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	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, I²C, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	91
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K 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 ~ 105°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=mc9s12e256vpve

- Operating frequency
  - 50MHz equivalent to 25MHz Bus Speed
- Internal 2.5V Regulator
  - Input voltage range from 2.97V to 5.5V
  - Low power mode capability
  - Includes low voltage reset (LVR) circuitry
  - Includes low voltage interrupt (LVI) circuitry
- 112-Pin LQFP or 80-Pin QFP package
  - Up to 90 I/O lines with 5V input and drive capability (112 pin package)
  - Up to two dedicated 5V input only lines (IRQ and XIRQ)
  - Sixteen 3.3V/5V A/D converter inputs
- Development Support.
  - Single-wire background debug™ mode
  - On-chip hardware breakpoints
  - Enhanced debug features

1.1.2 Modes of Operation

User modes **(Expanded modes are only available in the 112-pin package version)**

- Normal modes
  - Normal Single-Chip Mode
  - Normal Expanded Wide Mode
  - Normal Expanded Narrow Mode
  - Emulation Expanded Wide Mode
  - Emulation Expanded Narrow Mode
- Special Operating Modes
  - Special Single-Chip Mode with active Background Debug Mode
  - Special Test Mode **(Freescale use only)**
  - Special Peripheral Mode **(Freescale use only)**
- Low power modes
  - Stop Mode
  - Pseudo Stop Mode
  - Wait Mode

2.3.1 Module Memory Map

The Flash memory map is shown in Figure 2-2. The HCS12 architecture places the Flash memory addresses between 0x4000 and 0xFFFF which corresponds to three 16-Kbyte pages. The content of the HCS12 core PPAGE register is used to map the logical middle page ranging from address 0x8000 to 0xBFFF to any physical 16 Kbyte page in the Flash memory. By placing 0x3E or 0x3F in the HCS12 Core PPAGE register, the associated 16 Kbyte pages appear twice in the MCU memory map.

The FPROT register, described in Section 2.3.2.5, “Flash Protection Register (FPROT)”, can be set to globally protect a Flash block. However, three separate memory regions, one growing upward from the first address in the next-to-last page in the Flash block (called the lower region), one growing downward from the last address in the last page in the Flash block (called the higher region), and the remaining addresses in the Flash block, can be activated for protection. The Flash locations of these protectable regions are shown in Table 2-3. The higher address region of Flash block 0 is mainly targeted to hold the boot loader code because it covers the vector space. The lower address region of any Flash block can be used for EEPROM emulation in an MCU without an EEPROM module because it can remain unprotected while the remaining addresses are protected from program or erase.

Security information that allows the MCU to restrict access to the Flash module is stored in the Flash configuration field found in Flash block 0, described in Table 2-2.

Table 2-2. Flash Configuration Field

Unpaged Flash Address	Paged Flash Address (PPAGE 0x3F)	Size (Bytes)	Description
0xFF00 – 0xFF07	0xBF00 – 0xBF07	8	Backdoor Comparison Key Refer to Section 2.6.1, “Unsecuring the MCU using Backdoor Key Access”
0xFF08 – 0xFF0B	0xBF08 – 0xBF0B	4	Reserved
0xFF0C	0xBF0C	1	Block 1 Flash Protection Byte Refer to Section 2.3.2.7, “Flash Status Register (FSTAT)”
0xFF0D	0xBF0D	1	Block 0 Flash Protection Byte Refer toSection 2.3.2.7, “Flash Status Register (FSTAT)”
0xFF0E	0xBF0E	1	Flash Nonvolatile Byte Refer to Section 2.3.2.9, “Flash Control Register (FCTL)”
0xFF0F	0xBF0F	1	Flash Security Byte Refer to Section 2.3.2.2, “Flash Security Register (FSEC)”

The Flash module also contains a set of 16 control and status registers located in address space module base + 0x0000 to module base + 0x000F. In order to accommodate more than one Flash block with a minimum register address space, a set of registers located from module base + 0x0004 to module base + 0x000B are repeated in all banks. The active register bank is selected by the BKSEL bits in the unbanked Flash configuration register (FCNFG). A summary of these registers is given in Table 2-4 while their accessibility in normal and special modes is detailed in Section 2.3.2, “Register Descriptions”.

Table 2-4. Flash Register Map

Module Base +	Register Name	Normal Mode Access
0x0000	Flash Clock Divider Register (FCLKDIV)	R/W
0x0001	Flash Security Register (FSEC)	R
0x0002	Flash Test Mode Register (FTSTMOD)[1]	R
0x0003	Flash Configuration Register (FCNFG)	R/W
0x0004	Flash Protection Register (FPROT)	R/W
0x0005	Flash Status Register (FSTAT)	R/W
0x0006	Flash Command Register (FCMD)	R/W
0x0007	Flash Control Register (FCTL)	R
0x0008	Flash High Address Register (FADDRHI)[1]	R
0x0009	Flash Low Address Register (FADDRLO)[1]	R
0x000A	Flash High Data Register (FDATAHI)	R
0x000B	Flash Low Data Register (FDATALO)	R
0x000C	RESERVED1[1]	R
0x000D	RESERVED2[1]	R
0x000E	RESERVED3[1]	R
0x000F	RESERVED4[1]	R

1 Intended for factory test purposes only.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000B ARMCOP	R	0	0	0	0	0	0	0	0
	W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

= Unimplemented or Reserved

Figure 4-3. CRG Register Summary (continued)

4.3.2.1 CRG Synthesizer Register (SYNR)

The SYNR register controls the multiplication factor of the PLL. If the PLL is on, the count in the loop divider (SYNR) register effectively multiplies up the PLL clock (PLLCLK) from the reference frequency by 2 x (SYNR+1). PLLCLK will not be below the minimum VCO frequency (f_{SCM}).

$$PLLCLK = 2 \times OSCCLK \times \frac{(SYNR + 1)}{(REFDV + 1)}$$

NOTE

If PLL is selected (PLLSEL=1), Bus Clock = PLLCLK / 2
Bus Clock must not exceed the maximum operating system frequency.

Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	0	0	SYN5	SYNR	SYN3	SYN2	SYN1	SYN0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 4-4. CRG Synthesizer Register (SYNR)

Read: anytime

Write: anytime except if PLLSEL = 1

NOTE

Write to this register initializes the lock detector bit and the track detector bit.

The PLL is a frequency generator that operates in either acquisition mode or tracking mode, depending on the difference between the output frequency and the target frequency. The PLL can change between acquisition and tracking modes either automatically or manually.

The VCO has a minimum operating frequency, which corresponds to the self-clock mode frequency f_{SCM}.

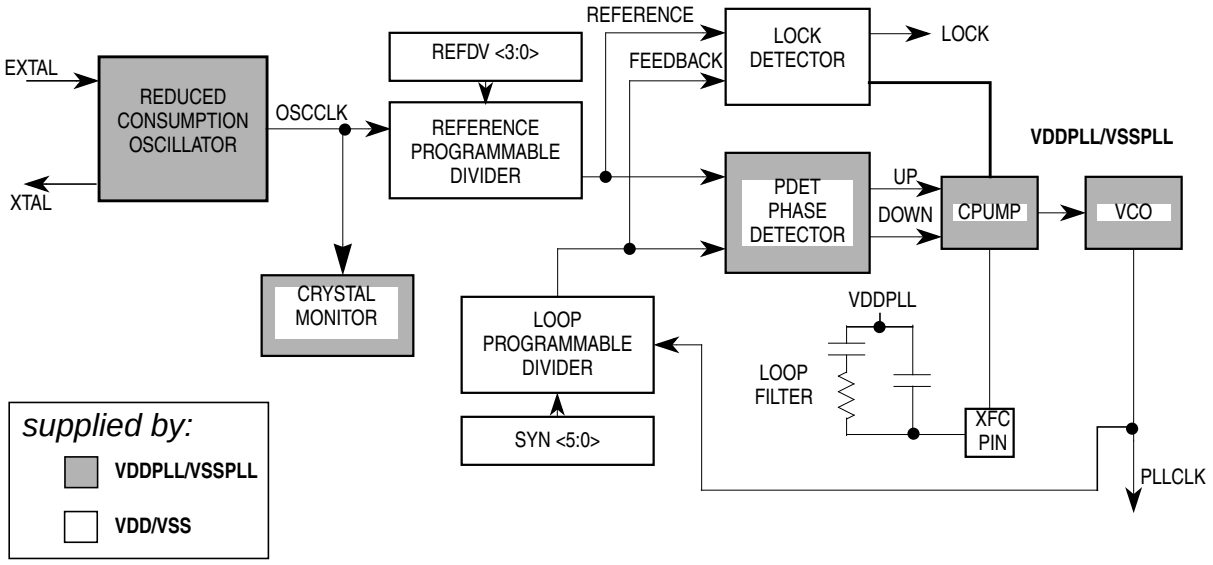


Figure 4-16. PLL Functional Diagram

4.4.1.1 PLL Operation

The oscillator output clock signal (OSCCLK) is fed through the reference programmable divider and is divided in a range of 1 to 16 (REFDV+1) to output the reference clock. The VCO output clock, (PLLCLK) is fed back through the programmable loop divider and is divided in a range of 2 to 128 in increments of [2 x (SYNR +1)] to output the feedback clock. See Figure 4-16.

The phase detector then compares the feedback clock, with the reference clock. Correction pulses are generated based on the phase difference between the two signals. The loop filter then slightly alters the DC voltage on the external filter capacitor connected to XFC pin, based on the width and direction of the correction pulse. The filter can make fast or slow corrections depending on its mode, as described in the next subsection. The values of the external filter network and the reference frequency determine the speed of the corrections and the stability of the PLL.

4.4.1.2 Acquisition and Tracking Modes

The lock detector compares the frequencies of the feedback clock, and the reference clock. Therefore, the speed of the lock detector is directly proportional to the final reference frequency. The circuit determines the mode of the PLL and the lock condition based on this comparison.

The internal reset of the MCU remains asserted while the reset generator completes the 192 SYSCLK long reset sequence. The reset generator circuitry always makes sure the internal reset is deasserted synchronously after completion of the 192 SYSCLK cycles. In case the RESET pin is externally driven low for more than these 192 SYSCLK cycles (external reset), the internal reset remains asserted too.

