Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

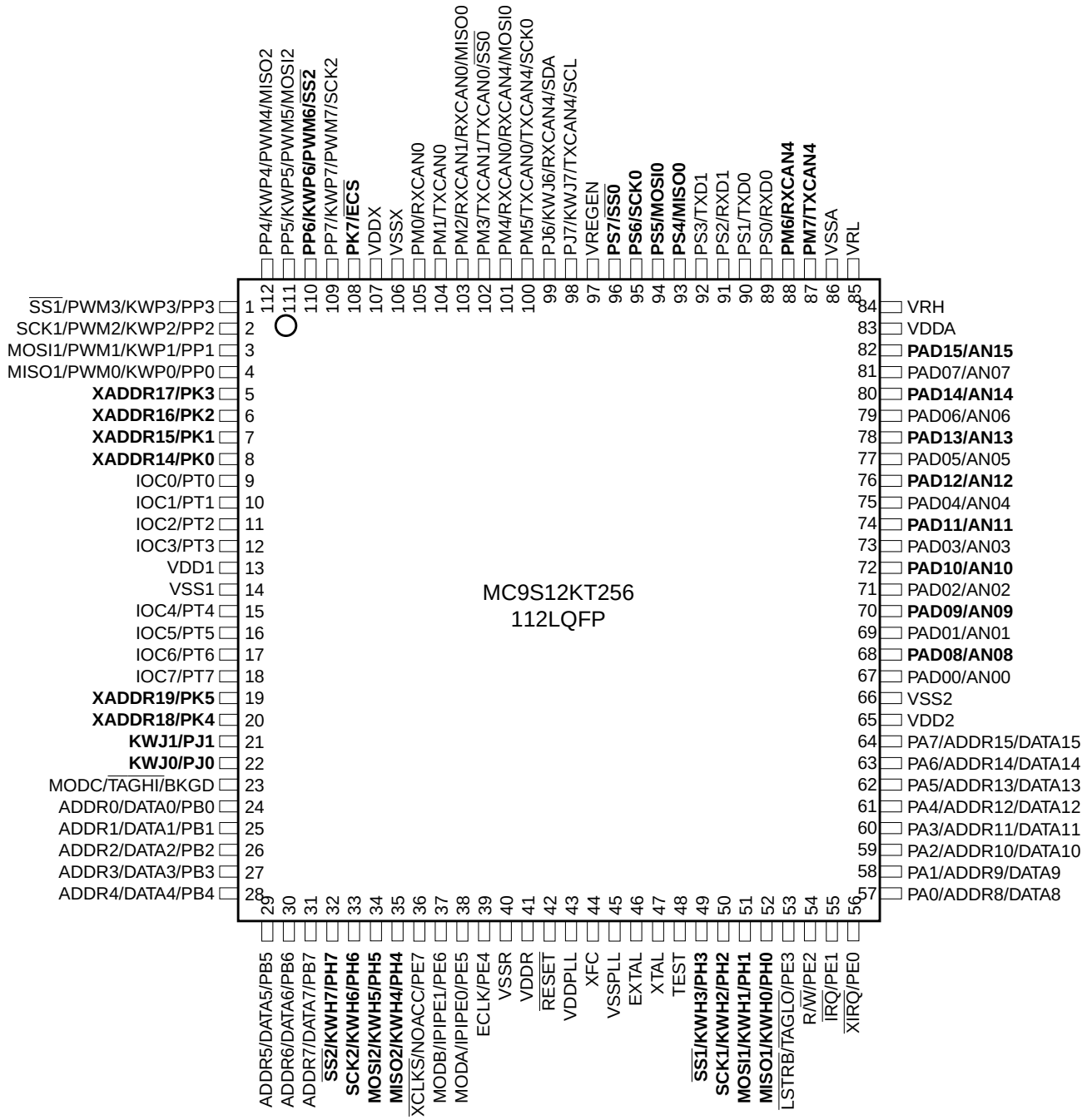
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, EBI/EMI, I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	91
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
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/product-detail/nxp-semiconductors/mc9s12kg256mpve

