



Welcome to [E-XFL.COM](http://E-XFL.COM)

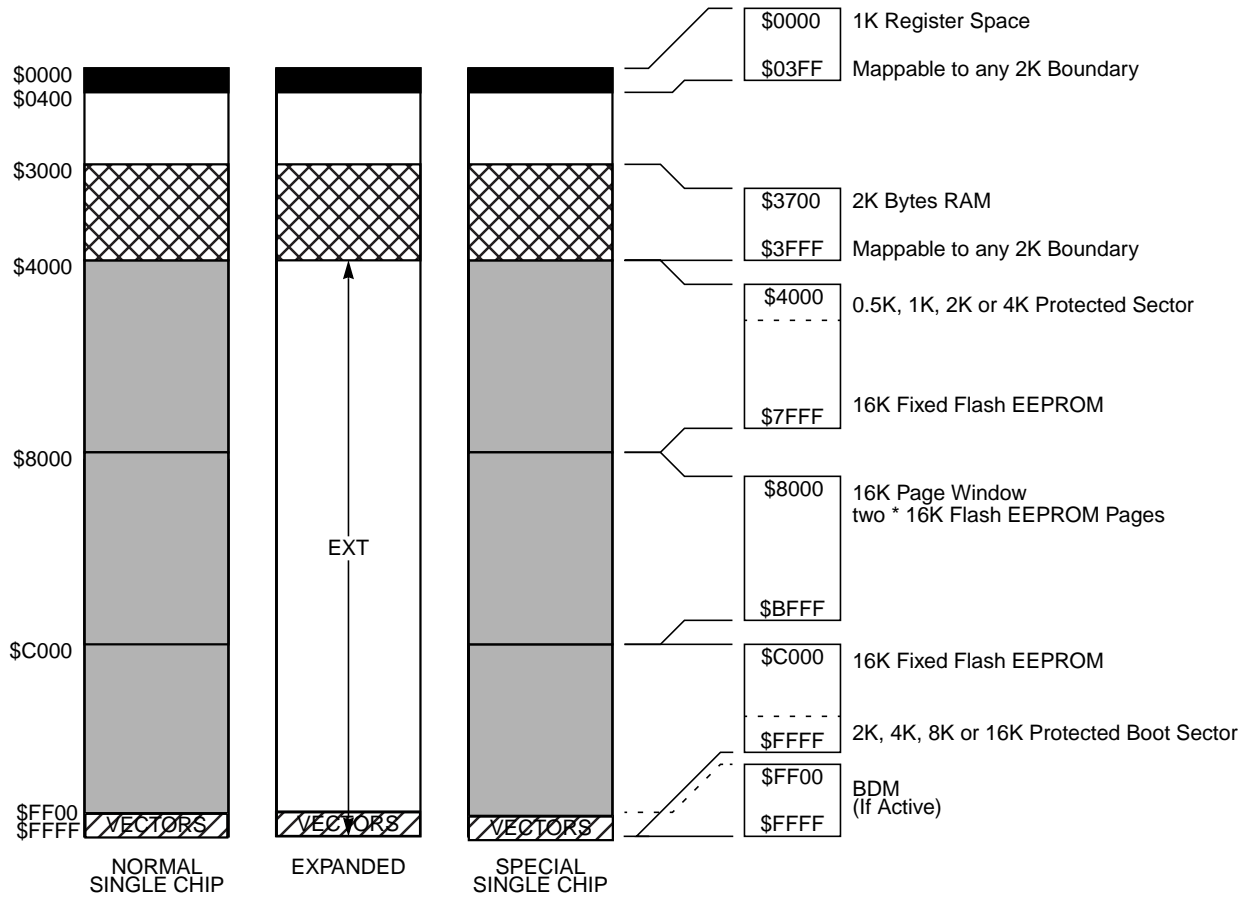
### What is "[Embedded - Microcontrollers](#)"?

"[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	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, I <sup>2</sup> C, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	91
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 2.75V
Data Converters	A/D 16x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12e128cpve">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12e128cpve</a>



The figure shows a useful map, which is not the map out of reset. After reset the map is:

- \$0000-\$03FF: Register Space
- \$0000-\$07FF: 2K RAM (only 1K RAM visible \$0400-\$07FF)

**Figure 1-4. MC9S12E32 User Configurable Memory Map**

**0x00D0 – 0x00D7 SCI1 (Asynchronous Serial Interface) (continued)**

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00D5	SCISR2	R	0	0	0	TXPOL <sup>1</sup>	RXPOL <sup>1</sup>	BRK13	TXDIR	RAF
		W								
0x00D6	SCIDRH	R	R8	T8	0	0	0	0	0	0
		W								
0x00D7	SCIDRL	R	R7	R6	R5	R4	R3	R2	R1	R0
		W	T7	T6	T5	T4	T3	T2	T1	T0

<sup>1</sup> TXPOL and RXPOL are available in version V04 of SCI

**0x00D8 – 0x00DF SPI (Serial Peripheral Interface)**

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00D8	SPICR1	R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
		W								
0x00D9	SPICR2	R	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
		W								
0x00DA	SPIBR	R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
		W								
0x00DB	SPISR	R	SPIF	0	SPTEF	MODF	0	0	0	0
		W								
0x00DC	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00DD	SPIDR	R	Bit7	6	5	4	3	2	1	Bit0
		W								
0x00DE	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00DF	Reserved	R	0	0	0	0	0	0	0	0
		W								

**0x00E0 – 0x00E7 IIC (Inter-IC Bus)**

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00E0	IBAD	R	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	0
		W								
0x00E1	IBFD	R	IBC7	IBC6	IBC5	IBC4	IBC3	IBC2	IBC1	IBC0
		W								
0x00E2	IBCR	R	IBEN	IBIE	MS/SL	Tx/Rx	TXAK	0	0	IBSWAI
		W						RSTA		
0x00E3	IBSR	R	TCF	IAAS	IBB	IBAL	0	SRW	IBIF	RXAK
		W								

