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	HCS12X
Core Size	16-Bit
Speed	50MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	91
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	64K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=s912xep100j5mal

4.3.1.6 MPU Descriptor Register 0 (MPUDESC0)

Address: Module Base + 0x0006

	7	6	5	4	3	2	1	0
R W	MSTR0	MSTR1	MSTR2	MSTR3	LOW_ADDR[22:19]			
Reset	1[1]	1[1]	1[1]	1[2]	0	0	0	0

1. initialized as set for descriptor 0 only, cleared for all others
2. initialized as set for descriptor 0 only, if MSTR3 is implemented on the device

Figure 4-8. MPU Descriptor Register 0 (MPUDESC0)

Read: Anytime

Write: Anytime

Table 4-8. MPUDESC0 Field Descriptions

Field	Description
7 MSTR0	**Master 0 select bit** — If this bit is set the descriptor is valid for bus master 0 (CPU in supervisor state).
6 MSTR1	**Master 1 select bit** — If this bit is set the descriptor is valid for bus master 1 (CPU in user state).
5 MSTR2	**Master 2 select bit** — If this bit is set the descriptor is valid for bus master 2 (XGATE).
4 MSTR3	**Master 3 select bit** — If this bit is set the descriptor is valid for bus master 3.
3–0 LOW_ADDR[22:19]	**Memory range lower boundary address bits** — The LOW_ADDR[22:19] bits represent bits [22:19] of the global memory address that is used as the lower boundary for the described memory range.

A descriptor can be configured as valid for more than one bus-master at the same time by setting multiple Master select bits to one. Setting all Master select bits of a descriptor to zero disables the descriptor.

4.3.1.7 MPU Descriptor Register 1 (MPUDESC1)

Address: Module Base + 0x0007

	7	6	5	4	3	2	1	0
R W	LOW_ADDR[18:11]							
Reset	0	0	0	0	0	0	0	0

Figure 4-9. MPU Descriptor Register 1 (MPUDESC1)

Read: Anytime

Write: Anytime

Figure 7-10 shows the host receiving a logic 0 from the target. Since the host is asynchronous to the target, there is up to a one clock-cycle delay from the host-generated falling edge on BKGD to the start of the bit time as perceived by the target. The host initiates the bit time but the target finishes it. Since the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 target clock cycles then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 target clock cycles after starting the bit time.

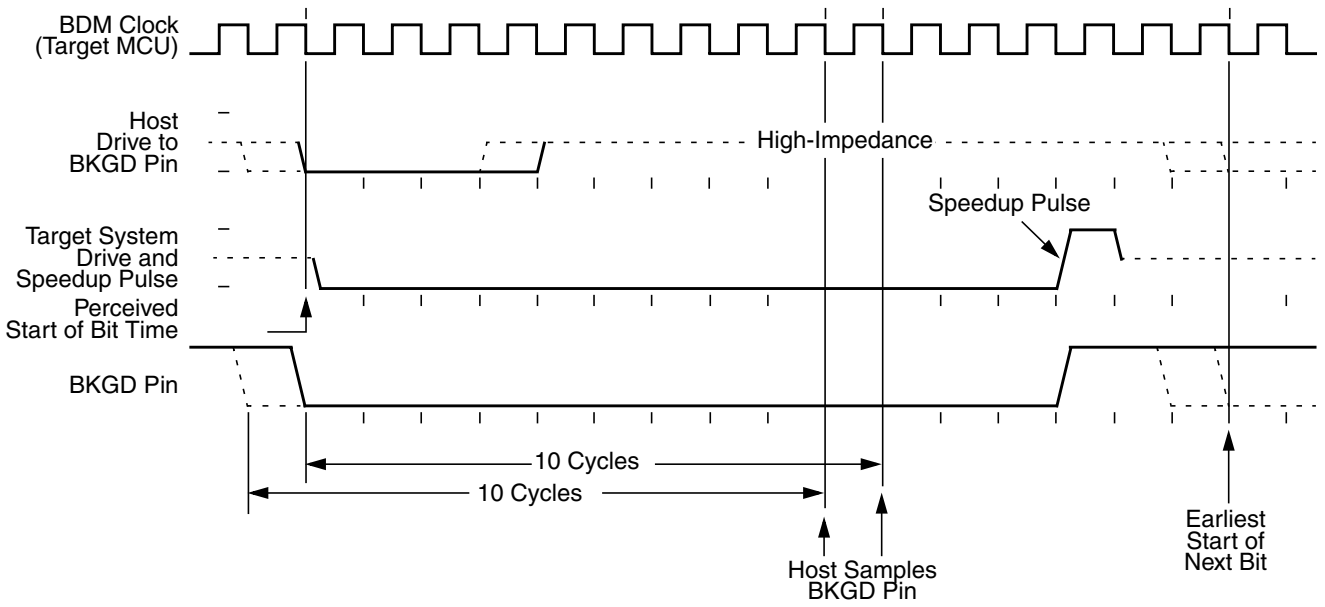


Figure 7-10. BDM Target-to-Host Serial Bit Timing (Logic 0)

7.4.7 Serial Interface Hardware Handshake Protocol

BDM commands that require CPU execution are ultimately treated at the MCU bus rate. Since the BDM clock source can be asynchronously related to the bus frequency, when CLKSW = 0, it is very helpful to provide a handshake protocol in which the host could determine when an issued command is executed by the CPU. The alternative is to always wait the amount of time equal to the appropriate number of cycles at the slowest possible rate the clock could be running. This sub-section will describe the hardware handshake protocol.