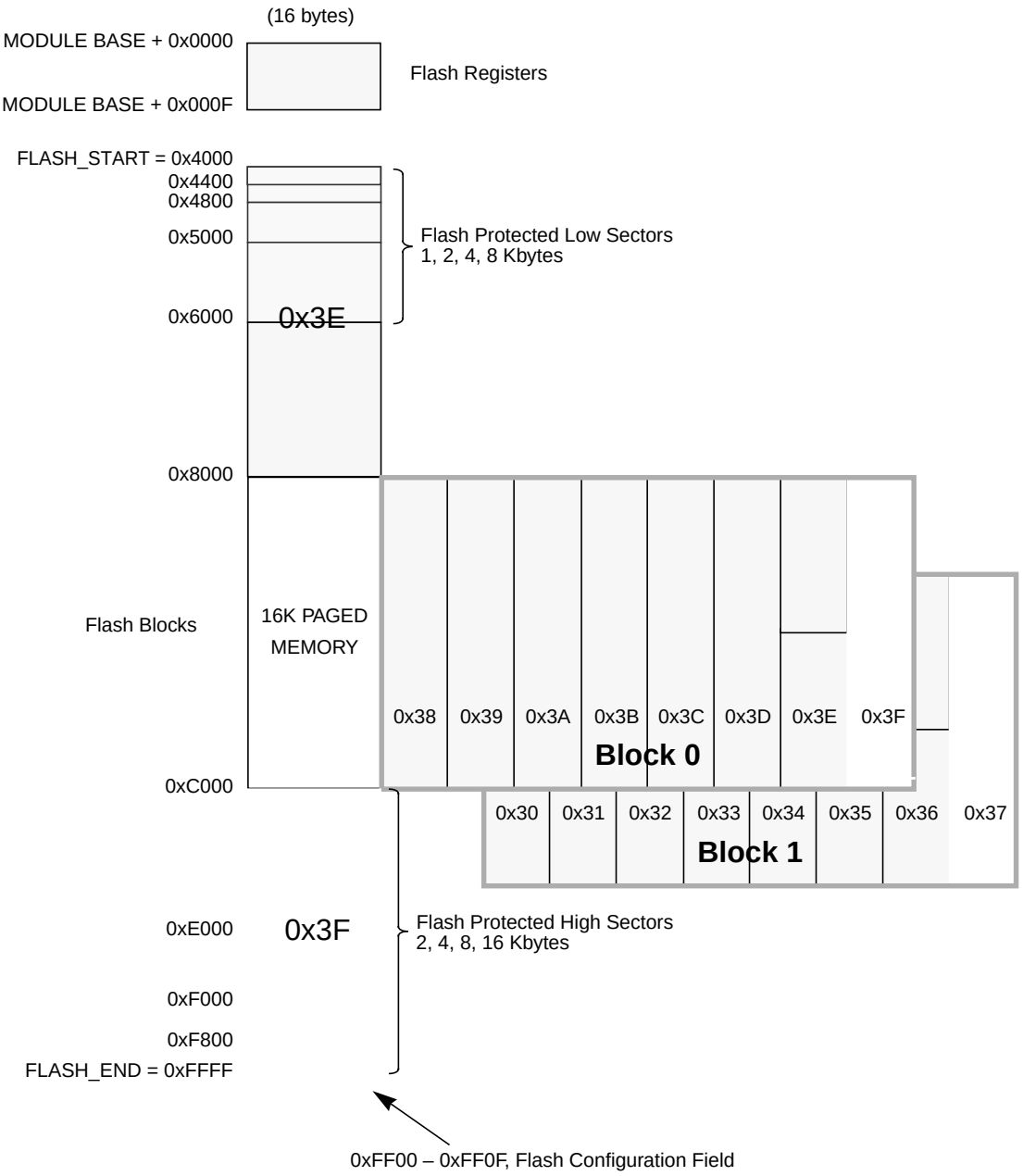
1.2 Signal Description

The MC9S12KT256 is available in a 112-pin low profile quad flat pack (LQFP) and a 80-pin quad flat pack (QFP). Most pins perform two or more functions, as described in Section 1.2.1, “Signal Properties Summary”. Figure 1-3 and Figure 1-4 show the pin assignments for different packages.



Signals shown in **Bold** are not available on the 80-pin package

Figure 1-3. Pin Assignments for 112 LQFP



Note: 0x30–0x3F correspond to the PPAGE register content

Figure 2-2. Flash Memory Map

3.1.1 Glossary

Banked Register — A memory-mapped register operating on one Flash block which shares the same register address as the equivalent registers for the other Flash blocks. The active register bank is selected by the BKSEL bit in the FCNFG register.

Command Write Sequence — A three-step MCU instruction sequence to execute built-in algorithms (including program and erase) on the Flash memory.

Common Register — A memory-mapped register which operates on all Flash blocks.