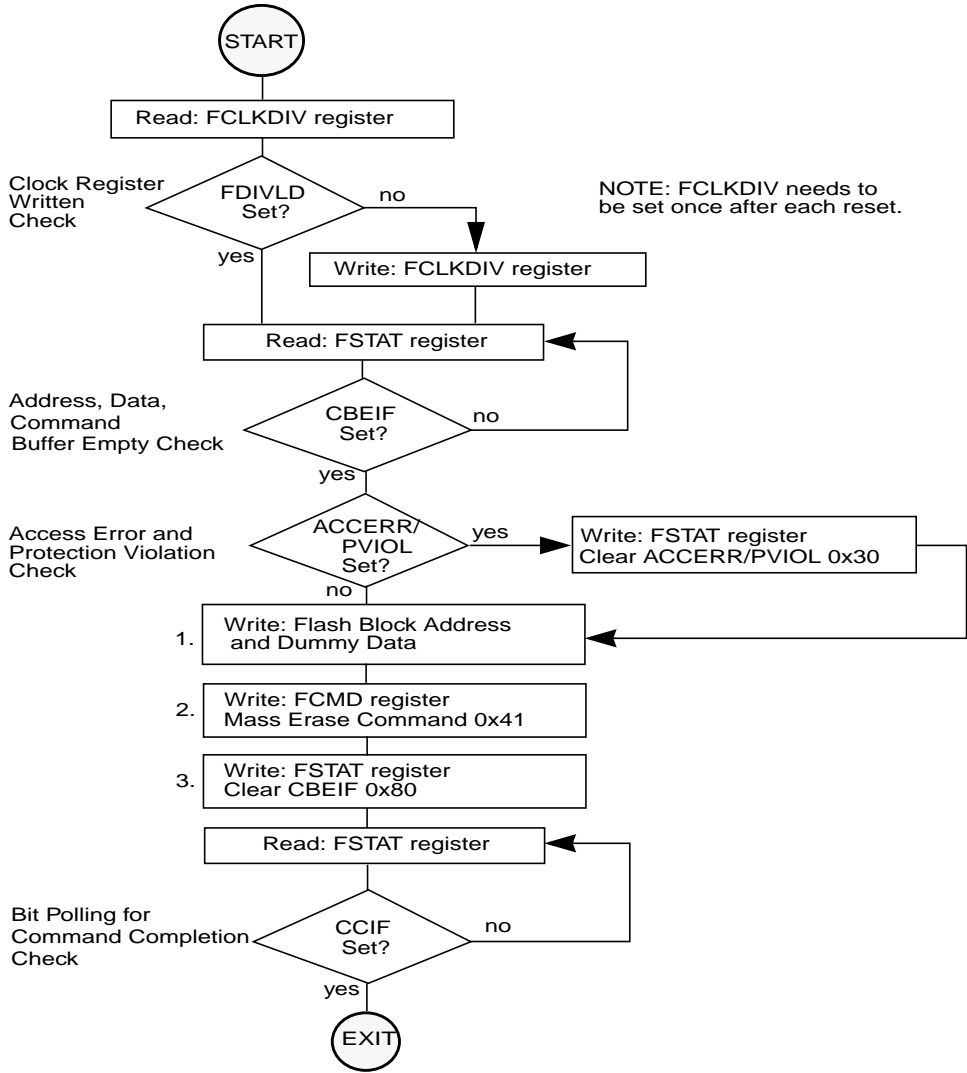


Figure 2-25. Example Mass Erase Command Flow

### 3.3.5.4 Port S Reduced Drive Register (RDRS)

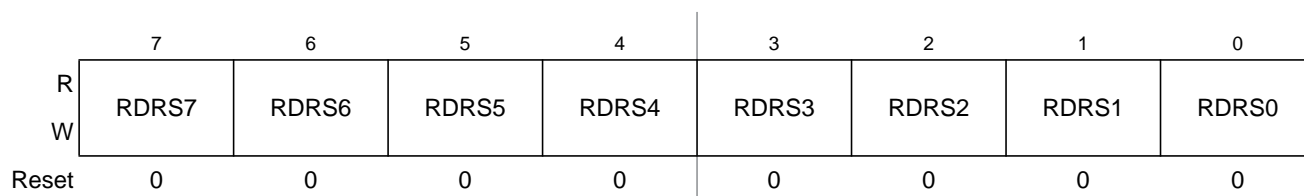


Figure 3-32. Port S Reduced Drive Register (RDRS)

Read: Anytime. Write: Anytime.

This register configures the drive strength of configured output pins as either full or reduced. If a pin is configured as input, the corresponding Reduced Drive Register bit has no effect.

Table 3-23. RDRS Field Descriptions

Field	Description
7:0 RDRS[7:0]	<b>Reduced Drive Port S</b> 0 Full drive strength at output. 1 Associated pin drives at about 1/3 of the full drive strength.

### 3.3.5.5 Port S Pull Device Enable Register (PERS)

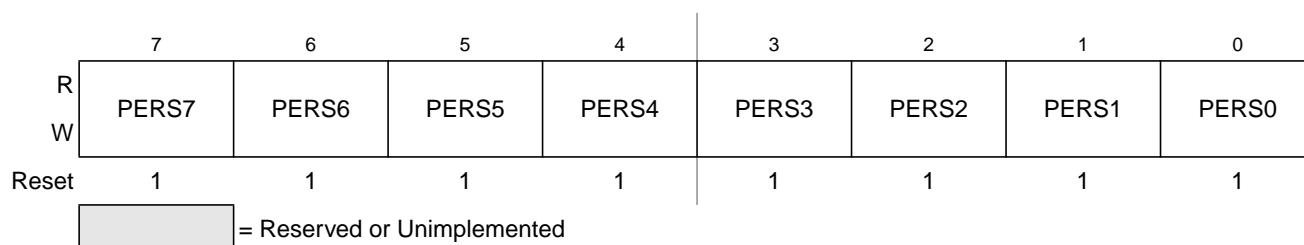


Figure 3-33. Port S Pull Device Enable Register (PERS)

Read: Anytime. Write: Anytime.

This register configures whether a pull-up or a pull-down device is activated on configured input or wired-or (open drain) output pins. If a pin is configured as push-pull output, the corresponding Pull Device Enable Register bit has no effect.

Table 3-24. PERS Field Descriptions

Field	Description
7:0 PERS[7:0]	<b>Pull Device Enable Port S</b> 0 Pull-up or pull-down device is disabled. 1 Pull-up or pull-down device is enabled.

## 3.4 Functional Description

Each pin associated with ports AD, M, P, Q, S, T and U can act as general-purpose I/O. In addition the pin can act as an output from a peripheral module or an input to a peripheral module.

A set of configuration registers is common to all ports. All registers can be written at any time, however a specific configuration might not become active.

Example: Selecting a pull-up resistor. This resistor does not become active while the port is used as a push-pull output.

### 3.4.1 I/O Register

The I/O Register holds the value driven out to the pin if the port is used as a general-purpose I/O. Writing to the I/O Register only has an effect on the pin if the port is used as general-purpose output.

When reading the I/O Register, the value of each pin is returned if the corresponding Data Direction Register bit is set to 0 (pin configured as input). If the data direction register bits is set to 1, the content of the I/O Register bit is returned. This is independent of any other configuration (Figure 3-49).