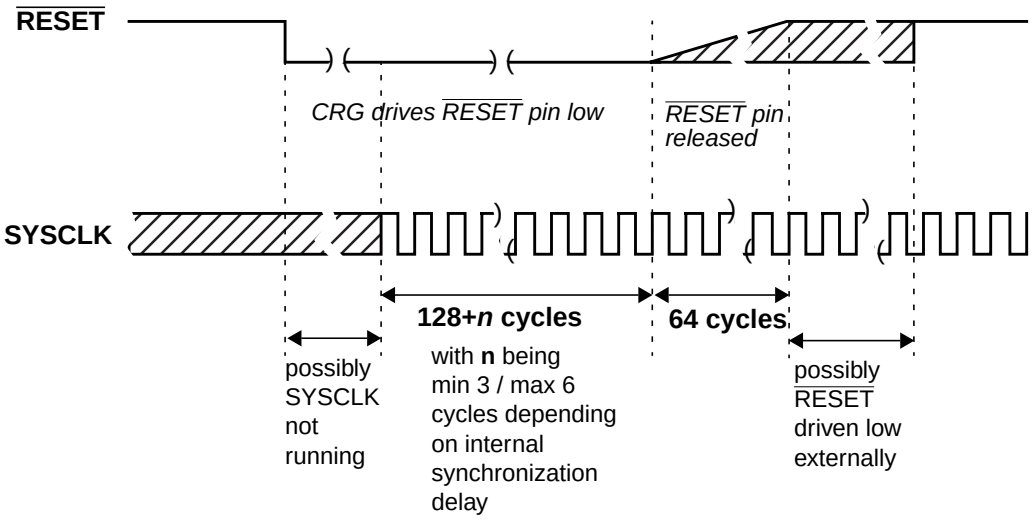


Figure 4-25. RESET Timing

4.5.1 Clock Monitor Reset

The CRGV4 generates a clock monitor reset in case all of the following conditions are true:

- Clock monitor is enabled (CME=1)
- Loss of clock is detected
- Self-clock mode is disabled (SCME=0)

The reset event asynchronously forces the configuration registers to their default settings (see Section 4.3, “Memory Map and Register Definition”). In detail the CME and the SCME are reset to logical ‘1’ (which doesn’t change the state of the CME bit, because it has already been set). As a consequence, the CRG immediately enters self-clock mode and starts its internal reset sequence. In parallel the clock quality check starts. As soon as clock quality check indicates a valid oscillator clock the CRG switches to OSCCLK and leaves self-clock mode. Because the clock quality checker is running in parallel to the reset generator, the CRG may leave self-clock mode while completing the internal reset sequence. When the reset sequence is finished the CRG checks the internally latched state of the clock monitor fail circuit. If a clock monitor fail is indicated processing begins by fetching the clock monitor reset vector.

4.5.2 Computer Operating Properly Watchdog (COP) Reset

When COP is enabled, the CRG expects sequential write of 0x0055 and 0x00AA (in this order) to the ARMCOP register during the selected time-out period. As soon as this is done, the COP time-out period restarts. If the program fails to do this the CRG will generate a reset. Also, if any value other than 0x0055 or 0x00AA is written, the CRG immediately generates a reset. In case windowed COP operation is enabled

6.2 External Signal Description

This section lists all inputs to the ATD10B16C block.

6.2.1 AN*x* (*x* = 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0) — Analog Input Channel *x* Pins

This pin serves as the analog input channel *x*. It can also be configured as general-purpose digital input and/or external trigger for the ATD conversion.

6.2.2 ETRIG3, ETRIG2, ETRIG1, ETRIG0 — External Trigger Pins

These inputs can be configured to serve as an external trigger for the ATD conversion.

Refer to the Device Overview chapter for availability and connectivity of these inputs.

6.2.3 V_{RH}, V_{RL} — High Reference Voltage Pin, Low Reference Voltage Pin

V_{RH} is the high reference voltage, V_{RL} is the low reference voltage for ATD conversion.

6.2.4 V_{DDA}, V_{SSA} — Analog Circuitry Power Supply Pins

These pins are the power supplies for the analog circuitry of the ATD10B16CV4 block.

6.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the ATD10B16C.

6.3.1 Module Memory Map

Table 6-1 gives an overview of all ATD10B16C registers

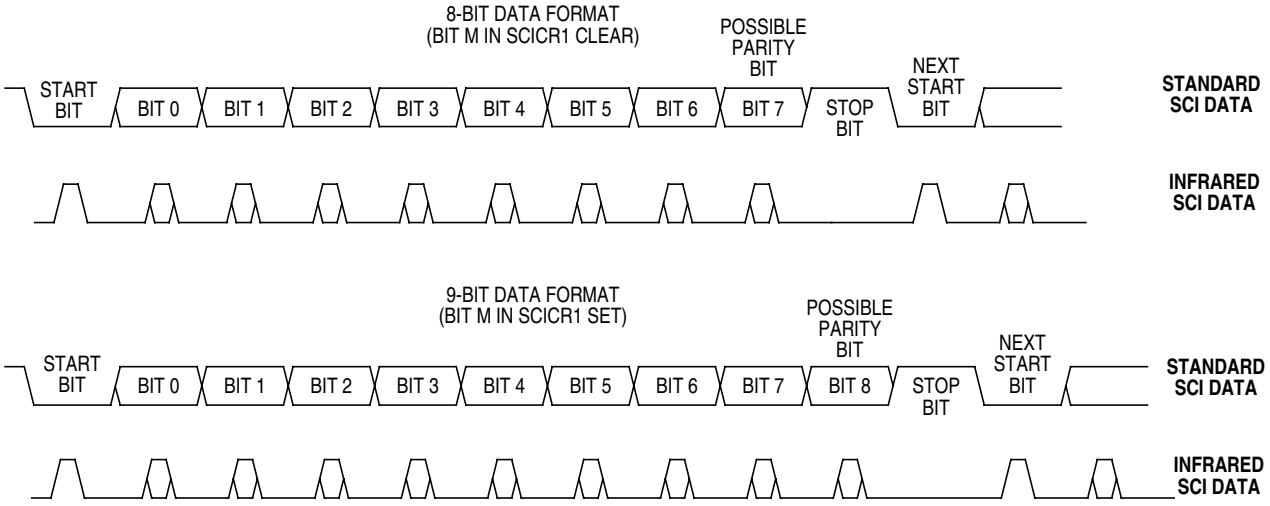


Figure 8-12. SCI Data Formats

Each data character is contained in a frame that includes a start bit, eight or nine data bits, and a stop bit. Clearing the M bit in SCI control register 1 configures the SCI for 8-bit data characters. A frame with eight data bits has a total of 10 bits. Setting the M bit configures the SCI for nine-bit data characters. A frame with nine data bits has a total of 11 bits

Table 8-12. Example of 8-bit Data Formats

Start Bit	Data Bits	Address Bits	Parity Bits	Stop Bit
1	8	0	0	1
1	7	0	1	1
1	7	1[1]	0	1

1 The address bit identifies the frame as an address character. See Section 8.5.5.6, “Receiver Wakeup”.

When the SCI is configured for 9-bit data characters, the ninth data bit is the T8 bit in SCI data register high (SCIDRH). It remains unchanged after transmission and can be used repeatedly without rewriting it. A frame with nine data bits has a total of 11 bits.