3.1.2 Features

- 256 Kbytes of Flash memory comprised of two 128 Kbyte blocks with each block divided into 128 sectors of 1024 bytes with every word (two bytes) accompanied by 6 ECC parity bits
- Single bit fault correction per word during read operations
- Automated program and erase algorithm with generation of ECC parity bits
- Interrupts on Flash command completion, command buffer empty and double bit fault detection
- Fast sector erase and word program operation
- 2-stage command pipeline for faster multi-word program times
- Sector erase abort feature for critical interrupt response
- Flexible protection scheme to prevent accidental program or erase
- Single power supply for all Flash operations including program and erase
- Security feature to prevent unauthorized access to the Flash memory
- Code integrity check using built-in data compression

3.1.3 Modes of Operation

Program, erase, erase verify, and data compress operations (please refer to Section 3.4.1 for details).

3.1.4 Block Diagram

A block diagram of the Flash module is shown in Figure 3-1.

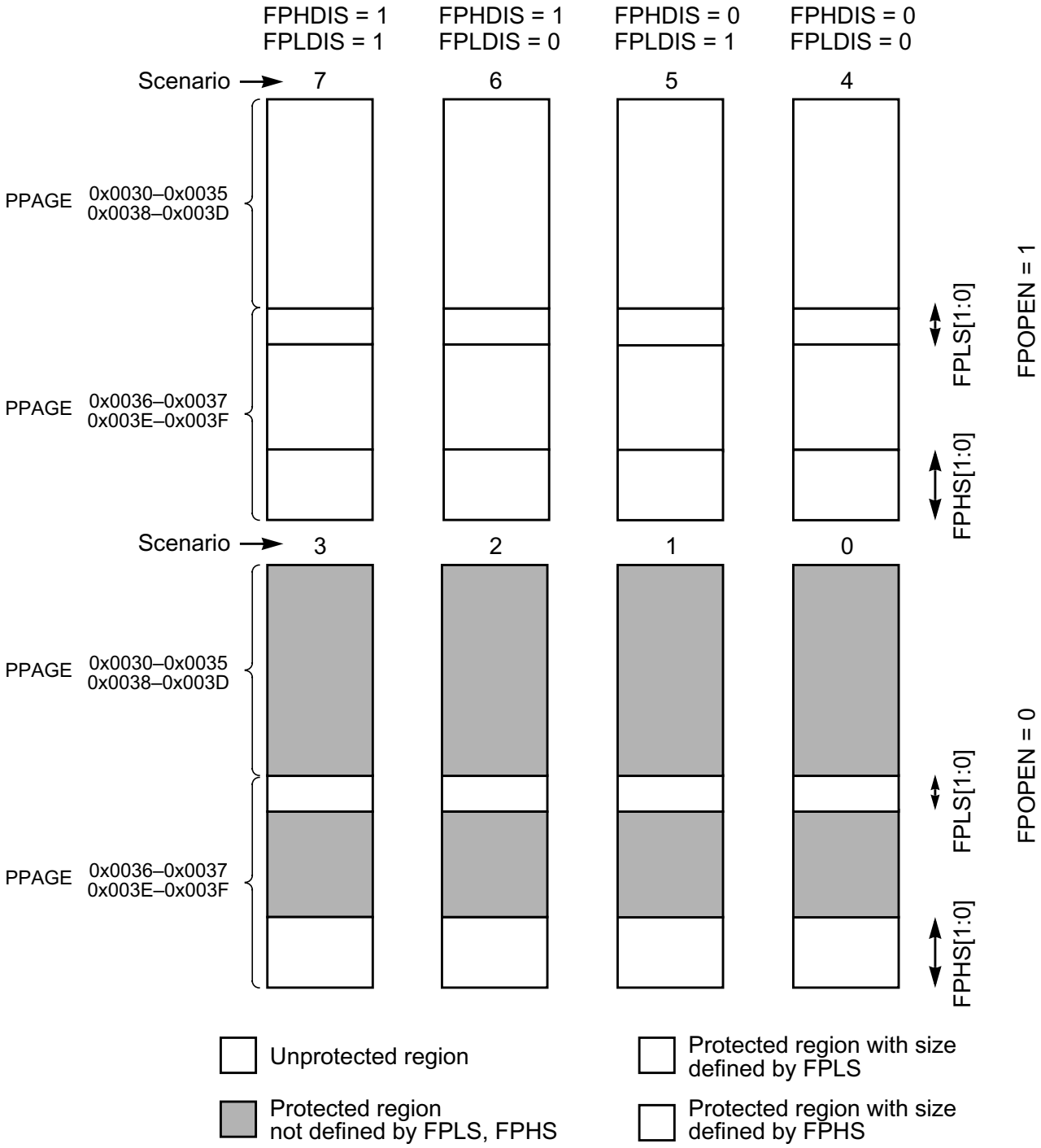


Figure 3-8. Flash Protection Scenarios

3.3.2.6 Flash Protection Restrictions

The general guideline is that Flash protection can only be added and not removed. Table 3-16 specifies all valid transitions between Flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register will be ignored and the FPROT register will remain unchanged. The contents of the

