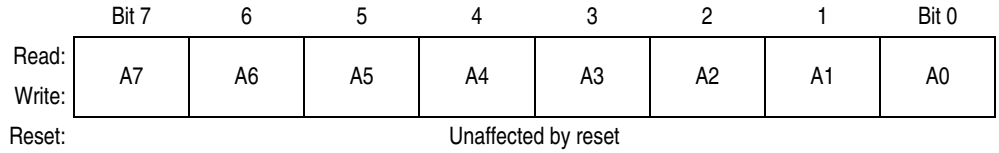

Figure 3-2. Accumulator A (A)

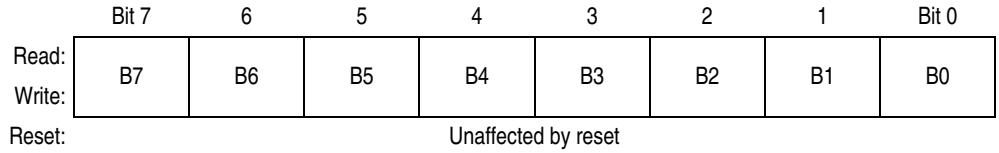

Figure 3-3. Accumulator B (B)

3.3.2 Accumulator D

Accumulator D is the concatenation of accumulators A and B. Some instructions treat the combination of these two 8-bit accumulators as a 16-bit double accumulator.

NOTE

The LDD and STD instructions can be used to manipulate data in and out of accumulator D.

Figure 3-4. Accumulator D (D)

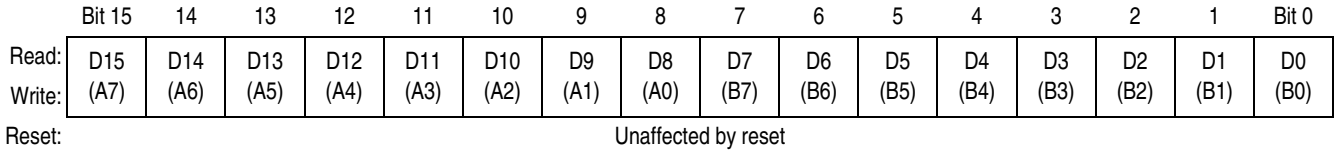

Table 4-2. MC68HC(9)12BC32 Interrupt Vector Map

Vector Address	Interrupt Source	CCR Mask	Local Enable		HPRIO Value to Elevate to Highest I Bit
			Register	Bit(s)	
$FFFE, $FFFF	Reset	None	None	None	—
$FFFC, $FFFD	COP clock monitor fail reset	None	COPCTL	CME, FCME	—
$FFFA, $FFFB	COP failure reset	None	None	COP rate selected	—
$FFF8, $FFF9	Unimplemented instruction trap	None	None	None	—
$FFF6, $FFF7	SWI	None	None	None	—
$FFF4, $FFF5	XIRQ	X bit	None	None	—
$FFF2, $FFF3	IRQ	I bit	INTCR	IRQEN	$F2
$FFF0, $FFF1	Real-time interrupt	I bit	RTICTL	RTIE	$F0
$FFEE, $FFEF	Timer channel 0	I bit	TMSK1	C0I	$EE
$FFEC, $FFED	Timer channel 1	I bit	TMSK1	C1I	$EC
$FFEA, $FFEB	Timer channel 2	I bit	TMSK1	C2I	$EA
$FFE8, $FFE9	Timer channel 3	I bit	TMSK1	C3I	$E8
$FFE6, $FFE7	Timer channel 4	I bit	TMSK1	C4I	$E6
$FFE4, $FFE5	Timer channel 5	I bit	TMSK1	C5I	$E4
$FFE2, $FFE3	Timer channel 6	I bit	TMSK1	C6I	$E2
$FFE0, $FFE1	Timer channel 7	I bit	TMSK1	C7I	$E0
$FFDE, $FFDF	Timer overflow	I bit	TMSK2	TOI	$DE
$FFDC, $FFDD	Pulse accumulator overflow	I bit	PACTL	PAOVI	$DC
$FFDA, $FFDB	Pulse accumulator input edge	I bit	PACTL	PAI	$DA
$FFD8, $FFD9	SPI serial transfer complete	I bit	SP0CR1	SPIE	$D8
$FFD6, $FFD7	SCI 0	I bit	SC0CR2	TIE, TCIE, RIE, ILIE	$D6
$FFD4, $FFD5	Reserved	I bit	—	—	$D4
$FFD2, $FFD3	ATD	I bit	ATDCTL2	ASCIE	$D2
$FFD0, $FFD1	MSCAN wakeup	I bit	CRIER	WUPIE	$D0
$FFCA–$FFCF	Reserved (not implemented)	I bit	—	—	$CA–$CF
$FFC8–$FFC9	MSCAN errors	I bit	CRIER	RWRNIE, TWRNIE, RERRIE, TERRIE, BOFFIE, OVRIE	$C8
$FFC6, $FFC7	MSCAN receive	I bit	CRIER	RXFIE	$C6
$FFC4, $FFC5	MSCAN transmit	I bit	CTCR	TXEIE[2:0]	$C4
$FF80, $FFC3	Reserved (implemented)	I bit	—	—	$80–$C3