The hardware handshake protocol signals to the host controller when an issued command was successfully executed by the target. This protocol is implemented by a 16 serial clock cycle low pulse followed by a brief speedup pulse in the BKGD pin. This pulse is generated by the target MCU when a command, issued by the host, has been successfully executed (see Figure 7-11). This pulse is referred to as the ACK pulse. After the ACK pulse has finished: the host can start the bit retrieval if the last issued command was a read command, or start a new command if the last command was a write command or a control command (BACKGROUND, GO, GO_UNTIL or TRACE1). The ACK pulse is not issued earlier than 32 serial clock cycles after the BDM command was issued. The end of the BDM command is assumed to be the 16th tick of the last bit. This minimum delay assures enough time for the host to perceive the ACK pulse. Note also that, there is no upper limit for the delay between the command and the related ACK pulse, since the command execution depends upon the CPU bus frequency, which in some cases could be very slow

ADC — Add with Carry — ADC

Operation

RS1 + RS2 + C ⇒ RD

Adds the content of register RS1, the content of register RS2 and the value of the Carry bit using binary addition and stores the result in the destination register RD. The Zero Flag is also carried forward from the previous operation allowing 32 and more bit additions.

Example:

```
ADD     R6,R2,R2
ADC     R7,R3,R3 ; R7:R6 = R5:R4 + R3:R2
BCC              ; conditional branch on 32 bit addition
```

CCR Effects

N	Z	V	C
Δ	Δ	Δ	Δ

N: Set if bit 15 of the result is set; cleared otherwise.

Z: Set if the result is $0000 and Z was set before this operation; cleared otherwise.

V: Set if a two´s complement overflow resulted from the operation; cleared otherwise.
RS1[15] & RS2[15] & $\overline{RD[15]}_{new}$ | $\overline{RS1[15]}$ & $\overline{RS2[15]}$ & $RD[15]_{new}$

C: Set if there is a carry from bit 15 of the result; cleared otherwise.
RS1[15] & RS2[15] | RS1[15] & $\overline{RD[15]}_{new}$ | RS2[15] & $\overline{RD[15]}_{new}$

Code and CPU Cycles

Source Form	Address Mode	Machine Code										Cycles
ADC RD, RS1, RS2	TRI	0	0	0	1	1	RD	RS1	RS2	1	1	P

BFINSI Bit Field Insert and Invert BFINSI

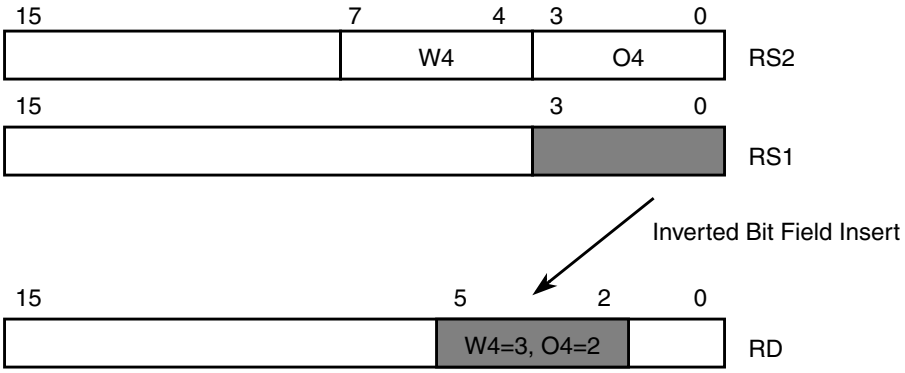
Operation

!RS1[*w*:*0*] ⇒ RD[*w*+o:o];

w = (RS2[7:4])

o = (RS2[3:0])

Extracts *w+1* bits from register RS1 starting at position 0, inverts them and writes into register RD starting at position *o*. The remaining bits in RD are not affected. If (o+w) > 15 the upper bits are ignored. Using R0 as a RS1, this command can be used to set bits.



CCR Effects

N	Z	V	C
Δ	Δ	0	—

N: Set if bit 15 of the result is set; cleared otherwise.

Z: Set if the result is $0000; cleared otherwise.

V: 0; cleared.

C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code										Cycles
BFINSI RD, RS1, RS2	TRI	0	1	1	1	0	RD	RS1	RS2	1	1	P

CPCH — Compare Immediate 8 bit Constant with Carry (High Byte) — CPCH

Operation

RS.H - IMM8 - C ⇒ NONE, only condition code flags get updated

Subtracts the carry bit and the 8 bit constant IMM8 contained in the instruction code from the high byte of the source register RD using binary subtraction and updates the condition code register accordingly. The carry bit and Zero bits are taken into account to allow a 16 bit compare in the form of

```
CMPL    R2,#LOWBYTE
CPCH    R2,#HIGHBYTE
BCC                         ; branch condition
```

Remark: There is no equivalent operation using triadic addressing. Comparing the values of two registers can be performed by using the subtract instruction with R0 as destination register.