Data Compress Operation

The Flash module contains a 16-bit multiple-input signature register (MISR) to generate a 16-bit signature based on selected Flash array data. The final signature, which is stored in the associated banked FDATA register, is based on the following logic equation which is executed on every data compression cycle during the operation:

MISR[15:0] = {MISR[14:0], ^MISR[15,4,2,1]} ^ DATA[15:0] ***Eqn. 3-1***

where MISR is the content of the internal signature register associated with each Flash block and DATA is the data to be compressed as shown in Figure 3-26.

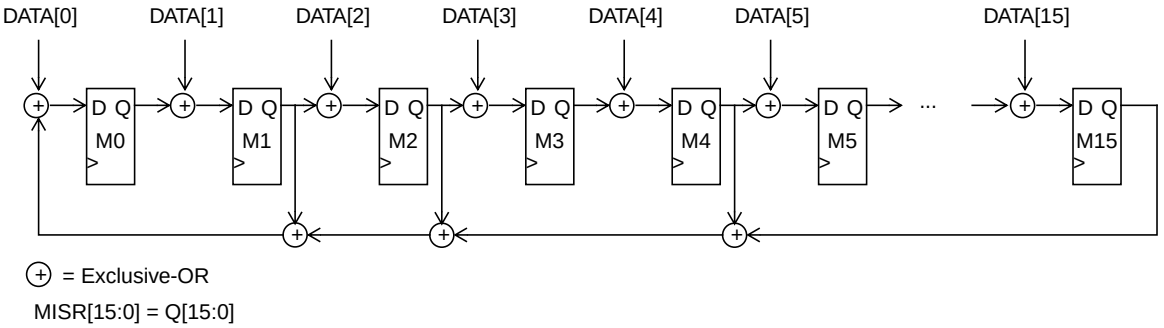


Figure 3-26. 16-Bit MISR Diagram

During the data compress operation, the following steps are executed:

1. MISR is reset to 0xFFFF.
2. DATA from the selected Flash array data range is read and compressed into the MISR with address incrementing.
3. DATA from the selected Flash array data range is read and compressed into the MISR with address decrementing.
4. The contents of the MISR are written to the associated banked FDATA register.

Chapter 4
4 Kbyte EEPROM Module (S12EETS4KV2)

4.1 Introduction

This document describes the EETS4K module which is a 4 Kbyte EEPROM (nonvolatile) memory. The EETS4K block uses a small sector Flash memory to emulate EEPROM functionality. It is an array of electrically erasable and programmable, nonvolatile memory. The EEPROM memory is organized as 2048 rows of 2 bytes (1 word). The EEPROM memory's erase sector size is 2 rows or 2 words (4 bytes).

The EEPROM memory may be read as either bytes, aligned words, or misaligned words. Read access time is one bus cycle for byte and aligned word, and two bus cycles for misaligned words.

Program and erase functions are controlled by a command driven interface. Both sector erase and mass erase of the entire EEPROM memory are supported. An erased bit reads 1 and a programmed bit reads 0. The high voltage required to program and erase is generated internally by on-chip charge pumps.

It is not possible to read from the EEPROM memory while it is being erased or programmed.

The EEPROM memory is ideal for data storage for single-supply applications allowing for field reprogramming without requiring external programming voltage sources.

> An EEPROM word must be in the erased state before being programmed. Cumulative programming of bits within a word is not allowed.

4.1.1 Glossary

Command Write Sequence — A three-step MCU instruction sequence to program, erase, or erase verify the EEPROM.

4.1.2 Features

- 4 Kbytes of EEPROM memory
- Minimum erase sector of 4 bytes
- Automated program and erase algorithms
- Interrupts on EEPROM command completion and command buffer empty
- Fast sector erase and word program operation
- 2-stage command pipeline
- Flexible protection scheme for protection against accidental program or erase
- Single power supply program and erase

5.2 External Signal Description

This section lists and describes the signals that do connect off chip.

5.2.1 Signal Properties

Table 5-1 shows all the pins and their functions that are controlled by the PIM9KT256. If there is more than one function associated with a pin, the priority is indicated by the position in the table from top (highest priority) to down (lowest priority). All pins have reset state as input.