5.4.4 EEPROM Initialization Register

The MCU has 768 bytes of EEPROM which are activated by the EEON bit in the EEPROM initialization register (INITEE).

Mapping of internal EEPROM is controlled by four bits in the INITEE register. After reset, EEPROM address space begins at location $0D00 but can be mapped to any 4-Kbyte boundary within the standard 64-Kbyte address space.

Address: $0012

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	EE15	EE14	EE13	EE12	0	0	0	EEON
Reset:	0	0	0	0	0	0	0	1

Figure 5-4. EEPROM Initialization Register (INITEE)

Read: anytime

Write: varies from bit to bit

EE15–EE12 — Internal EEPROM Position Bits

These bits specify the upper four bits of the 16-bit EEPROM address. Write once in normal modes or anytime in special modes.

EEON — EEPROM On Bit

EEON allows read access to the EEPROM array. EEPROM control registers can be accessed and EEPROM locations can be programmed or erased regardless of the state of EEON.
EEON is forced to 1 in single-chip modes.
Write only in expanded and peripheral modes.

1 = EEPROM in memory map
0 = EEPROM removed from memory map

5.4.5 Miscellaneous Mapping Control Register

Additional mapping controls are available that can be used in conjunction with FLASH EEPROM/ROM and memory expansion.

The 32-Kbyte FLASH EEPROM/ROM can be mapped to either the upper or lower half of the 64-Kbyte address space. When mapping conflicts occur, registers, RAM, and EEPROM have priority over FLASH EEPROM.

NOTE

Only the MC68HC912B32 contains FLASH EEPROM. The MC68HC12BE32 contains ROM.

To use memory expansion, the part must be operated in one of the expanded modes.

8.3.4 FLASH EEPROM Control Register

Address: $00F7

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	0	0	0	FEESWAI	SVFP	ERAS	LAT	ENPE
Reset:	0	0	0	0	0	0	0	0

Figure 8-4. FLASH EEPROM Control Register (FEECTL)

This register controls the programming and erasure of the FLASH EEPROM.

FEESWAI — FLASH EEPROM Stop in Wait Control Bit

0 = Do not halt FLASH EEPROM clock when in wait mode.
1 = Halt FLASH EEPROM clock when in wait mode.

NOTE

The FEESWAI bit cannot be asserted if the interrupt vector resides in the FLASH EEPROM array.

SVFP — Status V_{FP} Voltage Bit

SVFP is a read-only bit.

0 = Voltage of V_{FP} pin is below normal programming voltage levels.
1 = Voltage of V_{FP} pin is above normal programming voltage levels.

ERAS — Erase Control Bit

This bit can be read anytime or written when ENPE = 0. When set, all locations in the array will be erased at the same time. The boot block will be erased only if BOOTP = 0. This bit also affects the result of attempted array reads. See Table 8-1 for more information. Status of ERAS cannot change if ENPE is set.

0 = FLASH EEPROM configured for programming
1 = FLASH EEPROM configured for erasure

LAT — Latch Control Bit

This bit can be read anytime or written when ENPE = 0. When set, the FLASH EEPROM is configured for programming or erasure and, upon the next valid write to the array, the address and data will be latched for the programming sequence. See Table 8-1 for the effects of LAT on array reads. A high voltage detect circuit on the V_{FP} pin will prevent assertion of the LAT bit when the programming voltage is at normal levels.