Due to internal synchronization circuits, it can take up to 2 bus cycles until the correct value is read on the I/O Register when changing the data direction register.

### 3.4.2 Input Register

The Input Register is a read-only register and generally returns the value of the pin (Figure 3-49). It can be used to detect overload or short circuit conditions.

Due to internal synchronization circuits, it can take up to 2 bus cycles until the correct value is read on the Input Register when changing the Data Direction Register.

### 3.4.3 Data Direction Register

The Data Direction Register defines whether the pin is used as an input or an output. A Data Direction Register bit set to 0 configures the pin as an input. A Data Direction Register bit set to 1 configures the pin as an output. If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 3-49).

**NOTE**

Register address = base address + address offset, where the base address is defined at the MCU level and the address offset is defined at the module level.

### 4.3.2 Register Descriptions

This section describes in address order all the CRGV4 registers and their individual bits.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
SYNR	R	0	0	SYN5	SYN4	SYN3	SYN2	SYN1	SYN0
	W								
REFDV	R	0	0	0	0	REFDV3	REFDV2	REFDV1	REFDV0
	W								
CTFLG	R	0	0	0	0	0	0	0	0
	W								
CRGFLG	R	RTIF	PORF	LVRF	LOCKIF	LOCK	TRACK	SCMIF	SCM
	W								
CRGINT	R	RTIE	0	0	LOCKIE	0	0	SCMIE	0
	W								
CLKSEL	R	PLLSEL	PSTP	SYSWAI	ROAWAI	PLLWAI	CWAI	RTIWAI	COPWAI
	W								
PLLCTL	R	CME	PLLON	AUTO	ACQ	0	PRE	PCE	SCME
	W								
RTICTL	R	0	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
	W								
COPCTL	R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
	W								
FORBYP	R	0	0	0	0	0	0	0	0
	W								
CTCTL	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

**Figure 4-3. CRG Register Summary**

### 4.4.6 Real-Time Interrupt (RTI)

The RTI can be used to generate a hardware interrupt at a fixed periodic rate. If enabled (by setting RTIE=1), this interrupt will occur at the rate selected by the RTICTL register. The RTI runs with a gated OSCCLK (see Section Figure 4-22., “Clock Chain for RTI”). At the end of the RTI time-out period the RTIF flag is set to 1 and a new RTI time-out period starts immediately.

A write to the RTICTL register restarts the RTI time-out period.

If the PRE bit is set, the RTI will continue to run in pseudo-stop mode.

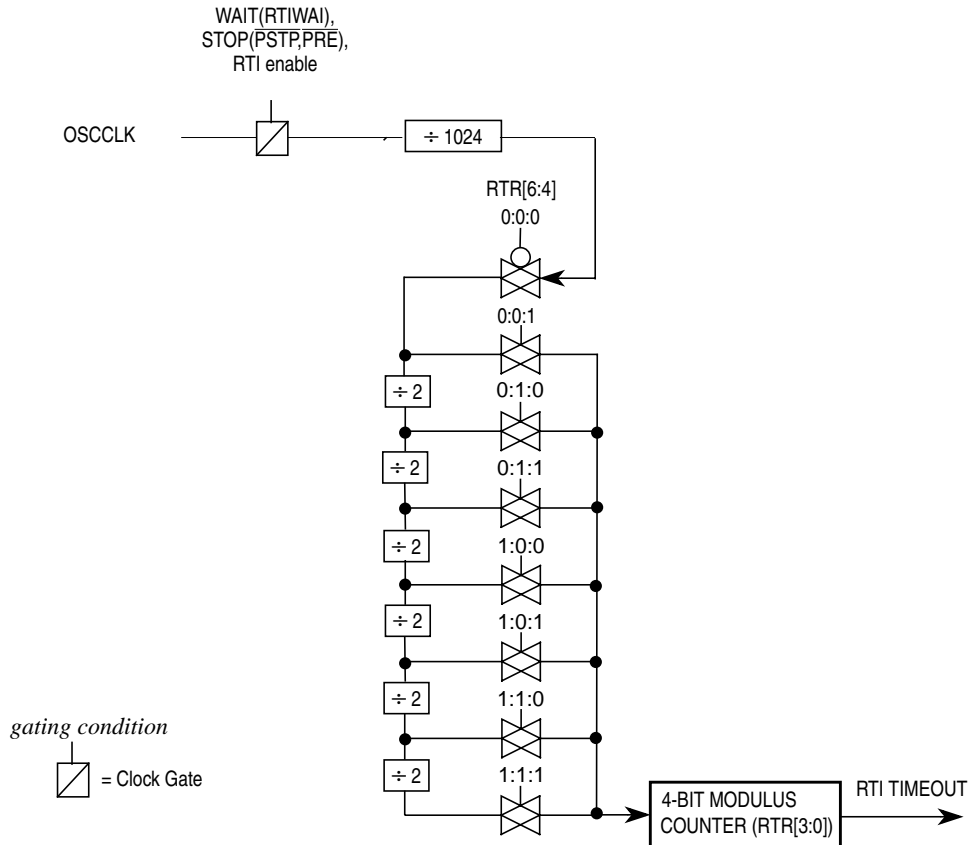


Figure 4-22. Clock Chain for RTI

### 4.4.7 Modes of Operation

#### 4.4.7.1 Normal Mode

The CRGV4 block behaves as described within this specification in all normal modes.

#### 4.4.7.2 Self-Clock Mode

The VCO has a minimum operating frequency,  $f_{SCM}$ . If the external clock frequency is not available due to a failure or due to long crystal start-up time, the bus clock and the core clock are derived from the VCO



If the MCU gets an external reset during pseudo-stop mode active, the CRG asynchronously restores all configuration bits in the register space to its default settings and starts the reset generator. After completing the reset sequence processing begins by fetching the normal reset vector. Pseudo-stop mode is exited and the MCU is in run mode again.

If the clock monitor is enabled ( $CME = 1$ ) the MCU is able to leave pseudo-stop mode when loss of oscillator/external clock is detected by a clock monitor fail. If the SCME bit is not asserted the CRG generates a clock monitor fail reset (CMRESET). The CRG's behavior for CMRESET is the same compared to external reset, but another reset vector is fetched after completion of the reset sequence. If the SCME bit is asserted the CRG generates a SCM interrupt if enabled ( $SCMIE=1$ ). After generating the interrupt the CRG enters self-clock mode and starts the clock quality checker (see [Section 4.4.4, "Clock Quality Checker"](#)). Then the MCU continues with normal operation. If the SCM interrupt is blocked by  $SCMIE = 0$ , the SCMIF flag will be asserted but the CRG will not wake-up from pseudo-stop mode.