Table 5-1. Pin Functions and Priorities (Sheet 1 of 4)

Port	Pin Name	Pin Function and Priority	Description	Pull Mode after Reset	Pin Function after Reset
Port T	PT[7:0]	IOC[7:0]	Timer Channels 7 to 0	Hi-Z	GPIO
		GPIO	General-purpose I/O		
Port S	PS7	$\overline{SS0}$	Serial Peripheral Interface 0 slave select output in master mode, input in slave mode or master mode.	Pull-up	GPIO
		GPIO	General-purpose I/O		
	PS6	SCK0	Serial Peripheral Interface 0 serial clock pin		
		GPIO	General-purpose I/O		
	PS5	MOSI0	Serial Peripheral Interface 0 master out/slave in pin		
		GPIO	General-purpose I/O		
	PS4	MISO0	Serial Peripheral Interface 0 master in/slave out pin		
		GPIO	General-purpose I/O		
	PS3	TXD1	Serial Communication Interface 1 transmit pin		
		GPIO	General-purpose I/O		
	PS2	RXD1	Serial Communication Interface 1 receive pin		
		GPIO	General-purpose I/O		
	PS1	TXD0	Serial Communication Interface 0 transmit pin		
		GPIO	General-purpose I/O		
	PS0	RXD0	Serial Communication Interface 0 receive pin		
		GPIO	General-purpose I/O		

5.3.2 Port S Registers

Port S is associated with the serial peripheral interface (SPI0) and serial communication interfaces (SCI1, SCI0). Each pin is assigned to these modules according to the following priority: SPI0/SCI1/SCI0 > general-purpose I/O.

When the SPI0 is enabled, the PS[7:4] pins become $\overline{SS0}$, SCK0, MOSI0, and MISO0 respectively. Refer to the SPI block description chapter for information on enabling and disabling the SPI0. The SPI0 pins can be re-routed. Refer to Section 5.3.3.8, “Module Routing Register (MODRR)”.

When the SCI1 receiver and transmitter are enabled, the PS[3:2] pins become TXD1 and RXD1 respectively. When the SCI0 receiver and transmitter are enabled, the PS[1:0] pins become TXD0 and RXD0 respectively. Refer to the SCI block description chapter for information on enabling and disabling the SCI receiver and transmitter.

During reset, port S pins are configured as inputs with pull-up.

5.3.2.1 Port S I/O Register (PTS)

Module Base + 0x0008

	7	6	5	4	3	2	1	0
R W	PTS7	PTS6	PTS5	PTS4	PTS3	PTS2	PTS1	PTS0
SPI/SCI	SS0	SCK0	MOSI0	MISO0	TXD1	RXD1	TXD0	RXD0
Reset	0	0	0	0	0	0	0	0

Figure 5-8. Port S I/O Register (PTS)

Read: Anytime. Write: Anytime.

If the data direction bits of the associated I/O pins are set to 1, a read returns the value of the port register, otherwise the value at the pins is read.

The SPI0 function takes precedence over the general-purpose I/O function if the SPI0 is enabled.

If enabled, the SCI0(1) transmitter takes precedence over the general-purpose I/O function, and the corresponding TXD0(1) pin is configured as an output. If enabled, the SCI0(1) receiver takes precedence over the general-purpose I/O function, and the corresponding RXD0(1) pin is configured as an input.

5.3.6.5 Port J Pull Device Enable Register (PERJ)

Module Base + 0x002C

	7	6	5	4	3	2	1	0
R	PERJ7	PERJ6	0	0	0	0	PERJ1	PERJ0
W								
Reset	1	1	—	—	—	—	1	1

= Unimplemented or Reserved

Figure 5-43. Port J Pull Device Enable Register (PERJ)

Read: Anytime. Write: Anytime.

This register configures whether a pull-up or a pull-down device is activated, if the port is used as input or as wired-OR output. This bit has no effect if the port is used as push-pull output. Out of reset a pull-up device is enabled.

Table 5-38. PERJ Field Descriptions

Field	Description
7, 6, 1, 0 PERJ[7:6] PERJ[1:0]	**Pull Device Enable Port J** 0 Pull-up or pull-down device is disabled. 1 Either a pull-up or pull-down device is enabled.

5.3.6.6 Port J Polarity Select Register (PPSJ)

Module Base + 0x002D

	7	6	5	4	3	2	1	0
R	PPSJ7	PPSJ6	0	0	0	0	PPSJ1	PPSJ0
W								
Reset	0	0	—	—	—	—	0	0

= Unimplemented or Reserved

Figure 5-44. Port J Polarity Select Register (PPSJ)

Read: Anytime. Write: Anytime.

This register serves a dual purpose by selecting the polarity of the active interrupt edge as well as selecting a pull-up or pull-down device if enabled.