Table 8-13. Example of 9-Bit Data Formats

Start Bit	Data Bits	Address Bits	Parity Bits	Stop Bit
1	9	0	0	1
1	8	0	1	1
1	8	1[1]	0	1

1 The address bit identifies the frame as an address character. See Section 8.5.5.6, “Receiver Wakeup”.

The baud rate generator is activated only when the SPI is in the master mode and a serial transfer is taking place. In the other cases, the divider is disabled to decrease I_{DD} current.

$$BaudRateDivisor = (SPPR + 1) \bullet 2^{(SPR + 1)}$$

Figure 9-11. Baud Rate Divisor Equation

9.4.5 Special Features

9.4.5.1 $\overline{SS}$ Output

The $\overline{SS}$ output feature automatically drives the $\overline{SS}$ pin low during transmission to select external devices and drives it high during idle to deselect external devices. When $\overline{SS}$ output is selected, the $\overline{SS}$ output pin is connected to the $\overline{SS}$ input pin of the external device.

The $\overline{SS}$ output is available only in master mode during normal SPI operation by asserting SSOE and MODFEN bit as shown in Table 9-3.

The mode fault feature is disabled while $\overline{SS}$ output is enabled.

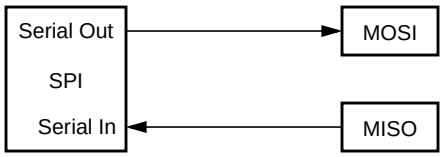
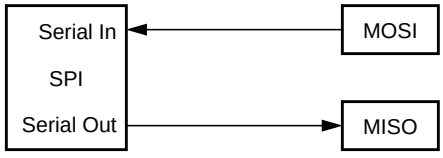
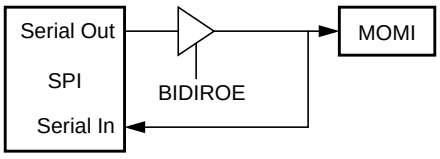
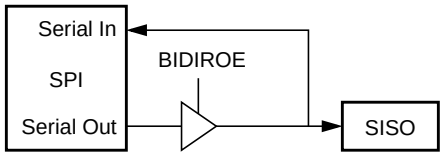
NOTE

Care must be taken when using the $\overline{SS}$ output feature in a multimaster system because the mode fault feature is not available for detecting system errors between masters.

9.4.5.2 Bidirectional Mode (MOSI or MISO)

The bidirectional mode is selected when the SPC0 bit is set in SPI Control Register 2 (see Table 9-9). In this mode, the SPI uses only one serial data pin for the interface with external device(s). The MSTR bit decides which pin to use. The MOSI pin becomes the serial data I/O (MOMI) pin for the master mode, and the MISO pin becomes serial data I/O (SISO) pin for the slave mode. The MISO pin in master mode and MOSI pin in slave mode are not used by the SPI.

Table 9-9. Normal Mode and Bidirectional Mode

When SPE = 1	Master Mode MSTR = 1	Slave Mode MSTR = 0
Normal Mode SPC0 = 0		
Bidirectional Mode SPC0 = 1		