If any other interrupt source (e.g. RTI) triggers exit from pseudo-stop mode the MCU immediately continues with normal operation. Because the PLL has been powered-down during stop mode the PLLSEL bit is cleared and the MCU runs on OSCCLK after leaving stop mode. The software must set the PLLSEL bit again, in order to switch system and core clocks to the PLLCLK.

[Table 4-12](#) summarizes the outcome of a clock loss while in pseudo-stop mode.

In either level or edge triggered modes, the first conversion begins when the trigger is received. In both cases, the maximum latency time is one bus clock cycle plus any skew or delay introduced by the trigger circuitry.

After ETRIGE is enabled, conversions cannot be started by a write to ATDCTL5, but rather must be triggered externally.

If the level mode is active and the external trigger both de-asserts and re-asserts itself during a conversion sequence, this does not constitute an overrun. Therefore, the flag is not set. If the trigger remains asserted in level mode while a sequence is completing, another sequence will be triggered immediately.

### 6.4.2.2 General-Purpose Digital Input Port Operation

The input channel pins can be multiplexed between analog and digital data. As analog inputs, they are multiplexed and sampled to supply signals to the A/D converter. As digital inputs, they supply external input data that can be accessed through the digital port registers (PORTAD0 & PORTAD1) (input-only).

The analog/digital multiplex operation is performed in the input pads. The input pad is always connected to the analog inputs of the ATD10B16C. The input pad signal is buffered to the digital port registers. This buffer can be turned on or off with the ATDDIEN0 & ATDDIEN1 register. This is important so that the buffer does not draw excess current when analog potentials are presented at its input.

### 6.4.3 Operation in Low Power Modes

The ATD10B16C can be configured for lower MCU power consumption in three different ways:

- **Stop Mode**  
 Stop Mode: This halts A/D conversion. Exit from Stop mode will resume A/D conversion, But due to the recovery time the result of this conversion should be ignored.  
 Entering stop mode causes all clocks to halt and thus the system is placed in a minimum power standby mode. This halts any conversion sequence in progress. During recovery from stop mode, there must be a minimum delay for the stop recovery time  $t_{SR}$  before initiating a new ATD conversion sequence.
- **Wait Mode**  
 Wait Mode with AWAI = 1: This halts A/D conversion. Exit from Wait mode will resume A/D conversion, but due to the recovery time the result of this conversion should be ignored.  
 Entering wait mode, the ATD conversion either continues or halts for low power depending on the logical value of the AWAIT bit.
- **Freeze Mode**  
 Writing ADPU = 0 (Note that all ATD registers remain accessible.): This aborts any A/D conversion in progress.  
 In freeze mode, the ATD10B16C will behave according to the logical values of the FRZ1 and FRZ0 bits. This is useful for debugging and emulation.

#### NOTE

The reset value for the ADPU bit is zero. Therefore, when this module is reset, it is reset into the power down state.

### 8.3.2.3 SCI Control Register 2 (SCICR2)

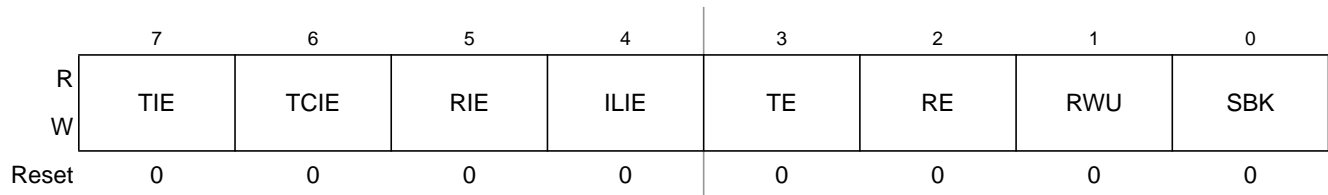


Figure 8-6. SCI Control Register 2 (SCICR2)

Read: anytime

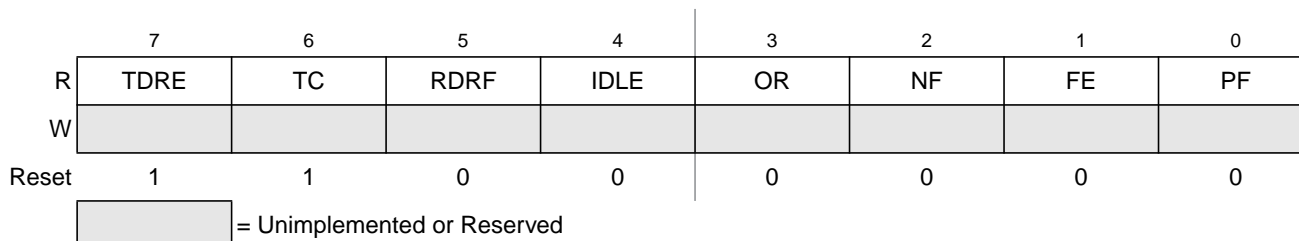
Write: anytime

Table 8-6. SCICR2 Field Descriptions

Field	Description
7 TIE	<b>Transmitter Interrupt Enable Bit</b> — TIE enables the transmit data register empty flag, TDRE, to generate interrupt requests. 0 TDRE interrupt requests disabled 1 TDRE interrupt requests enabled
6 TCIE	<b>Transmission Complete Interrupt Enable Bit</b> — TCIE enables the transmission complete flag, TC, to generate interrupt requests. 0 TC interrupt requests disabled 1 TC interrupt requests enabled
5 RIE	<b>Receiver Full Interrupt Enable Bit</b> — RIE enables the receive data register full flag, RDRF, or the overrun flag, OR, to generate interrupt requests. 0 RDRF and OR interrupt requests disabled 1 RDRF and OR interrupt requests enabled
4 ILIE	<b>Idle Line Interrupt Enable Bit</b> — ILIE enables the idle line flag, IDLE, to generate interrupt requests. 0 IDLE interrupt requests disabled 1 IDLE interrupt requests enabled
3 TE	<b>Transmitter Enable Bit</b> — TE enables the SCI transmitter and configures the TXD pin as being controlled by the SCI. The TE bit can be used to queue an idle preamble. 0 Transmitter disabled 1 Transmitter enabled
2 RE	<b>Receiver Enable Bit</b> — RE enables the SCI receiver. 0 Receiver disabled 1 Receiver enabled
1 RWU	<b>Receiver Wakeup Bit</b> — Standby state 0 Normal operation. 1 RWU enables the wakeup function and inhibits further receiver interrupt requests. Normally, hardware wakes the receiver by automatically clearing RWU.
0 SBK	<b>Send Break Bit</b> — Toggling SBK sends one break character (10 or 11 logic 0s, respectively 13 or 14 logics 0s if BRK13 is set). Toggling implies clearing the SBK bit before the break character has finished transmitting. As long as SBK is set, the transmitter continues to send complete break characters (10 or 11 bits, respectively 13 or 14 bits). 0 No break characters 1 Transmit break characters

### 8.3.2.4 SCI Status Register 1 (SCISR1)

The SCISR1 and SCISR2 registers provide inputs to the MCU for generation of SCI interrupts. Also, these registers can be polled by the MCU to check the status of these bits. The flag-clearing procedures require that the status register be read followed by a read or write to the SCI data register. It is permissible to execute other instructions between the two steps as long as it does not compromise the handling of I/O. Note that the order of operations is important for flag clearing.