0 = Programming latches disabled
1 = Programming latches enabled

ENPE — Enable Programming/Erase Bit

0 = Disables program/erase voltage to FLASH EEPROM
1 = Applies program/erase voltage to FLASH EEPROM

ENPE can be asserted only after LAT has been asserted and a write to the data and address latches has occurred. If an attempt is made to assert ENPE when LAT is negated, or if the latches have not been written to after LAT was asserted, ENPE will remain negated after the write cycle is complete. The LAT, ERAS, and BOOTP bits cannot be changed when ENPE is asserted. A write to FEECTL may affect only the state of ENPE. Attempts to read a FLASH EEPROM array location in the FLASH EEPROM module while ENPE is asserted will not return the data addressed. See Table 8-1 for more information.

Chapter 9
Read-Only Memory (ROM)

9.1 Introduction

The MC68HC12BE32 and MC68HC12BC32 contain 32 Kbytes of read-only memory (ROM). The ROM array is arranged in a 16-bit configuration and may be read as either bytes, aligned words or misaligned words. Access time is one bus cycle for byte and aligned word access and two bus cycles for misaligned word operations.

9.2 ROM Array

After reset, the ROM array is located from addresses $8000 to $FFFF in single-chip mode. In expanded modes, the ROM array is located from address $0000 to $7FFF; however, it is disabled from the memory map. The ROM can be mapped to an alternate address range. See Chapter 5 Operating Modes and Resource Mapping.

11.4.1 PWM Period Calculation

These parameters were used to calculate the high-time values shown in Table 11-3:

- Period = $1000 (Hex) = 4096 (decimal)
- E clock = 8 MHz
- Prescaler = 4
- Frequency = (8 MHz) / (#clocks_count*prescaler) = (8 MHz) (4096*4) = 500 Hz
- If period ($4096 Clocks) => frequency = 500 Hz

Table 11-3. PWM Period Calculations

High-Time Values		Duty Cycle
HEX Count	Decimal Count	
$0020	32	0%
$0040	64	1.5%
$0080	128	3.1%
$0100	256	6.25%
$0200	512	12.50%
$0040	1024	25.00%
$0080	2048	50.00%
$0996	2454	60.00%
$0C00	3072	75.00%
$0D9A	3482	85.00%
$1000	4096	100.00%

11.4.2 Equipment

For this exercise, use the M68HC912B32EVB emulation board.

Address: $0092–$0093

	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 13-22. Timer Input Capture/Output Compare Register 1 (TC1)

Address: $0094–$0095

	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 13-23. Timer Input Capture/Output Compare Register 2 (TC2)

Address: $0096–$0097

	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 13-24. Timer Input Capture/Output Compare Register 3 (TC3)

Address: $0098–$0099

	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 13-25. Timer Input Capture/Output Compare Register 4 (TC4)

15.5.3 BDLC Stop and CPU Stop Mode

This power-conserving mode is entered automatically from run mode when the WCM bit in the BCR1 register is set followed by a CPU STOP instruction. This is the lowest-power mode that the BDLC can enter.

In this mode:

- The BDLC internal clocks are stopped.
- The CPU internal clocks are stopped.
- The BDLC awaits J1850 network activity.

The first passive-to-active transition on the J1850 network generates a non-maskable ($20) CPU interrupt request by the BDLC, allowing the CPU clocks to restart and the BDLC internal clocks to restart. Therefore, the new message which wakes up the BDLC from the BDLC stop mode and the CPU from the CPU wait mode are not received correctly. This is due primarily to the time required for the MCU's oscillator to stabilize before the clocks can be applied internally to the other MCU modules, including the BDLC.

NOTE

Ensure that all transmissions are complete or aborted prior to putting the BDLC into stop mode (WCM = 1 in BCR1).

15.6 Loopback Modes

Two loopback modes are used to determine the source of bus faults.

15.6.0.1 Digital Loopback Mode

When a bus fault has been detected, the digital loopback mode is used to determine if the fault condition is caused by failure in the node's internal circuits or elsewhere in the network, including the node's analog physical interface. In this mode, the transmit digital output pin (BDTxD) and the receive digital input pin (BDRxD) of the digital interface are disconnected from the analog physical interface and tied together to allow the digital portion of the BDLC to transmit and receive its own messages without driving the J1850 bus.