CCR Effects

N	Z	V	C
Δ	Δ	Δ	Δ

N: Set if bit 15 of the result is set; cleared otherwise.

Z: Set if the result is $00 and Z was set before this operation; cleared otherwise.

V: Set if a two´s complement overflow resulted from the operation; cleared otherwise.
RS[15] & $\overline{\text{IMM8[7]}}$ & $\overline{\text{result[15]}}$ | $\overline{\text{RS[15]}}$ & IMM8[7] & result[15]

C: Set if there is a carry from the bit 15 of the result; cleared otherwise.
$\overline{\text{RS[15]}}$ & IMM8[7] | $\overline{\text{RS[15]}}$ & result[15] | IMM8[7] & result[15]

Code and CPU Cycles

Source Form	Address Mode	Machine Code							Cycles
CPCH RD, #IMM8	IMM8	1	1	0	1	1	RS	IMM8	P

Module Base + 0x0013

	7	6	5	4	3	2	1	0
R W	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 14-23. Timer Input Capture/Output Compare Register 1 Low (TC1)

Module Base + 0x0014

	15	14	13	12	11	10	9	8
R W	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

Figure 14-24. Timer Input Capture/Output Compare Register 2 High (TC2)

Module Base + 0x0015

	7	6	5	4	3	2	1	0
R W	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 14-25. Timer Input Capture/Output Compare Register 2 Low (TC2)

Module Base + 0x0016

	15	14	13	12	11	10	9	8
R W	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

Figure 14-26. Timer Input Capture/Output Compare Register 3 High (TC3)

Module Base + 0x0017

	7	6	5	4	3	2	1	0
R W	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 14-27. Timer Input Capture/Output Compare Register 3 Low (TC3)

Module Base + 0x0018

	15	14	13	12	11	10	9	8
R W	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

Figure 14-28. Timer Input Capture/Output Compare Register 4 High (TC4)

Table 14-23. MCCTL Field Descriptions (continued)

Field	Description
4 ICLAT	**Input Capture Force Latch Action** — When input capture latch mode is enabled (LATQ and BUFEN bit in ICSYS are set), a write one to this bit immediately forces the contents of the input capture registers TC0 to TC3 and their corresponding 8-bit pulse accumulators to be latched into the associated holding registers. The pulse accumulators will be automatically cleared when the latch action occurs. Writing zero to this bit has no effect. Read of this bit will always return zero.
3 FLMC	**Force Load Register into the Modulus Counter Count Register** — This bit is active only when the modulus down-counter is enabled (MCEN = 1). A write one into this bit loads the load register into the modulus counter count register (MCCNT). This also resets the modulus counter prescaler. Write zero to this bit has no effect. Read of this bit will return always zero.
2 MCEN	**Modulus Down-Counter Enable** 0 Modulus counter disabled. The modulus counter (MCCNT) is preset to 0xFFFF. This will prevent an early interrupt flag when the modulus down-counter is enabled. 1 Modulus counter is enabled.
1:0 MCPR[1:0]	**Modulus Counter Prescaler Select** — These two bits specify the division rate of the modulus counter prescaler when PRNT of TSCR1 is set to 0. The newly selected prescaler division rate will not be effective until a load of the load register into the modulus counter count register occurs.

Table 14-24. Modulus Counter Prescaler Select

MCPR1	MCPR0	Prescaler Division
0	0	1
0	1	4
1	0	8
1	1	16

14.3.2.20 16-Bit Modulus Down-Counter FLAG Register (MCFLG)

Module Base + 0x0027

	7	6	5	4	3	2	1	0
R	MCZF	0	0	0	POLF3	POLF2	POLF1	POLF0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 14-43. 16-Bit Modulus Down-Counter FLAG Register (MCFLG)

Read: Anytime

Write only used in the flag clearing mechanism for bit 7. Writing a one to bit 7 clears the flag. Writing a zero will not affect the current status of the bit.

16.3.3.3 Data Length Register (DLR)

This register keeps the data length field of the CAN frame.