Table 5-39. PPSJ Field Descriptions

Field	Description
7, 6, 1, 0 PPSJ[7:6] PPSJ[1:0]	**Polarity Select Port J** 0 Falling edge on the associated port J pin sets the associated flag bit in the PIFJ register. A pull-up device is connected to the associated port J pin, if enabled by the associated bit in register PERJ and if the port is used as general purpose input or as IIC port. 1 Rising edge on the associated port J pin sets the associated flag bit in the PIFJ register. A pull-down device is connected to the associated port J pin, if enabled by the associated bit in register PERJ and if the port is used as input.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0011 ATDDR0L	10-BIT	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	8-BIT	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	W								
0x0012 ATDDR1H	10-BIT	0	0	0	0	0	0	BIT 9 MSB	BIT 8
	8-BIT	0	0	0	0	0	0	0	0
	W								
0x0013 ATDDR1L	10-BIT	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	8-BIT	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	W								
0x0014 ATDDR2H	10-BIT	0	0	0	0	0	0	BIT 9 MSB	BIT 8
	8-BIT	0	0	0	0	0	0	0	0
	W								
0x0015 ATDDR2L	10-BIT	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	8-BIT	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	W								
0x0016 ATDDR3H	10-BIT	0	0	0	0	0	0	BIT 9 MSB	BIT 8
	8-BIT	0	0	0	0	0	0	0	0
	W								
0x0017 ATDDR3L	10-BIT	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	8-BIT	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	W								
0x0018 ATDDR4H	10-BIT	0	0	0	0	0	0	BIT 9 MSB	BIT 8
	8-BIT	0	0	0	0	0	0	0	0
	W								
0x0019 ATDDR4L	10-BIT	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	8-BIT	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	W								
0x001A ATDD45H	10-BIT	0	0	0	0	0	0	BIT 9 MSB	BIT 8
	8-BIT	0	0	0	0	0	0	0	0
	W								
0x001B ATDD45L	10-BIT	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	8-BIT	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	W								
0x001C ATDD46H	10-BIT	0	0	0	0	0	0	BIT 9 MSB	BIT 8
	8-BIT	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 8-2. ATD Register Summary (Sheet 4 of 5)

8.5.1.5 Step 5

Configure starting channel, single/multiple channel, continuous or single sequence and result data format in ATDCTL5. Writing ATDCTL5 will start the conversion, so make sure your write ATDCTL5 in the last step.

Example: Leave CC,CB,CA clear to start on channel AN0. Write MULT=1 to convert channel AN0 to AN3 in a sequence (4 conversion per sequence selected in ATDCTL3).

8.5.2 Aborting an A/D conversion

8.5.2.1 Step 1

Write to ATDCTL4. This will abort any ongoing conversion sequence.

(Do not use write to other ATDCTL registers to abort, as this under certain circumstances might not work correctly.)

8.5.2.2 Step 2

Disable the ATD Interrupt by writing ASCIE=0 in ATDCTL2.

It is important to clear the interrupt enable at this point, prior to step 3, as depending on the device clock gating it may not always be possible to clear it or the SCF flag once the module is disabled (ADPU=0).

8.5.2.3 Step 3

Clear the SCF flag by writing a 1 in ATDSTAT0.

(Remaining flags will be cleared with the next start of a conversions, but SCF flag should be cleared to avoid SCF interrupt.)

8.5.2.4 Step 4

Power down ATD by writing ADPU=0 in ATDCTL2.

8.6 Resets

At reset the ATD is in a power down state. The reset state of each individual bit is listed within the Register Description section (see Section 8.3, “Memory Map and Register Definition”), which details the registers and their bit-field.

8.7 Interrupts

The interrupt requested by the ATD is listed in Table 8-24. Refer to the device overview chapter for related vector address and priority.

Table 8-24. ATD Interrupt Vectors

Interrupt Source	CCR Mask	Local Enable
Sequence complete interrupt	I bit	ASCIE in ATDCTL2

See register descriptions for further details.

Table 10-8. Time Segment 1 Values

TSEG13	TSEG12	TSEG11	TSEG10	Time segment 1
0	0	0	0	1 Tq clock cycle[1]
0	0	0	1	2 Tq clock cycles[1]
0	0	1	0	3 Tq clock cycles[1]
0	0	1	1	4 Tq clock cycles
:	:	:	:	:
1	1	1	0	15 Tq clock cycles
1	1	1	1	16 Tq clock cycles

1 This setting is not valid. Please refer to Table 10-34 for valid settings.

The bit time is determined by the oscillator frequency, the baud rate prescaler, and the number of time quanta (Tq) clock cycles per bit (as shown in Table 10-7 and Table 10-8).