15.6.0.2 Analog Loopback Mode

Analog loopback mode is used to determine if a bus fault has been caused by a failure in the node's off-chip analog transceiver or elsewhere in the network. The BDLC analog loopback mode does not modify the digital transmit or receive functions of the BDLC. It does, however, ensure that once analog loopback mode is exited, the BDLC waits for an idle bus condition before participation in network communication resumes. If the off-chip analog transceiver has a loopback mode, it usually causes the input to the output drive stage to be looped back into the receiver, allowing the node to receive messages it has transmitted without driving the J1850 bus. In this mode, the output to the J1850 bus typically is high impedance. This allows the communication path through the analog transceiver to be tested without interfering with network activity. Using the BDLC analog loopback mode in conjunction with the analog transceiver's loopback mode ensures that, once the off-chip analog transceiver has exited loopback mode, the BDLC does not begin communicating before a known condition exists on the J1850 bus.

If the TMIFR1 bit is set, the BDLC attempts to transmit the normalization symbol followed by the byte in the BDR. After the byte in the BDR has been loaded into the transmit shift register, a TDRE interrupt (see 15.9.3 BDLC State Vector Register) occurs similar to the main message transmit sequence. The programmer should then load the next byte of the IFR into the BDR for transmission. When the last byte of the IFR has been loaded into the BDR, the programmer should set the TEOD bit in the BDLC control register 2 (BCR2). This instructs the BDLC to transmit a CRC byte once the byte in the BDR is transmitted, and then transmit an EOD symbol, indicating the end of the IFR portion of the message frame.

However, to transmit a single byte followed by a CRC byte, the programmer should load the byte into the BDR before the EOD symbol has been received, and then set the TMIFR1 bit. Once the TDRE interrupt occurs, the programmer sets the TEOD bit in the BCR2. This results in the byte in the BDR being the only byte transmitted before the IFR CRC byte, and no TDRE interrupt is generated.

If the programmer attempts to set the TMIFR1 bit immediately after the EOD symbol has been received from the bus, the TMIFR1 bit remains in the reset state, and no attempt is made to transmit an IFR byte.

If a loss of arbitration occurs when the BDLC is transmitting any byte of a multiple byte IFR, the BDLC goes to the loss of arbitration state, sets the appropriate flag, and ceases transmission.

If the BDLC loses arbitration during the IFR, the TMIFR1 bit is cleared and no attempt is made to retransmit the byte in the BDR. If loss of arbitration occurs in the last bit of the IFR byte, two additional 1 bits are sent out.

NOTE

The extra logic 1 bits are an enhancement to the J1850 protocol which forces a byte boundary condition fault. This is helpful in preventing noise on the J1850 bus from corrupting a message.

15.9.3 BDLC State Vector Register

Address: $00F9

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	I3	I2	I1	I0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 15-15. BDLC State Vector Register (BSVR)

This register is provided to substantially decrease the CPU overhead associated with servicing interrupts while under operation of a multiplex protocol. It provides an index offset that is directly related to the BDLC's current state, which can be used with a user-supplied jump table to rapidly enter an interrupt service routine. This eliminates the need for the user to maintain a duplicate state machine in software.

I0, I1, I2, I3 — Interrupt Source Bits

These bits indicate the source of the pending interrupt request. Bits are encoded according to Table 15-4.

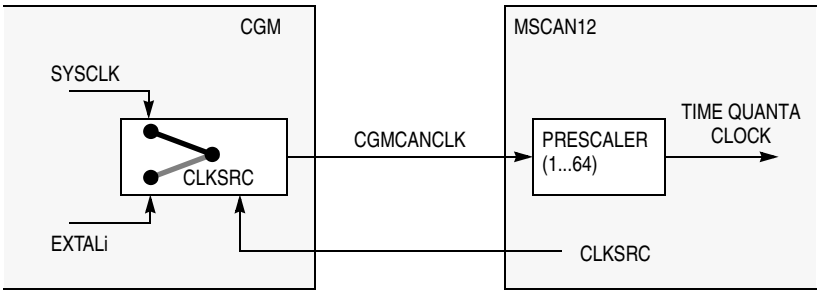

Figure 16-7. Clocking Scheme

The clock source bit (CLKSRC) in the msCAN12 module control register (CMCR1) (see 16.12.3 msCAN12 Bus Timing Register 0) defines whether the msCAN12 is connected to the output of the crystal oscillator (EXTALi) or to a clock twice as fast as the system clock (ECLK).

The clock source has to be chosen so that the tight oscillator tolerance requirements (up to 0.4 percent) of the CAN protocol are met. Additionally, for high CAN bus rates (1 Mbps), a 50 percent duty cycle of the clock is required.