Module Base + 0x00XC

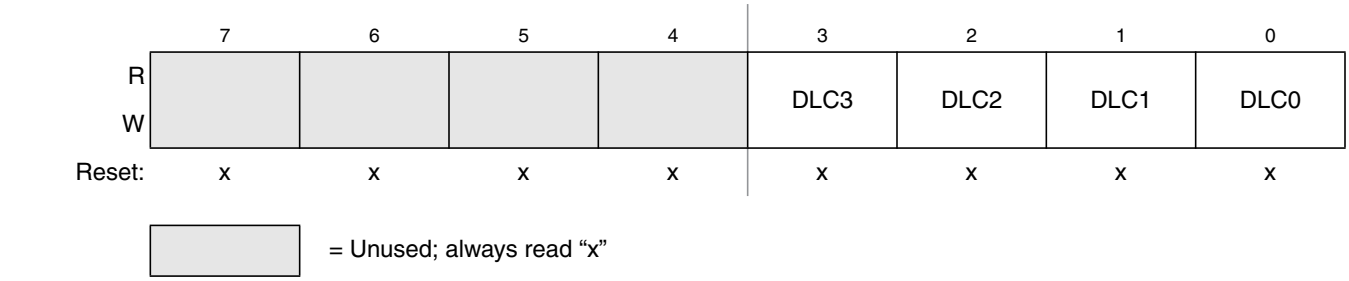


Figure 16-35. Data Length Register (DLR) — Extended Identifier Mapping

Table 16-34. DLR Register Field Descriptions

Field	Description
3-0 DLC[3:0]	**Data Length Code Bits** — The data length code contains the number of bytes (data byte count) of the respective message. During the transmission of a remote frame, the data length code is transmitted as programmed while the number of transmitted data bytes is always 0. The data byte count ranges from 0 to 8 for a data frame. Table 16-35 shows the effect of setting the DLC bits.

Table 16-35. 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

16.3.3.4 Transmit Buffer Priority Register (TBPR)

This register defines the local priority of the associated message buffer. The local priority is used for the internal prioritization process of the MSCAN and is defined to be highest for the smallest binary number. The MSCAN implements the following internal prioritization mechanisms:

- All transmission buffers with a cleared TXEx flag participate in the prioritization immediately before the SOF (start of frame) is sent.

16.4.5.5 MSCAN Sleep Mode

The CPU can request the MSCAN to enter this low power mode by asserting the SLPRQ bit in the CANCTL0 register. The time when the MSCAN enters sleep mode depends on a fixed synchronization delay and its current activity:

- If there are one or more message buffers scheduled for transmission (TXEx = 0), the MSCAN will continue to transmit until all transmit message buffers are empty (TXEx = 1, transmitted successfully or aborted) and then goes into sleep mode.
- If the MSCAN is receiving, it continues to receive and goes into sleep mode as soon as the CAN bus next becomes idle.
- If the MSCAN is neither transmitting nor receiving, it immediately goes into sleep mode.

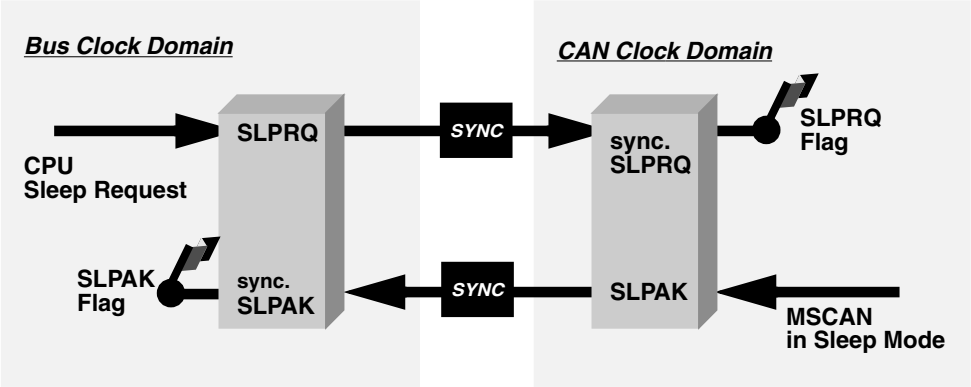


Figure 16-46. Sleep Request / Acknowledge Cycle

NOTE

The application software must avoid setting up a transmission (by clearing one or more TXEx flag(s)) and immediately request sleep mode (by setting SLPRQ). Whether the MSCAN starts transmitting or goes into sleep mode directly depends on the exact sequence of operations.

If sleep mode is active, the SLPRQ and SLPAK bits are set (Figure 16-46). The application software must use SLPAK as a handshake indication for the request (SLPRQ) to go into sleep mode.

When in sleep mode (SLPRQ = 1 and SLPAK = 1), the MSCAN stops its internal clocks. However, clocks that allow register accesses from the CPU side continue to run.

If the MSCAN is in bus-off state, it stops counting the 128 occurrences of 11 consecutive recessive bits due to the stopped clocks. TXCAN remains in a recessive state. If RXF = 1, the message can be read and RXF can be cleared. Shifting a new message into the foreground buffer of the receiver FIFO (RxFG) does not take place while in sleep mode.

It is possible to access the transmit buffers and to clear the associated TXE flags. No message abort takes place while in sleep mode.

- Run mode

  This is the basic mode of operation.
- Wait mode

  PIT operation in wait mode is controlled by the PITSWAI bit located in the PITCFLMT register. In wait mode, if the bus clock is globally enabled and if the PITSWAI bit is clear, the PIT operates like in run mode. In wait mode, if the PITSWAI bit is set, the PIT module is stalled.
- Stop mode

  In full stop mode or pseudo stop mode, the PIT module is stalled.
- Freeze mode

  PIT operation in freeze mode is controlled by the PITFRZ bit located in the PITCFLMT register. In freeze mode, if the PITFRZ bit is clear, the PIT operates like in run mode. In freeze mode, if the PITFRZ bit is set, the PIT module is stalled.

17.1.4 Block Diagram

Figure 17-1 shows a block diagram of the PIT module.