**Figure 8-7. SCI Status Register 1 (SCISR1)**

Read: anytime

Write: has no meaning or effect

**Table 8-7. SCISR1 Field Descriptions**

Field	Description
7 TDRE	<b>Transmit Data Register Empty Flag</b> — TDRE is set when the transmit shift register receives a byte from the SCI data register. When TDRE is 1, the transmit data register (SCIDRH/L) is empty and can receive a new value to transmit. Clear TDRE by reading SCI status register 1 (SCISR1), with TDRE set and then writing to SCI data register low (SCIDRL). 0 No byte transferred to transmit shift register 1 Byte transferred to transmit shift register; transmit data register empty
6 TC	<b>Transmit Complete Flag</b> — TC is set low when there is a transmission in progress or when a preamble or break character is loaded. TC is set high when the TDRE flag is set and no data, preamble, or break character is being transmitted. When TC is set, the TXD pin becomes idle (logic 1). Clear TC by reading SCI status register 1 (SCISR1) with TC set and then writing to SCI data register low (SCIDRL). TC is cleared automatically when data, preamble, or break is queued and ready to be sent. TC is cleared in the event of a simultaneous set and clear of the TC flag (transmission not complete). 0 Transmission in progress 1 No transmission in progress
5 RDRF	<b>Receive Data Register Full Flag</b> — RDRF is set when the data in the receive shift register transfers to the SCI data register. Clear RDRF by reading SCI status register 1 (SCISR1) with RDRF set and then reading SCI data register low (SCIDRL). 0 Data not available in SCI data register 1 Received data available in SCI data register
4 IDLE	<b>Idle Line Flag<sup>1</sup></b> — IDLE is set when 10 consecutive logic 1s (if M = 0) or 11 consecutive logic 1s (if M = 1) appear on the receiver input. After the IDLE flag is cleared, a valid frame must again set the RDRF flag before an idle condition can set the IDLE flag. Clear IDLE by reading SCI status register 1 (SCISR1) with IDLE set and then reading SCI data register low (SCIDRL). 0 Receiver input is either active now or has never become active since the IDLE flag was last cleared 1 Receiver input has become idle

Figure 8-20 shows a burst of noise near the beginning of the start bit that resets the RT clock. The sample after the reset is low but is not preceded by three high samples that would qualify as a falling edge. Depending on the timing of the start bit search and on the data, the frame may be missed entirely or it may set the framing error flag.

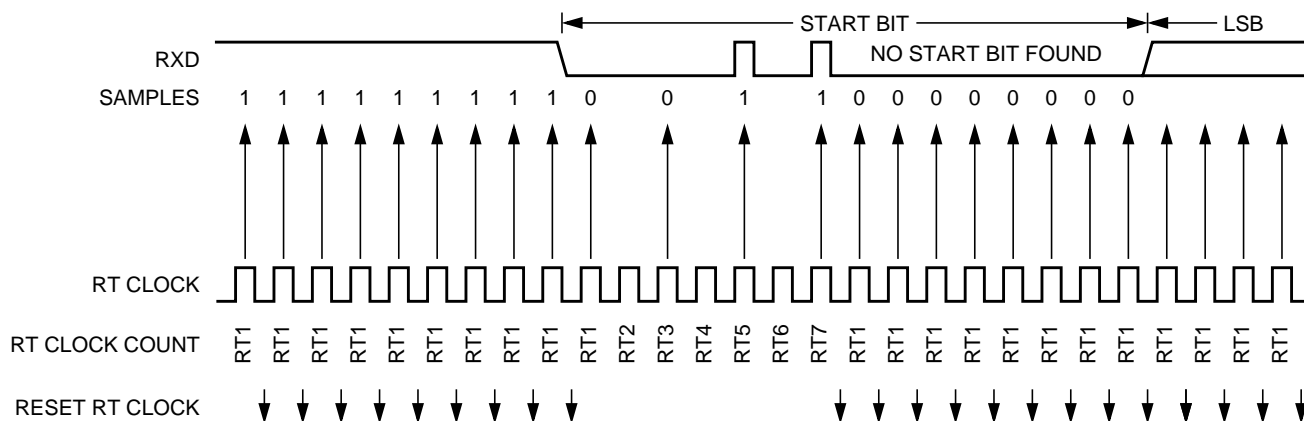


Figure 8-20. Start Bit Search Example 5

In Figure 8-21, a noise burst makes the majority of data samples RT8, RT9, and RT10 high. This sets the noise flag but does not reset the RT clock. In start bits only, the RT8, RT9, and RT10 data samples are ignored.

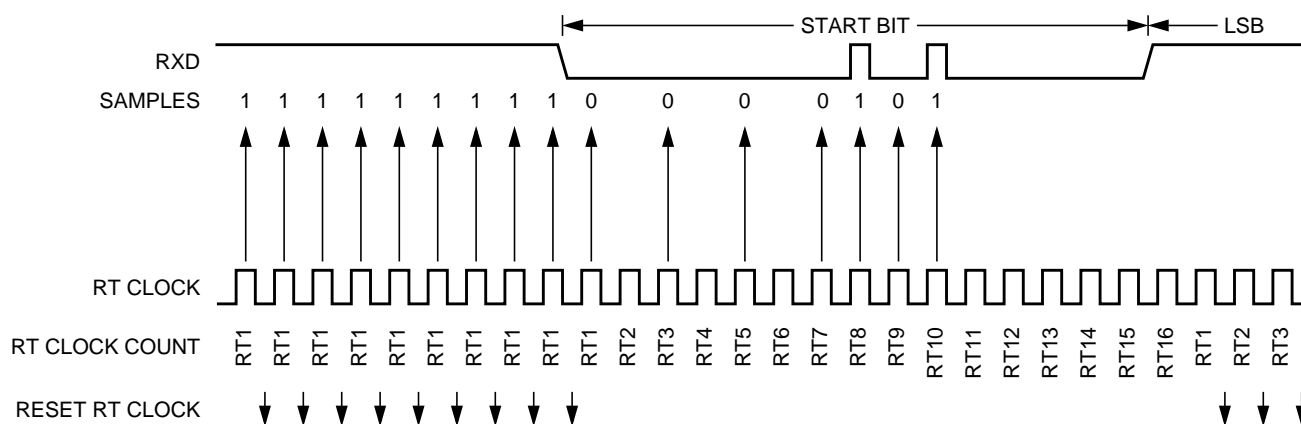


Figure 8-21. Start Bit Search Example 6

### 8.4.5.4 Framing Errors

If the data recovery logic does not detect a logic 1 where the stop bit should be in an incoming frame, it sets the framing error flag, FE, in SCI status register 1 (SCISR1). A break character also sets the FE flag because a break character has no stop bit. The FE flag is set at the same time that the RDRF flag is set.