Eqn. 10-1

$$\text{Bit Time} = \frac{(\text{Prescaler value})}{f_{CANCLK}} \bullet (1 + \text{TimeSegment1} + \text{TimeSegment2})$$

10.3.2.5 MSCAN Receiver Flag Register (CANRFLG)

A flag can be cleared only by software (writing a 1 to the corresponding bit position) when the condition which caused the setting is no longer valid. Every flag has an associated interrupt enable bit in the CANRIER register.

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	WUPIF	CSCIF	RSTAT1	RSTAT0	TSTAT1	TSTAT0	OVRIF	RXF
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-8. MSCAN Receiver Flag Register (CANRFLG)

NOTE

The CANRFLG register is held in the reset state[1] when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable again as soon as the initialization mode is exited (INITRQ = 0 and INITAK = 0).

Read: Anytime
Write: Anytime when out of initialization mode, except RSTAT[1:0] and TSTAT[1:0] flags which are read-only; write of 1 clears flag; write of 0 is ignored.

1. The RSTAT[1:0], TSTAT[1:0] bits are not affected by initialization mode.

Figure 11-18 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.

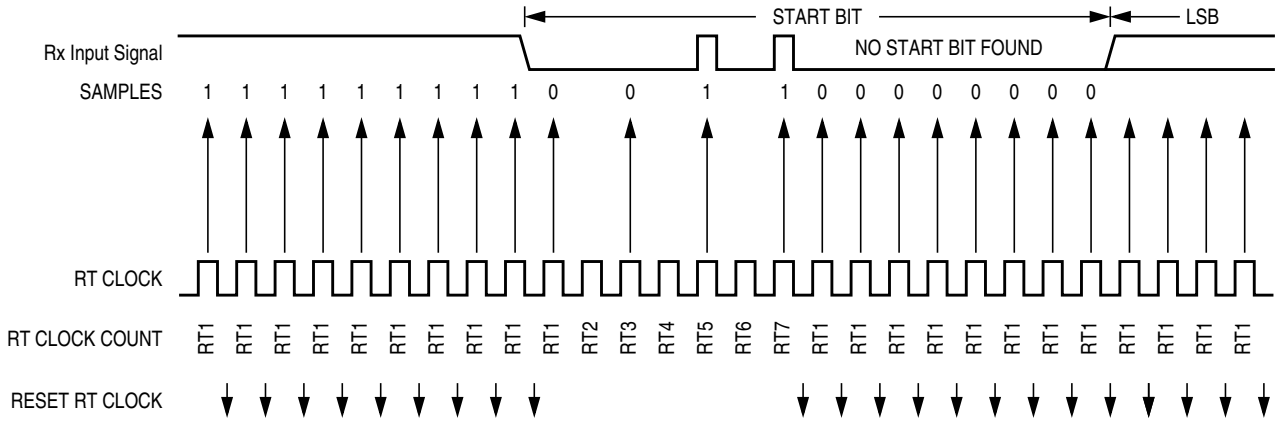


Figure 11-18. Start Bit Search Example 5

In Figure 11-19, 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.

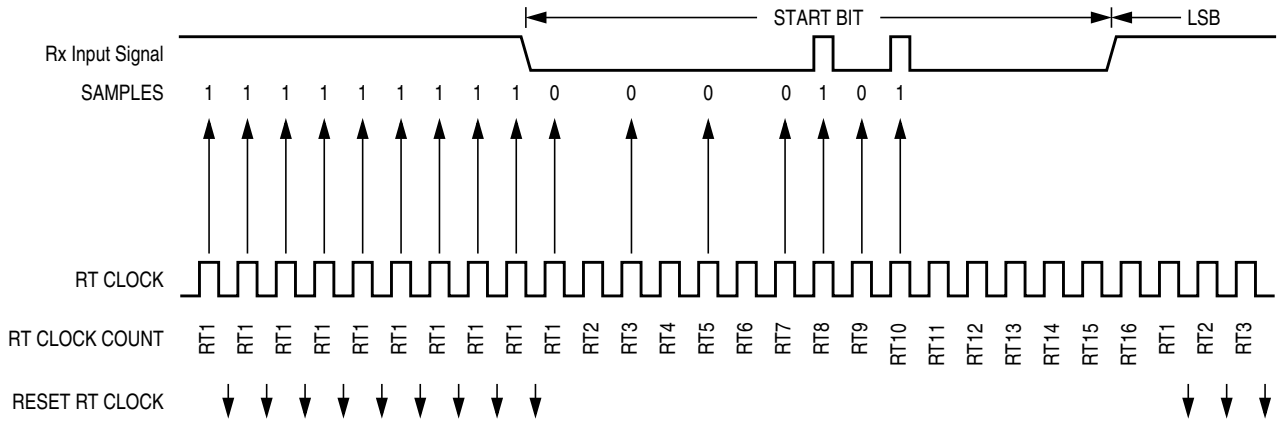


Figure 11-19. Start Bit Search Example 6

Table 13-1. PWME Field Descriptions (continued)