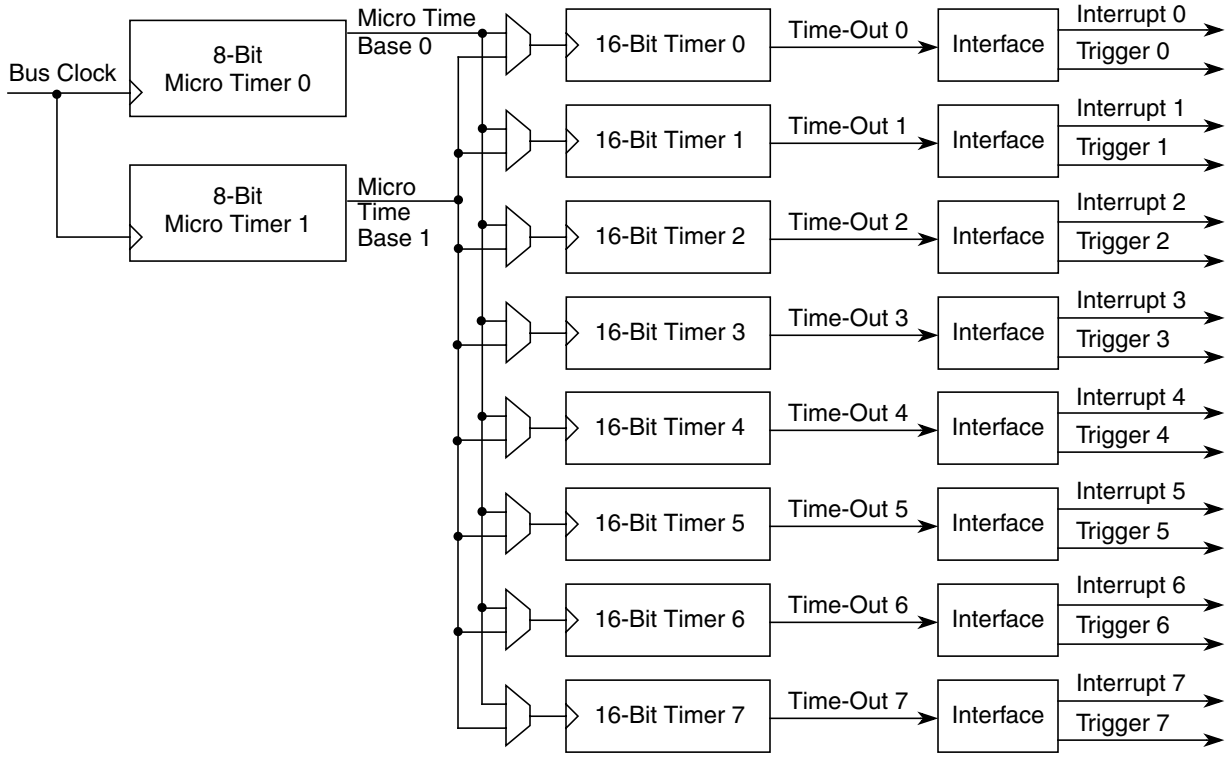


Figure 17-1. PIT24B8C Block Diagram

17.2 External Signal Description

The PIT module has no external pins.

20.3.2.8 SCI Status Register 2 (SCISR2)

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	AMAP	0	0	TXPOL	RXPOL	BRK13	TXDIR	RAF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 20-11. SCI Status Register 2 (SCISR2)

Read: Anytime

Write: Anytime

Table 20-12. SCISR2 Field Descriptions

Field	Description
7 AMAP	**Alternative Map** — This bit controls which registers sharing the same address space are accessible. In the reset condition the SCI behaves as previous versions. Setting AMAP=1 allows the access to another set of control and status registers and hides the baud rate and SCI control Register 1. 0 The registers labelled SCIBDH (0x0000),SCIBDL (0x0001), SCICR1 (0x0002) are accessible 1 The registers labelled SCIASR1 (0x0000),SCIACR1 (0x0001), SCIACR2 (0x00002) are accessible
4 TXPOL	**Transmit Polarity** — This bit control the polarity of the transmitted data. In NRZ format, a one is represented by a mark and a zero is represented by a space for normal polarity, and the opposite for inverted polarity. In IrDA format, a zero is represented by short high pulse in the middle of a bit time remaining idle low for a one for normal polarity, and a zero is represented by short low pulse in the middle of a bit time remaining idle high for a one for inverted polarity. 0 Normal polarity 1 Inverted polarity
3 RXPOL	**Receive Polarity** — This bit control the polarity of the received data. In NRZ format, a one is represented by a mark and a zero is represented by a space for normal polarity, and the opposite for inverted polarity. In IrDA format, a zero is represented by short high pulse in the middle of a bit time remaining idle low for a one for normal polarity, and a zero is represented by short low pulse in the middle of a bit time remaining idle high for a one for inverted polarity. 0 Normal polarity 1 Inverted polarity
2 BRK13	**Break Transmit Character Length** — This bit determines whether the transmit break character is 10 or 11 bit respectively 13 or 14 bits long. The detection of a framing error is not affected by this bit. 0 Break character is 10 or 11 bit long 1 Break character is 13 or 14 bit long
1 TXDIR	**Transmitter Pin Data Direction in Single-Wire Mode** — This bit determines whether the TXD pin is going to be used as an input or output, in the single-wire mode of operation. This bit is only relevant in the single-wire mode of operation. 0 TXD pin to be used as an input in single-wire mode 1 TXD pin to be used as an output in single-wire mode
0 RAF	**Receiver Active Flag** — RAF is set when the receiver detects a logic 0 during the RT1 time period of the start bit search. RAF is cleared when the receiver detects an idle character. 0 No reception in progress 1 Reception in progress

22.6.1 Channel [7:0] Interrupt (C[7:0]F)

This active high outputs will be asserted by the module to request a timer channel 7 – 0 interrupt to be serviced by the system controller.