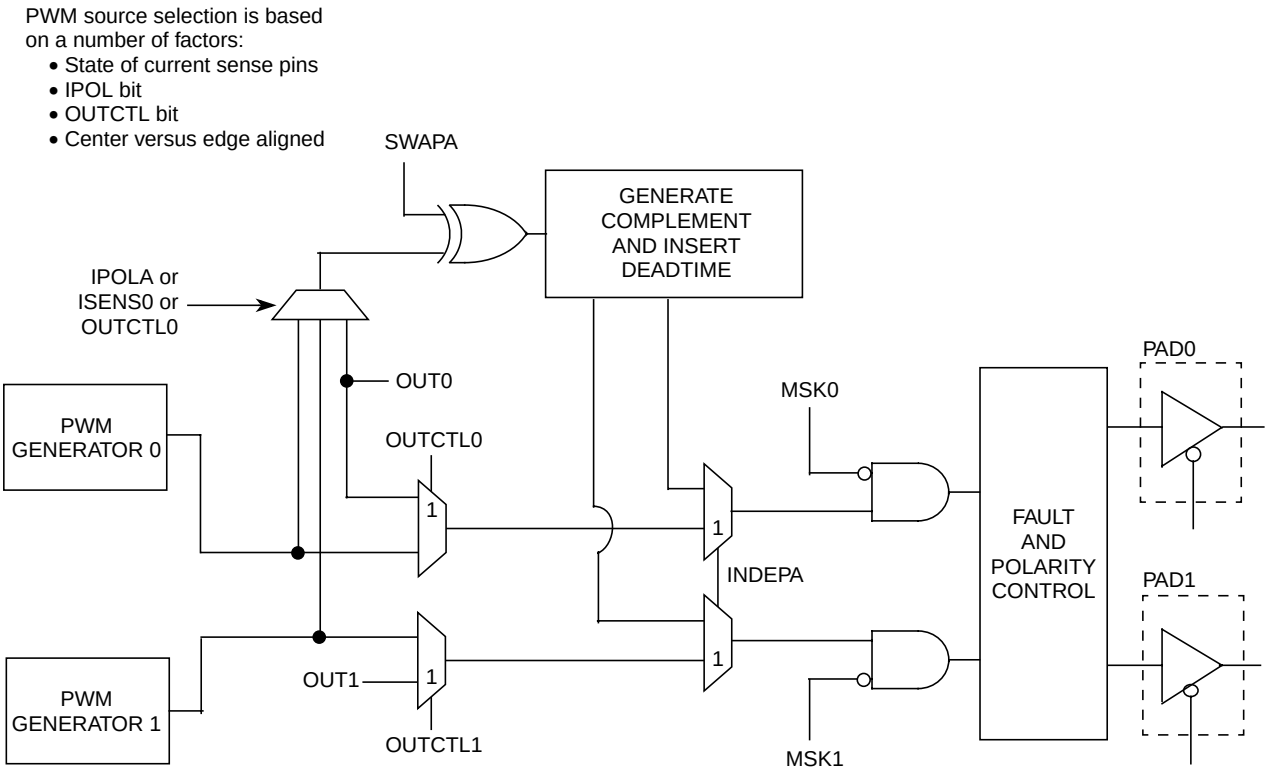


Figure 11-2. Detail of Mux, Swap, and Deadtime Functions

11.2 Signal Descriptions

The pulse width modulator has external pins named PWM0–5, FAULT0–3, and $\overline{IS0}$–$\overline{IS2}$.

11.2.1 PWM0–PWM5 Pins

PWM0–PWM5 are the output pins of the six PWM channels.

11.2.2 FAULT0–FAULT3 Pins

FAULT0–FAULT3 are input pins for disabling selected PWM outputs.

11.2.3 $\overline{IS0}$–$\overline{IS2}$ Pins

$\overline{IS0}$–$\overline{IS2}$ are current status pins for top/bottom pulse width correction in complementary channel operation while deadtime is asserted.

channel. In a complementary channel operation the even OUTCTL bit is used to enable software output control for the pair. But the OUTCTL bits must be switched in pairs for proper operation. The OUTCTLx and OUTx bits are in the PWM output control register.

NOTE

During software output control, TOPNEG and BOTNEG still control output polarity. It will take up to 3 clock cycles to see the effect of output control on the PWM output pins.

In independent PWM operation, setting or clearing the OUTx bit activates or deactivates the PWMx output.

In complementary channel operation, the even-numbered OUTx bits replace the PWM generator outputs as inputs to the deadtime generators. Complementary channel pairs still cannot be active simultaneously, and the deadtime generators continue to insert deadtime in both channels of that pair, whenever an even OUTx bit toggles. Even OUTx bits control the top PWM signals while the odd OUTx bits control the bottom PWM signals with respect to the even OUTx bits. Setting the odd OUTx bit makes its corresponding PWMx the complement of its even pair, while clearing the odd OUTx bit deactivates the odd PWMx.

Setting the OUTCTLx bits do not disable the PWM generators and current status sensing circuitry. They continue to run, but no longer control the output pins. When the OUTCTLx bits are cleared, the outputs of the PWM generator become the inputs to the deadtime generators at the beginning of the next PWM cycle. Software can drive the PWM outputs even when PWM enable bit (PWMEN) is set to zero.

NOTE

Avoid an unexpected deadtime insertion by clearing the OUTx bits before setting and after clearing the OUTCTLx bits.