Table 10-7. IBCR Field Descriptions

Field	Description
7 IBEN	<p><b>I-Bus Enable</b> — This bit controls the software reset of the entire IIC bus module.</p> <p>0 The module is reset and disabled. This is the power-on reset situation. When low the interface is held in reset but registers can be accessed</p> <p>1 The IIC bus module is enabled. This bit must be set before any other IBCR bits have any effect</p> <p>If the IIC bus module is enabled in the middle of a byte transfer the interface behaves as follows: slave mode ignores the current transfer on the bus and starts operating whenever a subsequent start condition is detected. Master mode will not be aware that the bus is busy, hence if a start cycle is initiated then the current bus cycle may become corrupt. This would ultimately result in either the current bus master or the IIC bus module losing arbitration, after which bus operation would return to normal.</p>
6 IBIE	<p><b>I-Bus Interrupt Enable</b></p> <p>0 Interrupts from the IIC bus module are disabled. Note that this does not clear any currently pending interrupt condition</p> <p>1 Interrupts from the IIC bus module are enabled. An IIC bus interrupt occurs provided the IBIF bit in the status register is also set.</p>
5 MS/SL	<p><b>Master/Slave Mode Select Bit</b> — Upon reset, this bit is cleared. When this bit is changed from 0 to 1, a START signal is generated on the bus, and the master mode is selected. When this bit is changed from 1 to 0, a STOP signal is generated and the operation mode changes from master to slave. A STOP signal should only be generated if the IBIF flag is set. MS/SL is cleared without generating a STOP signal when the master loses arbitration.</p> <p>0 Slave Mode</p> <p>1 Master Mode</p>
4 Tx/Rx	<p><b>Transmit/Receive Mode Select Bit</b> — This bit selects the direction of master and slave transfers. When addressed as a slave this bit should be set by software according to the SRW bit in the status register. In master mode this bit should be set according to the type of transfer required. Therefore, for address cycles, this bit will always be high.</p> <p>0 Receive</p> <p>1 Transmit</p>
3 TXAK	<p><b>Transmit Acknowledge Enable</b> — This bit specifies the value driven onto SDA during data acknowledge cycles for both master and slave receivers. The IIC module will always acknowledge address matches, provided it is enabled, regardless of the value of TXAK. Note that values written to this bit are only used when the IIC bus is a receiver, not a transmitter.</p> <p>0 An acknowledge signal will be sent out to the bus at the 9th clock bit after receiving one byte data</p> <p>1 No acknowledge signal response is sent (i.e., acknowledge bit = 1)</p>
2 RSTA	<p><b>Repeat Start</b> — Writing a 1 to this bit will generate a repeated START condition on the bus, provided it is the current bus master. This bit will always be read as a low. Attempting a repeated start at the wrong time, if the bus is owned by another master, will result in loss of arbitration.</p> <p>1 Generate repeat start cycle</p>
1 RESERVED	<p><b>Reserved</b> — Bit 1 of the IBCR is reserved for future compatibility. This bit will always read 0.</p>
0 IBSWAI	<p><b>I Bus Interface Stop in Wait Mode</b></p> <p>0 IIC bus module clock operates normally</p> <p>1 Halt IIC bus module clock generation in wait mode</p>

Wait mode is entered via execution of a CPU WAI instruction. In the event that the IBSWAI bit is set, all clocks internal to the IIC will be stopped and any transmission currently in progress will halt. If the CPU were woken up by a source other than the IIC module, then clocks would restart and the IIC would resume

# Chapter 11

## Pulse Width Modulator with Fault Protection (PMF15B6CV2)

### 11.1 Introduction

The Pulse width Modulator with Fault protection (PMF) module can be configured for one, two, or three complementary pairs. For example:

- One complementary pair and four independent PWM outputs
- Two complementary pair and two independent PWM outputs
- Three complementary pair and zero independent PWM outputs
- Zero complementary pair and six independent PWM outputs

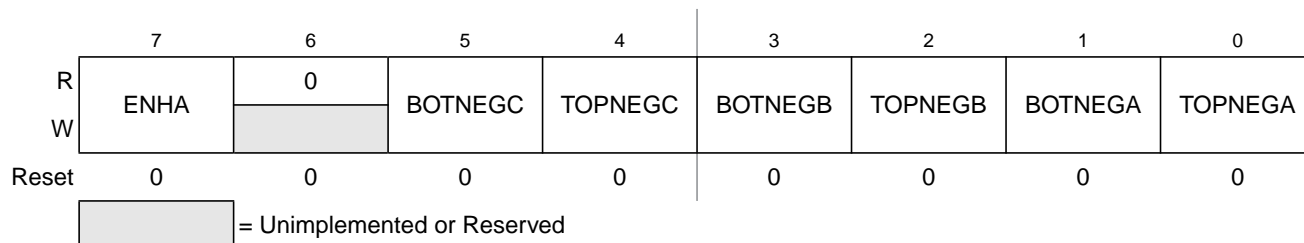
All PWM outputs can be generated from the same counter, or each pair can have its own counter for three independent PWM frequencies. Complementary operation permits programmable dead-time insertion, distortion correction through current sensing by software, and separate top and bottom output polarity control. Each counter value is programmable to support a continuously variable PWM frequency. Both edge- and center-aligned synchronous pulse width-control and full range modulation from 0 percent to 100 percent, are supported. The PMF is capable of controlling most motor types: AC induction motors (ACIM), both brushless (BLDC) and brush DC motors (BDC), switched (SRM), and variable reluctance motors (VRM), and stepper motors.