22.6.2 Pulse Accumulator Input Interrupt (PAOVI)

This active high output will be asserted by the module to request a timer pulse accumulator input interrupt to be serviced by the system controller.

22.6.3 Pulse Accumulator Overflow Interrupt (PAOVF)

This active high output will be asserted by the module to request a timer pulse accumulator overflow interrupt to be serviced by the system controller.

22.6.4 Timer Overflow Interrupt (TOF)

This active high output will be asserted by the module to request a timer overflow interrupt to be serviced by the system controller.

Table 24-32. D-Flash Commands

FCMD	Command	Function on D-Flash Memory
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all D-Flash (and P-Flash) blocks and verifying that all D-Flash (and P-Flash) blocks are erased.
0x0D	Set User Margin Level	Specifies a user margin read level for the D-Flash block.
0x0E	Set Field Margin Level	Specifies a field margin read level for the D-Flash block (special modes only).
0x0F	Full Partition D-Flash	Erase the D-Flash block and partition an area of the D-Flash block for user access.
0x10	Erase Verify D-Flash Section	Verify that a given number of words starting at the address provided are erased.
0x11	Program D-Flash	Program up to four words in the D-Flash block.
0x12	Erase D-Flash Sector	Erase all bytes in a sector of the D-Flash block.
0x13	Enable EEPROM Emulation	Enable EEPROM emulation where writes to the buffer RAM EEE partition will be copied to the D-Flash EEE partition.
0x14	Disable EEPROM Emulation	Suspend all current erase and program activity related to EEPROM emulation but leave current EEE tags set.
0x15	EEPROM Emulation Query	Returns EEE partition and status variables.
0x20	Partition D-Flash	Partition an area of the D-Flash block for user access.

24.4.2 Flash Command Description

This section provides details of all available Flash commands launched by a command write sequence. The ACCERR bit in the FSTAT register will be set during the command write sequence if any of the following illegal steps are performed, causing the command not to be processed by the Memory Controller:

- Starting any command write sequence that programs or erases Flash memory before initializing the FCLKDIV register
- Writing an invalid command as part of the command write sequence
- For additional possible errors, refer to the error handling table provided for each command

If a Flash block is read during execution of an algorithm (CCIF = 0) on that same block, the read operation will return invalid data. If the SFDIF or DFDIF flags were not previously set when the invalid read operation occurred, both the SFDIF and DFDIF flags will be set and the FECCR registers will be loaded with the global address used in the invalid read operation with the data and parity fields set to all 0.

If the ACCERR or FPVIOL bits are set in the FSTAT register, the user must clear these bits before starting any command write sequence (see Section 24.3.2.7).

CAUTION

A Flash word or phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash word or phrase is not allowed.

Offset Module Base + 0x000A

	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Figure 25-16. Flash Common Command Object High Register (FCCOBHI)

Offset Module Base + 0x000B

	7	6	5	4	3	2	1	0
R W	CCOB[7:0]							
Reset	0	0	0	0	0	0	0	0

Figure 25-17. Flash Common Command Object Low Register (FCCOBLO)

25.3.2.11.1 FCCOB - NVM Command Mode

NVM command mode uses the indexed FCCOB register to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all FCCOB parameter fields are locked and cannot be changed by the user until the command completes (as evidenced by the Memory Controller returning CCIF to 1). Some commands return information to the FCCOB register array.

The generic format for the FCCOB parameter fields in NVM command mode is shown in Table 25-26. The return values are available for reading after the CCIF flag in the FSTAT register has been returned to 1 by the Memory Controller. Writes to the unimplemented parameter fields (CCOBIX = 110 and CCOBIX = 111) are ignored with reads from these fields returning 0x0000.

Table 25-26 shows the generic Flash command format. The high byte of the first word in the CCOB array contains the command code, followed by the parameters for this specific Flash command. For details on the FCCOB settings required by each command, see the Flash command descriptions in Section 25.4.2.

Table 25-26. FCCOB - NVM Command Mode (Typical Usage)

CCOBIX[2:0]	Byte	FCCOB Parameter Fields (NVM Command Mode)
000	HI	FCMD[7:0] defining Flash command
	LO	0, Global address [22:16]
001	HI	Global address [15:8]
	LO	Global address [7:0]
010	HI	Data 0 [15:8]
	LO	Data 0 [7:0]

Upon clearing CCIF to launch the Verify Backdoor Access Key command, the Memory Controller will check the FSEC KEYEN bits to verify that this command is enabled. If not enabled, the Memory Controller sets the ACCERR bit in the FSTAT register and terminates. If the command is enabled, the Memory Controller compares the key provided in FCCOB to the backdoor comparison key in the Flash configuration field with Key 0 compared to 0x7F_FF00, etc. If the backdoor keys match, security will be released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are aborted (set ACCERR) until a reset occurs. The CCIF flag is set after the Verify Backdoor Access Key operation has completed.