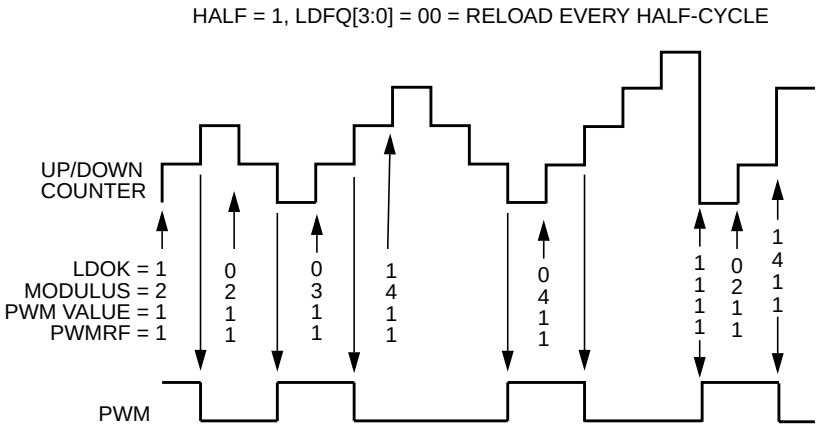


Figure 11-71. Half-Cycle Center-Aligned Modulus Loading

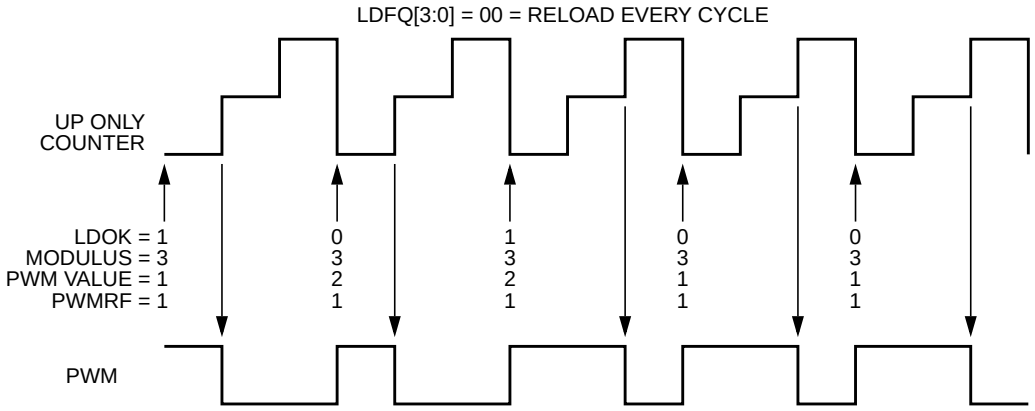


Figure 11-72. Edge-Aligned PWM Value Loading

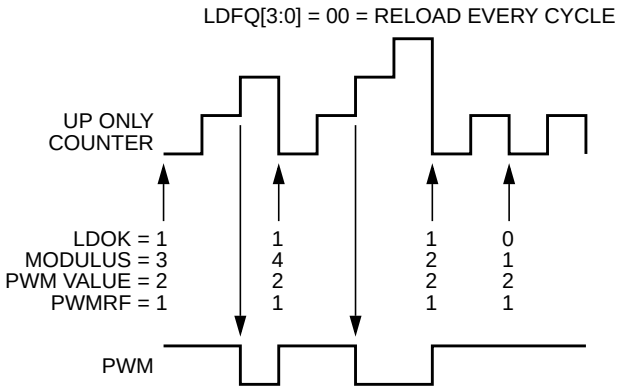


Figure 11-73. Untitled Figure

11.4.7.4 Initialization

Initialize all registers and set the LDOK bit before setting the PWMEN bit. With LDOK set, setting PWMEN for the first time after reset, immediately loads the PWM generator thereby setting the PWMRF flag. PWMRF generates a CPU interrupt request if the PWMRIE bit is set. In complementary channel

Table 12-5. PWMPRCLK Field Descriptions

Field	Description
6:5 PCKB[2:0]	**Prescaler Select for Clock B** — Clock B is 1 of two clock sources which can be used for channels 2 or 3. These three bits determine the rate of clock B, as shown in Table 12-6.
2:0 PCKA[2:0]	**Prescaler Select for Clock A** — Clock A is 1 of two clock sources which can be used for channels 0, 1, 4, or 5. These three bits determine the rate of clock A, as shown in Table 12-7.

Table 12-6. Clock B Prescaler Selects

PCKB2	PCKB1	PCKB0	Value of Clock B
0	0	0	Bus Clock
0	0	1	Bus Clock / 2
0	1	0	Bus Clock / 4
0	1	1	Bus Clock / 8
1	0	0	Bus Clock / 16
1	0	1	Bus Clock / 32
1	1	0	Bus Clock / 64
1	1	1	Bus Clock / 128

Table 12-7. Clock A Prescaler Selects

PCKA2	PCKA1	PCKA0	Value of Clock A
0	0	0	Bus Clock
0	0	1	Bus Clock / 2
0	1	0	Bus Clock / 4
0	1	1	Bus Clock / 8
1	0	0	Bus Clock / 16
1	0	1	Bus Clock / 32
1	1	0	Bus Clock / 64
1	1	1	Bus Clock / 128

12.3.2.5 PWM Center Align Enable Register (PWMCAE)

The PWMCAE register contains six control bits for the selection of center aligned outputs or left aligned outputs for each PWM channel. If the CAEx bit is set to a 1, the corresponding PWM output will be center aligned. If the CAEx bit is cleared, the corresponding PWM output will be left aligned. Reference Section 12.4.2.5, “Left Aligned Outputs,” and Section 12.4.2.6, “Center Aligned Outputs,” for a more detailed description of the PWM output modes.

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	0	CON45	CON23	CON01	PSWAI	PFRZ	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 12-8. PWM Control Register (PWMCTL)

Read: anytime

Write: anytime

There are three control bits for concatenation, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. When channels 4 and 5 are concatenated, channel 4 registers become the high-order bytes of the double-byte channel. When channels 2 and 3 are concatenated, channel 2 registers become the high-order bytes of the double-byte channel. When channels 0 and 1 are concatenated, channel 0 registers become the high-order bytes of the double-byte channel.

Reference Section 12.4.2.7, “PWM 16-Bit Functions,” for a more detailed description of the concatenation PWM function.

NOTE

Change these bits only when both corresponding channels are disabled.

14.1.3 Block Diagram

Figure 14-1 shows the function principle of VREG3V3V2 by means of a block diagram. The regulator core REG consists of two parallel sub-blocks, REG1 and REG2, providing two independent output voltages.