#### 11.1.1 Features

- Three complementary PWM signal pairs, or six independent PWM signals
- Three 15-bit counters
- Features of complementary channel operation
  - Deadtime insertion
  - Separate top and bottom pulse width correction via current status inputs or software
  - Separate top and bottom polarity control
- Edge-aligned or center-aligned PWM signals
- Half-cycle reload capability
- Integral reload rates from 1 to 16
- Individual software-controlled PWM output
- Programmable fault protection
- Polarity control

### 11.3.2.2 PMF Configure 1 Register (PMFCFG1)

Module Base + 0x0001



**Figure 11-5. PMF Configure 1 Register (PMFCFG1)**

Read anytime. This register cannot be modified after the WP bit is set.

A normal PWM output or positive polarity means that the PWM channel outputs high when the counter value is smaller than or equal to the pulse width value and outputs low otherwise. An inverted output or negative polarity means that the PWM channel outputs low when the counter value is smaller than or equal to the pulse width value and outputs high otherwise.

**Table 11-3. PMFCFG1 Field Descriptions**

Field	Description
7 ENHA	<b>Enable Hardware Acceleration</b> — This bit enables writing to the VLMODE[1:0], SWAPC, SWAPB, and SWAPA bits in the PMFCFG3 register. This bit cannot be modified after the WP bit is set. 0 Disable writing to VLMODE[1:0], SWAPC, SWAPB, and SWAPA bits 1 Enable writing to VLMODE[1:0], SWAPC, SWAPB, and SWAPA bits
5 BOTNEGC	<b>Pair C Bottom-side PWM Polarity</b> — This bit determines the polarity for Pair C bottom-side PWM (PWM5). This bit cannot be modified after the WP bit is set. 0 Positive PWM5 polarity 1 Negative PWM5 polarity
4 TOPNEGC	<b>Pair C Top-side PWM Polarity</b> — This bit determines the polarity for Pair C top-side PWM (PWM4). This bit cannot be modified after the WP bit is set. 0 Positive PWM4 polarity 1 Negative PWM4 polarity
3 BOTNEGB	<b>Pair B Bottom-side PWM Polarity</b> — This bit determines the polarity for Pair B bottom-side PWM (PWM3). This bit cannot be modified after the WP bit is set. 0 Positive PWM3 polarity 1 Negative PWM3 polarity
2 TOPNEGB	<b>Pair B Top-side PWM Polarity</b> — This bit determines the polarity for Pair B top-side PWM (PWM2). This bit cannot be modified after the WP bit is set. 0 Positive PWM2 polarity 1 Negative PWM2 polarity
1 BOTNEGA	<b>Pair A Bottom-side PWM Polarity</b> — This bit determines the polarity for Pair A bottom-side PWM (PWM1). This bit cannot be modified after the WP bit is set. 0 Positive PWM1 polarity 1 Negative PWM1 polarity
0 TOPNEGA	<b>Pair A Top-side PWM Polarity</b> — This bit determines the polarity for Pair A top-side PWM (PWM0). This bit cannot be modified after the WP bit is set. 0 Positive PWM0 polarity 1 Negative PWM0 polarity



### 11.3.2.5 PMF Fault Control Register (PMFFCTL)

Module Base + 0x0004

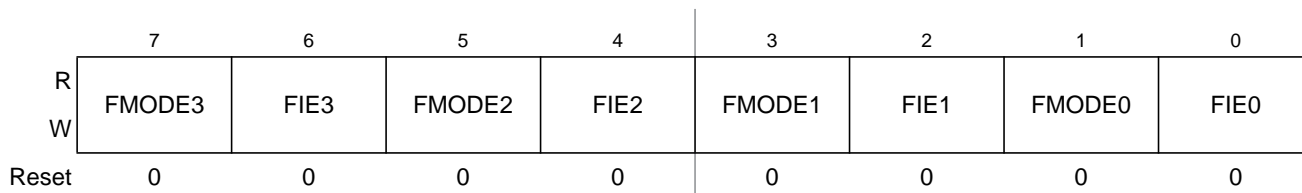


Figure 11-8. PMF Fault Control Register (PMFFCTL)

Read and write anytime.

Table 11-6. PMFFCTL Field Descriptions

Field	Description
7, 5, 3, 1 FMODE[3:0]	<b>Fault x Pin Clearing Mode</b> — This bit selects automatic or manual clearing of FAULTx pin faults. See Section 11.4.8.2, “Automatic Fault Clearing” and Section 11.4.8.3, “Manual Fault Clearing” for more details. 0 Manual fault clearing of FAULTx pin faults. 1 Automatic fault clearing of FAULTx pin faults. where x is 0, 1, 2, and 3.
6, 4, 2, 0 FIE[3:0]	<b>Fault x Pin Interrupt Enable</b> — This bit enables CPU interrupt requests to be generated by the FAULTx pin. The fault protection circuit is independent of the FIE <sub>x</sub> bit and is active when FPIN <sub>x</sub> is set. If a fault is detected, the PWM pins are disabled according to the PMF Disable Mapping registers. 0 Fault x CPU interrupt requests disabled. 1 Fault x CPU interrupt requests enabled. where x is 0, 1, 2 and 3.

### 11.3.2.6 PMF Fault Pin Enable Register (PMFFPIN)

Module Base + 0x0005

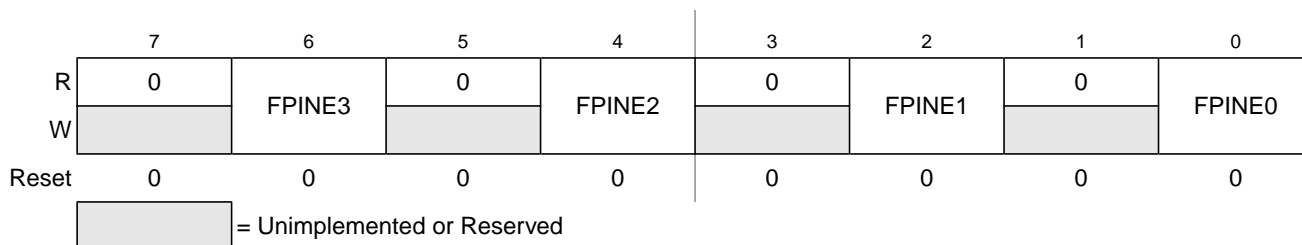


Figure 11-9. PMF Fault Pin Enable Register (PMFFPIN)

Read anytime. This register cannot be modified after the WP bit is set.

Table 11-7. PMFFPIN Field Descriptions

Field	Description
6, 4, 2, 0 FPINE[2:0]	<b>Fault x Pin Enable</b> — Where x is 0, 1, 2 and 3. 0 FAULTx pin is disabled for fault protection. 1 FAULTx pin is enabled for fault protection.