Table 25-56. Verify Backdoor Access Key Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 100 at command launch
		Set if a Load Data Field command sequence is currently active
		Set if an incorrect backdoor key is supplied
		Set if backdoor key access has not been enabled (KEYEN[1:0] != 10, see Section 25.3.2.2)
		Set if the backdoor key has mismatched since the last reset
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

25.4.2.13 Set User Margin Level Command

The Set User Margin Level command causes the Memory Controller to set the margin level for future read operations of a specific P-Flash or D-Flash block.

Table 25-57. Set User Margin Level Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0D	Global address [22:16] to identify the Flash block
001	Margin level setting	

Upon clearing CCIF to launch the Set User Margin Level command, the Memory Controller will set the user margin level for the targeted block and then set the CCIF flag.

Valid margin level settings for the Set User Margin Level command are defined in Table 25-58.

Table 25-58. Valid Set User Margin Level Settings

CCOB (CCOBIX=001)	Level Description
0x0000	Return to Normal Level
0x0001	User Margin-1 Level(1)

The security state out of reset can be permanently changed by programming the security byte of the Flash configuration field. This assumes that you are starting from a mode where the necessary P-Flash erase and program commands are available and that the upper region of the P-Flash is unprotected. If the Flash security byte is successfully programmed, its new value will take affect after the next MCU reset.

The following subsections describe these security-related subjects:

- Unsecuring the MCU using Backdoor Key Access
- Unsecuring the MCU in Special Single Chip Mode using BDM
- Mode and Security Effects on Flash Command Availability

26.5.1 Unsecuring the MCU using Backdoor Key Access

The MCU may be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor keys (four 16-bit words programmed at addresses 0x7F_FF00–0x7F_FF07). If the KEYEN[1:0] bits are in the enabled state (see Section 26.3.2.2), the Verify Backdoor Access Key command (see Section 26.4.2.12) allows the user to present four prospective keys for comparison to the keys stored in the Flash memory via the Memory Controller. If the keys presented in the Verify Backdoor Access Key command match the backdoor keys stored in the Flash memory, the SEC bits in the FSEC register (see Table 26-12) will be changed to unsecure the MCU. Key values of 0x0000 and 0xFFFF are not permitted as backdoor keys. While the Verify Backdoor Access Key command is active, P-Flash block 0 will not be available for read access and will return invalid data.

The user code stored in the P-Flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 26.3.2.2), the MCU can be unsecured by the backdoor key access sequence described below:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Section 26.4.2.12
2. If the Verify Backdoor Access Key command is successful, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 10

The Verify Backdoor Access Key command is monitored by the Memory Controller and an illegal key will prohibit future use of the Verify Backdoor Access Key command. A reset of the MCU is the only method to re-enable the Verify Backdoor Access Key command.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the sector containing the Flash security byte can be erased and the Flash security byte can be reprogrammed to the unsecure state, if desired.

In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0x7F_FF00–0x7F_FF07 in the Flash configuration field.

The security as defined in the Flash security byte (0x7F_FF0F) is not changed by using the Verify Backdoor Access Key command sequence. The backdoor keys stored in addresses 0x7F_FF00–0x7F_FF07 are unaffected by the Verify Backdoor Access Key command sequence. After the next reset of the MCU, the security state of the Flash module is determined by the Flash security byte

Table 27-4. Program IFR Fields

Global Address (PGMIFRON)	Size (Bytes)	Field Description
0x40_0000 – 0x40_0007	8	Device ID
0x40_0008 – 0x40_00E7	224	Reserved
0x40_00E8 – 0x40_00E9	2	Version ID
0x40_00EA – 0x40_00FF	22	Reserved
0x40_0100 – 0x40_013F	64	Program Once Field Refer to Section 27.4.2.7, “Program Once Command”
0x40_0140 – 0x40_01FF	192	Reserved

Table 27-5. P-Flash IFR Accessibility

Global Address (PGMIFRON)	Size (Bytes)	Accessed From
0x40_0000 – 0x40_01FF	512	XBUS0 (PBLK0S)(1)
0x40_0200 – 0x40_03FF	512	Unimplemented
0x40_0400 – 0x40_05FF	512	XBUS0 (PBLK1N)
0x40_0600 – 0x40_07FF	512	XBUS1 (PBLK1S)