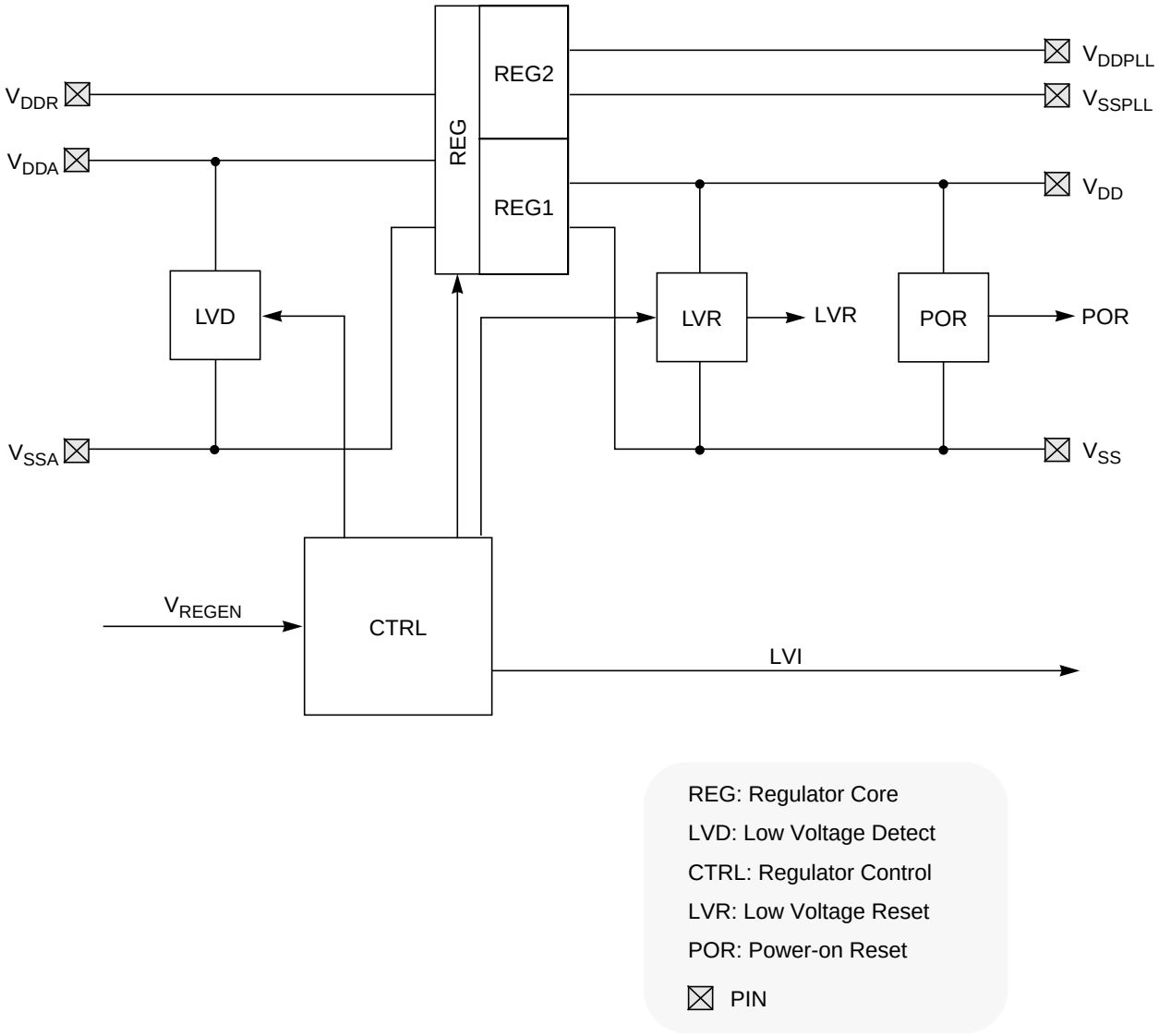


Figure 14-1. VREG3V3 Block Diagram

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).

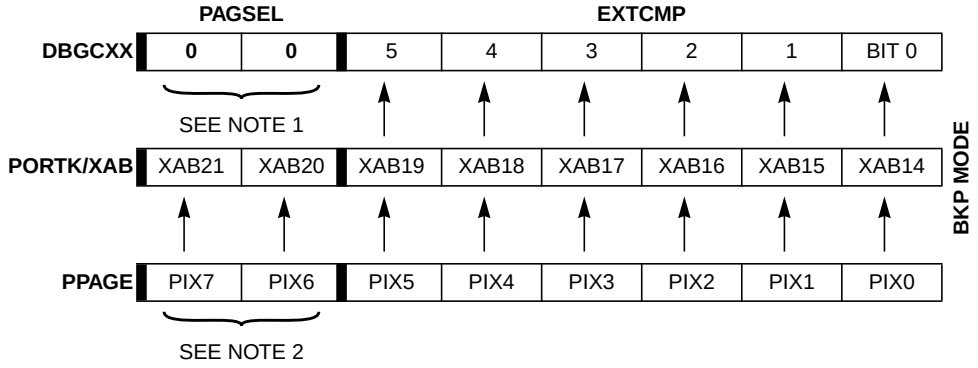
Table 15-7. Tag Pin Function

$\overline{\text{TAGHI}}$	$\overline{\text{TAGLO}}$	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.



NOTES:
1. In BKP mode, PAGSEL has no functionality. Therefore, set PAGSEL to 00 (reset state).
2. Current HCS12 implementations are limited to six PPAGE bits, PIX[5:0].

Figure 16-16. Comparators A and B Extended Comparison in BKP Mode

16.3.2.10 Debug Comparator A Register (DBGCA)

Module Base + 0x002B
Starting address location affected by INITRG register setting.

	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 16-17. Debug Comparator A Register High (DBGCAH)

Module Base + 0x002C
Starting address location affected by INITRG register setting.