For microcontrollers without the CGM module, CGMCANCLK is driven from the crystal oscillator (EXTALi).

A programmable prescaler is used to generate out of msCANCLK the time quanta (Tq) clock. A time quantum is the atomic unit of time handled by the msCAN12.

$$f_{Tq} = \frac{f_{CGMCANCLK}}{Presc\ \ value}$$

A bit time is subdivided into three segments[1]:

- SYNC_SEG — This segment has a fixed length of one time quantum. Signal edges are expected to happen within this section.
- Time segment 1 — This segment includes the PROP_SEG and the PHASE_SEG1 of the CAN standard. It can be programmed by setting the parameter TSEG1 to consist of 4 to 16 time quanta.
- Time segment 2 — This segment represents the PHASE_SEG2 of the CAN standard. It can be programmed by setting the TSEG2 parameter to be 2 to 8 time quanta.

$$BitRate = \frac{f_{Tq}}{number\ \ of\ \ TimeQuanta}$$

The synchronization jump width can be programmed in a range of 1 to 4 time quanta by setting the SJW parameter. These parameters can be set by programming the bus timing registers (CBTR0 and CBTR1). See **16.12.3 msCAN12 Bus Timing Register 0** and 16.12.4 msCAN12 Bus Timing Register 1.

NOTE

It is the user's responsibility to make sure that the bit time settings are in compliance with the CAN standard. Table 16-3 gives an overview on the CAN-conforming segment settings and the related parameter values.

1. For further explanation of the underlying concepts, refer to ISO/DIS 11519-1, Section 10.3.

RTR — Remote Transmission Request Flag

This flag reflects the status of the remote transmission request bit in the CAN frame. In case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In case of a transmit buffer, this flag defines the setting of the RTR bit to be sent.

0 = Data frame
1 = Remote frame

16.11.3 Data Length Register

The data length register (DLR) keeps the data length field of the CAN frame.

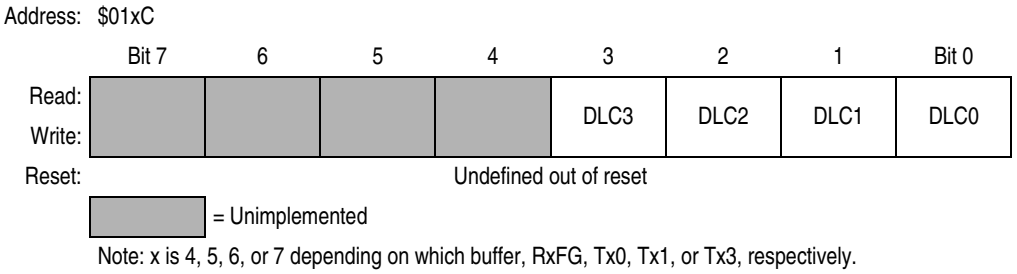

Figure 16-13. Data Length Register (DLR)

DLC3–DLC0 — Data Length Code Bits

The data length code contains the number of bytes (data byte count) of the respective message. At transmission of a remote frame, the data length code is transmitted as programmed while the number of transmitted bytes is always 0. The data byte count ranges from 0 to 8 for a data frame. Table 16-4 shows the effect of setting the DLC bits.

Table 16-4. Data Length Codes

Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8

16.12.11 msCAN12 Transmit Error Counter

Address: $010F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 16-26. msCAN12 Transmit Error Counter (CTXERR)

This register reflects the status of the msCAN12 transmit error counter. The register is read only.

NOTE

Both error counters may be read only when in sleep or soft-reset mode.

16.12.12 msCAN12 Identifier Acceptance Registers

On reception, each message is written into the background receive buffer. The CPU is only signalled to read the message if it passes the criteria in the identifier acceptance and identifier mask registers (accepted); otherwise, the message will be overwritten by the next message (dropped).

The acceptance registers of the msCAN12 are applied on the IDR0 to IDR3 registers of incoming messages in a bit-by-bit manner.

For extended identifiers, all four acceptance and mask registers are applied. For standard identifiers only the first two (CIDMR0/CIDMR1 and CIDAR0/CIDAR1) are applied.