**NOTE**

16-bit misaligned reads and writes are not allowed. If attempted, the BDM will ignore the least significant bit of the address and will assume an even address from the remaining bits.

For hardware data read commands, the external host must wait 150 bus clock cycles after sending the address before attempting to obtain the read data. This is to be certain that valid data is available in the BDM shift register, ready to be shifted out. For hardware write commands, the external host must wait 150 bus clock cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed. The 150 bus clock cycle delay in both cases includes the maximum 128 cycle delay that can be incurred as the BDM waits for a free cycle before stealing a cycle.

For firmware read commands, the external host should wait 44 bus clock cycles after sending the command opcode and before attempting to obtain the read data. This includes the potential of an extra 7 cycles when the access is external with a narrow bus access (+1 cycle) and / or a stretch (+1, 2, or 3 cycles), (7 cycles could be needed if both occur). The 44 cycle wait allows enough time for the requested data to be made available in the BDM shift register, ready to be shifted out.

**NOTE**

This timing has increased from previous BDM modules due to the new capability in which the BDM serial interface can potentially run faster than the bus. On previous BDM modules this extra time could be hidden within the serial time.

For firmware write commands, the external host must wait 32 bus clock cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed.

The external host should wait 64 bus clock cycles after a TRACE1 or GO command before starting any new serial command. This is to allow the CPU to exit gracefully from the standard BDM firmware lookup table and resume execution of the user code. Disturbing the BDM shift register prematurely may adversely affect the exit from the standard BDM firmware lookup table.

**NOTE**

If the bus rate of the target processor is unknown or could be changing, it is recommended that the ACK (acknowledge function) be used to indicate when an operation is complete. When using ACK, the delay times are automated.

Figure 15-6 represents the BDM command structure. The command blocks illustrate a series of eight bit times starting with a falling edge. The bar across the top of the blocks indicates that the BKGD line idles in the high state. The time for an 8-bit command is  $8 \times 16$  target clock cycles.<sup>1</sup>

1. Target clock cycles are cycles measured using the target MCU's serial clock rate. See Section 15.4.6, "BDM Serial Interface," and Section 15.3.2.1, "BDM Status Register (BDMSTS)," for information on how serial clock rate is selected.

If an interrupt is pending when a TRACE1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. Upon return to standard BDM firmware execution, the program counter points to the first instruction in the interrupt service routine.

### 15.4.11 Instruction Tagging

The instruction queue and cycle-by-cycle CPU activity are reconstructible in real time or from trace history that is captured by a logic analyzer. However, the reconstructed queue cannot be used to stop the CPU at a specific instruction. This is because execution already has begun by the time an operation is visible outside the system. A separate instruction tagging mechanism is provided for this purpose.

The tag follows program information as it advances through the instruction queue. When a tagged instruction reaches the head of the queue, the CPU enters active BDM rather than executing the instruction.

#### NOTE

Tagging is disabled when BDM becomes active and BDM serial commands are not processed while tagging is active.

Executing the BDM TAGGO command configures two system pins for tagging. The  $\overline{\text{TAGLO}}$  signal shares a pin with the  $\overline{\text{LSTRB}}$  signal, and the  $\overline{\text{TAGHI}}$  signal shares a pin with the BKGD signal.

Table 15-7 shows the functions of the two tagging pins. The pins operate independently, that is the state of one pin does not affect the function of the other. The presence of logic level 0 on either pin at the fall of the external clock (ECLK) performs the indicated function. High tagging is allowed in all modes. Low tagging is allowed only when low strobe is enabled (LSTRB is allowed only in wide expanded modes and emulation expanded narrow mode).

Table 15-7. Tag Pin Function

$\overline{\text{TAGHI}}$	$\overline{\text{TAGLO}}$	Tag
1	1	No tag
1	0	Low byte
0	1	High byte
0	0	Both bytes

### 15.4.12 Serial Communication Time-Out

The host initiates a host-to-target serial transmission by generating a falling edge on the BKGD pin. If BKGD is kept low for more than 128 target clock cycles, the target understands that a SYNC command was issued. In this case, the target will keep waiting for a rising edge on BKGD in order to answer the SYNC request pulse. If the rising edge is not detected, the target will keep waiting forever without any time-out limit.

Consider now the case where the host returns BKGD to logic one before 128 cycles. This is interpreted as a valid bit transmission, and not as a SYNC request. The target will keep waiting for another falling edge marking the start of a new bit. If, however, a new falling edge is not detected by the target within 512 clock cycles since the last falling edge, a time-out occurs and the current command is discarded without affecting memory or the operating mode of the MCU. This is referred to as a soft-reset.

## 17.7 Exception Priority

The priority (from highest to lowest) and address of all exception vectors issued by the INT upon request by the CPU is shown in [Table 17-5](#).

**Table 17-5. Exception Vector Map and Priority**

Vector Address	Source
0xFFFE–0xFFFF	System reset
0xFFFC–0xFFFD	Crystal monitor reset
0xFFFA–0xFFFB	COP reset
0xFFF8–0xFFF9	Unimplemented opcode trap
0xFFF6–0xFFF7	Software interrupt instruction (SWI) or BDM vector request
0xFFF4–0xFFF5	$\overline{XIRQ}$ signal
0xFFF2–0xFFF3	$\overline{IRQ}$ signal
0xFFF0–0xFF00	Device-specific I-bit maskable interrupt sources (priority in descending order)

### 18.3.2.9 Mode Register (MODE)

	7	6	5	4	3	2	1	0
R				0		0		
W	MODC	MODB	MODA		IVIS		EMK	EME
Reset								
Special Single Chip	0	0	0	0	0	0	0	0
Emulation Expanded Narrow	0	0	1	0	1	0	1	1
Special Test	0	1	0	0	1	0	0	0
Emulation Expanded Wide	0	1	1	0	1	0	1	1
Normal Single Chip	1	0	0	0	0	0	0	0
Normal Expanded Narrow	1	0	1	0	0	0	0	0
Peripheral	1	1	0	0	0	0	0	0
Normal Expanded Wide	1	1	1	0	0	0	0	0

= Unimplemented or Reserved

**Figure 18-13. Mode Register (MODE)**

Read: Anytime (provided this register is in the map).

Write: Each bit has specific write conditions. Please refer to the descriptions of each bit on the following pages.

The MODE register is used to establish the operating mode and other miscellaneous functions (i.e., internal visibility and emulation of port E and K).

In special peripheral mode, this register is not accessible but it is reset as shown to system configuration features. Changes to bits in the MODE register are delayed one cycle after the write.

This register is not in the on-chip memory map in expanded and special peripheral modes. Therefore, these accesses will be echoed externally.