1. Refer to Table 27-4 for more details.

Table 27-6. EEE Resource Fields

Global Address	Size (Bytes)	Description
0x10_0000 – 0x10_7FFF	32,768	D-Flash Memory (User and EEE)
0x10_8000 – 0x11_FFFF	98,304	Reserved
0x12_0000 – 0x12_007F	128	EEE Nonvolatile Information Register (EEEIFRON(1) = 1)
0x12_0080 – 0x12_0FFF	3,968	Reserved
0x12_1000 – 0x12_1EFF	3,840	Reserved
0x12_1F00 – 0x12_1FFF	256	EEE Tag RAM (TMGRAMON[1] = 1)
0x12_2000 – 0x12_3BFF	7,168	Reserved
0x12_3C00 – 0x12_3FFF	1,024	Memory Controller Scratch RAM (TMGRAMON[1] = 1)
0x12_4000 – 0x12_DFFF	40,960	Reserved
0x12_E000 – 0x12_FFFF	8,192	Reserved
0x13_0000 – 0x13_EFFF	61,440	Reserved
0x13_F000 – 0x13_FFFF	4,096	Buffer RAM (User and EEE)

1. MMCCTL1 register bit

Table 27-14. FECCRIX Field Descriptions

Field	Description
2-0 ECCRIX[2:0]	**ECC Error Register Index**— The ECCRIX bits are used to select which word of the FECCR register array is being read. See Section 27.3.2.13, “Flash ECC Error Results Register (FECCR),” for more details.

27.3.2.5 Flash Configuration Register (FCNFG)

The FCNFG register enables the Flash command complete interrupt and forces ECC faults on Flash array read access from the CPU or XGATE.

Offset Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 27-9. Flash Configuration Register (FCNFG)

CCIE, IGNSF, FDFD, and FSFD bits are readable and writable while remaining bits read 0 and are not writable.

Table 27-15. FCNFG Field Descriptions

Field	Description
7 CCIE	**Command Complete Interrupt Enable** — The CCIE bit controls interrupt generation when a Flash command has completed. 0 Command complete interrupt disabled 1 An interrupt will be requested whenever the CCIF flag in the FSTAT register is set (see Section 27.3.2.7)
4 IGNSF	**Ignore Single Bit Fault** — The IGNSF controls single bit fault reporting in the FERSTAT register (see Section 27.3.2.8). 0 All single bit faults detected during array reads are reported 1 Single bit faults detected during array reads are not reported and the single bit fault interrupt will not be generated

0x0368–0x037F Port Integration Module (PIM) Map 6 of 6

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0368	PTR	R	PTR7	PTR6	PTR5	PTR4	PTR3	PTR2	PTR1	PTR0
		W								
0x0369	PTIR	R	PTIR7	PTIR6	PTIR5	PTIR4	PTIR3	PTIR2	PTIR1	PTIR0
		W								
0x036A	DDRR	R	DDRR7	DDRR7	DDRR5	DDRR4	DDRR3	DDRR2	DDRR1	DDRR0
		W								
0x036B	RDRR	R	RDRR7	RDRR6	RDRR5	RDRR4	RDRR3	RDRR2	RDRR1	RDRR0
		W								
0x036C	PERR	R	PERR7	PERR6	PERR5	PERR4	PERR3	PERR2	PERR1	PERR0
		W								
0x036D	PPSR	R	PPSR7	PPSR6	PPSR5	PPSR4	PPSR3	PPSR2	PPSR1	PPSR0
		W								
0x036E	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x036F	PTRRR	R	PTRRR7	PTRRR6	PTRRR5	PTRRR4	PTRRR3	PTRRR2	PTRRR1	PTRRR0
		W								
0x0370	PTL	R	PTL7	PTL6	PTL5	PTL4	PTL3	PTL2	PTL1	PTL0
		W								
0x0371	PTIL	R	PTIL7	PTIL6	PTIL5	PTIL4	PTIL3	PTIL2	PTIL1	PTIL0
		W								
0x0372	DDRL	R	DDRL7	DDRL7	DDRL5	DDRL4	DDRL3	DDRL2	DDRL1	DDRL0
		W								
0x0373	RDRL	R	RDRL7	RDRL6	RDRL5	RDRL4	RDRL3	RDRL2	RDRL1	RDRL0
		W								
0x0374	PERL	R	PERL7	PERL6	PERL5	PERL4	PERL3	PERL2	PERL1	PERL0
		W								
0x0375	PPSL	R	PPSL7	PPSL6	PPSL5	PPSL4	PPSL3	PPSL2	PPSL1	PPSL0
		W								
0x0376	WOML	R	WOML7	WOML6	WOML5	WOML4	WOML3	WOML2	WOML1	WOML0
		W								
0x0377	PTLRR	R	PTLRR7	PTLRR6	PTLRR5	PTLRR4	0	0	0	0
		W								
0x0378	PTF	R	PTF7	PTF6	PTF5	PTF4	PTF3	PTF2	PTF1	PTF0
		W								
0x0379	PTIF	R	PTIF7	PTIF6	PTIF5	PTIF4	PTIF3	PTIF2	PTIF1	PTIF0
		W								
0x037A	DDRF	R	DDRF7	DDRF7	DDRF5	DDRF4	DDRF3	DDRF2	DDRF1	DDRF0
		W								
0x037B	RDRF	R	RDRF7	RDRF6	RDRF5	RDRF4	RDRF3	RDRF2	RDRF1	RDRF0
		W								
0x037C	PERF	R	PERF7	PERF6	PERF5	PERF4	PERF3	PERF2	PERF1	PERF0
		W								