Address: $0110

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
Reset:	Unaffected by reset							

Address: $0111

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
Reset:	Unaffected by reset							

Address: $0112

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
Reset:	Unaffected by reset							

Address: $0113

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
Reset:	Unaffected by reset							

Figure 16-27. First Bank msCAN12 Identifier Acceptance Registers (CIDAR0–CIDAR3)

FRZ1 and FRZ0 — Background Debug (Freeze) Enable Bits

When debugging an application, it is useful in many cases to have the ATD pause when a breakpoint is encountered. These two bits determine how the ATD will respond when background debug mode becomes active. See Table 17-1.

Table 17-1. ATD Response to Background Debug Enable

FRZ1	FRZ0	ATD Response
0	0	Continue conversions in active background mode
0	1	Reserved
1	0	Finish current conversion, then freeze
1	1	Freeze when BDM is active

17.3.5 ATD Control Register 4

Address: $0064

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	S10BM	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0
Reset:	0	0	0	0	0	0	0	1

Figure 17-6. ATD Control Register 4 (ATDCTL4)

The ATD control register 4 (ATDCTL4) selects the clock source and sets up the prescaler. Writes to the ATD control registers initiate a new conversion sequence. If a write occurs while a conversion is in progress, the conversion is aborted and ATD activity halts until a write to ATDCTL5 occurs.

S10BM — ATD 10-Bit Mode Control Bit

0 = 8-bit operation
1 = 10-bit operation

SMP1 and SMP0 — Select Sample Time Bits

These bits are used to select one of four sample times after the buffered sample and transfer has occurred. See Table 17-2.

Table 17-2. Final Sample Time Selection

SMP1	SMP0	Final Sample Time	Total 8-Bit Conversion Time	Total 10-Bit Conversion Time
0	0	2 ATD clock periods	18 ATD clock periods	20 ATD clock periods
0	1	4 ATD clock periods	20 ATD clock periods	22 ATD clock periods
1	0	8 ATD clock periods	24 ATD clock periods	26 ATD clock periods
1	1	16 ATD clock periods	32 ATD clock periods	34 ATD clock periods

PRS4–PRS0 — Select Divide-By Factor for ATD P-Clock Prescaler Bits

The binary value written to these bits (1 to 31) selects the divide-by factor for the modulo counter-based prescaler. The P clock is divided by this value plus one, and then fed into a divide-by-two circuit to generate the ATD module clock. The divide-by-two circuit ensures symmetry of the output clock signal.

18.4.2 Breakpoint Registers

Breakpoint operation consists of comparing data in the breakpoint address registers (BRKAH/BRKAL) to the address bus and comparing data in the breakpoint data registers (BRKDH/BRKDL) to the data bus. The breakpoint data registers also can be compared to the address bus. The scope of comparison can be expanded by ignoring the least significant byte of address or data matches.

The scope of comparison can be limited to program data only by setting the BKPM bit in breakpoint control register 0.

To trace program flow, setting the BKPM bit causes address comparison of program data only. Control bits are also available that allow checking read/write matches.

18.4.2.1 Breakpoint Control Register 0

Address: $0020

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BKEN1	BKEN0	BKPM	0	BK1ALE	BK0ALE	0	0
Reset:	0	0	0	0	0	0	0	0

Figure 18-10. Breakpoint Control Register 0 (BRKCT0)

Read and write anytime.
This register is used to control the breakpoint logic.

BKEN1 and BKEN0 — Breakpoint Mode Enable Bits

See Table 18-7.

Table 18-7. Breakpoint Mode Control

BKEN1	BKEN0	Mode Selected	BRKAH/L Usage	BRKDH/L Usage	R/$\overline{W}$	Range
0	0	Breakpoints off	—	—	—	—
0	1	SWI — dual address mode	Address match	Address match	No	Yes
1	0	BDM — full breakpoint mode	Address match	Data match	Yes	Yes
1	1	BDM — dual address mode	Address match	Address match	Yes	Yes

BKPM — Break on Program Addresses

This bit controls whether the breakpoint causes an immediate data breakpoint (next instruction boundary) or a delayed program breakpoint related to an executable opcode. Data and unexecuted opcodes cannot cause a break if this bit is set. This bit has no meaning in SWI dual address mode. The SWI mode only performs program breakpoints.

0 = On match, break at the next instruction boundary
1 = On match, break if the match is an instruction to be executed. This uses tagging as its breakpoint mechanism.

BK1ALE — Breakpoint 1 Range Control Bit

Only valid in dual address mode

0 = BRKDL is not used to compare to the address bus.
1 = BRKDL is used to compare to the address bus.