	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 16-18. Debug Comparator A Register Low (DBGCAL)

Table 16-21. DBGCA Field Descriptions

Field	Description
15:0 15:0	**Comparator A Compare Bits** — The comparator A compare bits control whether comparator A compares the address bus bits [15:0] to a logic 1 or logic 0. See Table 16-20. 0 Compare corresponding address bit to a logic 0 1 Compare corresponding address bit to a logic 1

Table 17-3. ITEST Field Descriptions

Field	Description
7:0 INT[E:0]	**Interrupt TEST Bits** — These registers are used in special modes for testing the interrupt logic and priority independent of the system configuration. Each bit is used to force a specific interrupt vector by writing it to a logic 1 state. Bits are named INTE through INT0 to indicate vectors 0xFFxE through 0xFFx0. These bits can be written only in special modes and only with the WRTINT bit set (logic 1) in the interrupt test control register (ITCR). In addition, I interrupts must be masked using the I bit in the CCR. In this state, the interrupt input lines to the interrupt sub-block will be disconnected and interrupt requests will be generated only by this register. These bits can also be read in special modes to view that an interrupt requested by a system block (such as a peripheral block) has reached the INT module. There is a test register implemented for every eight interrupts in the overall system. All of the test registers share the same address and are individually selected using the value stored in the ADR[3:0] bits of the interrupt test control register (ITCR). **Note:** When ADR[3:0] have the value of 0x000F, only bits 2:0 in the ITEST register will be accessible. That is, vectors higher than 0xFFF4 cannot be tested using the test registers and bits 7:3 will always read as a logic 0. If ADR[3:0] point to an unimplemented test register, writes will have no effect and reads will always return a logic 0 value.

17.3.2.3 Highest Priority I Interrupt (Optional)

Module Base + 0x001F
Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	PSEL7	PSEL6	PSEL5	PSEL4	PSEL3	PSEL2	PSEL1	0
W								
Reset	1	1	1	1	0	0	1	0

= Unimplemented or Reserved

Figure 17-4. Highest Priority I Interrupt Register (HPRIO)

Read: Anytime

Write: Only if I mask in CCR = 1

Table 17-4. HPRIO Field Descriptions

Field	Description
7:1 PSEL[7:1]	**Highest Priority I Interrupt Select Bits** — The state of these bits determines which I-bit maskable interrupt will be promoted to highest priority (of the I-bit maskable interrupts). To promote an interrupt, the user writes the least significant byte of the associated interrupt vector address to this register. If an unimplemented vector address or a non I-bit masked vector address (value higher than 0x00F2) is written, IRQ (0xFFF2) will be the default highest priority interrupt.

17.4 Functional Description

The interrupt sub-block processes all exception requests made by the CPU. These exceptions include interrupt vector requests and reset vector requests. Each of these exception types and their overall priority level is discussed in the subsections below.

17.4.1 Low-Power Modes

The INT does not contain any user-controlled options for reducing power consumption. The operation of the INT in low-power modes is discussed in the following subsections.

17.4.1.1 Operation in Run Mode

The INT does not contain any options for reducing power in run mode.

17.4.1.2 Operation in Wait Mode

Clocks to the INT can be shut off during system wait mode and the asynchronous interrupt path will be used to generate the wake-up signal upon recognition of a valid interrupt or any $\overline{\text{XIRQ}}$ request.

17.4.1.3 Operation in Stop Mode

Clocks to the INT can be shut off during system stop mode and the asynchronous interrupt path will be used to generate the wake-up signal upon recognition of a valid interrupt or any $\overline{\text{XIRQ}}$ request.

17.5 Resets

The INT supports three system reset exception request types: normal system reset or power-on-reset request, crystal monitor reset request, and COP watchdog reset request. The type of reset exception request must be decoded by the system and the proper request made to the core. The INT will then provide the service routine address for the type of reset requested.

17.6 Interrupts

As shown in the block diagram in Figure 17-1, the INT contains a register block to provide interrupt status and control, an optional highest priority I interrupt (HPRIO) block, and a priority decoder to evaluate whether pending interrupts are valid and assess their priority.

17.6.1 Interrupt Registers

The INT registers are accessible only in special modes of operation and function as described in Section 17.3.2.1, “Interrupt Test Control Register,” and Section 17.3.2.2, “Interrupt Test Registers,” previously.

17.6.2 Highest Priority I-Bit Maskable Interrupt

When the optional HPRIO block is implemented, the user is allowed to promote a single I-bit maskable interrupt to be the highest priority I interrupt. The HPRIO evaluates all interrupt exception requests and passes the HPRIO vector to the priority decoder if the highest priority I interrupt is active. RTI replaces the promoted interrupt source.

A.4 Flash NVM

A.4.1 NVM Timing

The time base for all NVM program or erase operations is derived from the oscillator. A minimum oscillator frequency f_{NVMOSC} is required for performing program or erase operations. The NVM modules do not have any means to monitor the frequency and will not prevent program or erase operation at frequencies above or below the specified minimum. Attempting to program or erase the NVM modules at a lower frequency a full program or erase transition is not assured.

The Flash program and erase operations are timed using a clock derived from the oscillator using the FCLKDIV register. The frequency of this clock must be set within the limits specified as f_{NVMOP}.

The minimum program and erase times shown in Table A-14 are calculated for maximum f_{NVMOP} and maximum f_{bus}. The maximum times are calculated for minimum f_{NVMOP} and a f_{bus} of 2MHz.

A.4.1.1 Single Word Programming

The programming time for single word programming is dependent on the bus frequency as a well as on the frequency f_{NVMOP} and can be calculated according to the following formula.

$$t_{swpgm} = 9 \cdot \frac{1}{f_{NVMOP}} + 25 \cdot \frac{1}{f_{bus}}$$

A.4.1.2 Row Programming

Flash programming where up to 64 words in a row can be programmed consecutively by keeping the command pipeline filled. The time to program a consecutive word can be calculated as:

$$t_{bwpgm} = 4 \cdot \frac{1}{f_{NVMOP}} + 9 \cdot \frac{1}{f_{bus}}$$

The time to program a whole row is:

$$t_{brpgm} = t_{swpgm} + 63 \cdot t_{bwpgm}$$

Row programming is more than 2 times faster than single word programming.

A.4.1.3 Sector Erase

Erasing a 1024 byte Flash sector takes:

$$t_{era} \approx 4000 \cdot \frac{1}{f_{NVMOP}}$$

The setup times can be ignored for this operation.