Field	Description
1 PWME1	**Pulse Width Channel 1 Enable** 0 Pulse width channel 1 is disabled. 1 Pulse width channel 1 is enabled. The pulse modulated signal becomes available at PWM, output bit 1 when its clock source begins its next cycle.
0 PWME0	**Pulse Width Channel 0 Enable** 0 Pulse width channel 0 is disabled. 1 Pulse width channel 0 is enabled. The pulse modulated signal becomes available at PWM, output bit 0 when its clock source begins its next cycle. If CON01 = 1, then bit has no effect and PWM output line0 is disabled.

13.3.2.2 PWM Polarity Register (PWMPOL)

The starting polarity of each PWM channel waveform is determined by the associated PPOLx bit in the PWMPOL register. If the polarity bit is one, the PWM channel output is high at the beginning of the cycle and then goes low when the duty count is reached. Conversely, if the polarity bit is zero, the output starts low and then goes high when the duty count is reached.

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R W	PPOL7	PPOL6	PPOL5	PPOL4	PPOL3	PPOL2	PPOL1	PPOL0
Reset	0	0	0	0	0	0	0	0

Figure 13-4. PWM Polarity Register (PWMPOL)

Read: Anytime

Write: Anytime

NOTE

PPOLx register bits can be written anytime. If the polarity is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition

Table 13-2. PWMPOL Field Descriptions

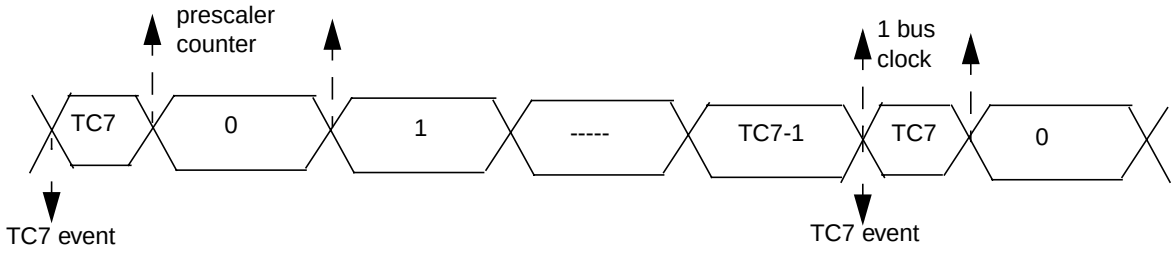
Field	Description
7–0 PPOL[7:0]	**Pulse Width Channel 7–0 Polarity Bits** 0 PWM channel 7–0 outputs are low at the beginning of the period, then go high when the duty count is reached. 1 PWM channel 7–0 outputs are high at the beginning of the period, then go low when the duty count is reached.

13.3.2.3 PWM Clock Select Register (PWMCLK)

Each PWM channel has a choice of two clocks to use as the clock source for that channel as described below.

Note: in Figure 14-29,if PR[2:0] is equal to 0, one prescaler counter equal to one bus clock

Figure 14-29. The TCNT cycle diagram under TCRE=1 condition



14.4.4 Pulse Accumulator

The pulse accumulator (PACNT) is a 16-bit counter that can operate in two modes:

Event counter mode — Counting edges of selected polarity on the pulse accumulator input pin, PAI.

Gated time accumulation mode — Counting pulses from a divide-by-64 clock. The PAMOD bit selects the mode of operation.

The minimum pulse width for the PAI input is greater than two bus clocks.

14.4.5 Event Counter Mode

Clearing the PAMOD bit configures the PACNT for event counter operation. An active edge on the IOC7 pin increments the pulse accumulator counter. The PEDGE bit selects falling edges or rising edges to increment the count.

NOTE

The PACNT input and timer channel 7 use the same pin IOC7. To use the IOC7, disconnect it from the output logic by clearing the channel 7 output mode and output level bits, OM7 and OL7. Also clear the channel 7 output compare 7 mask bit, OC7M7.

The Pulse Accumulator counter register reflect the number of active input edges on the PACNT input pin since the last reset.

The PAOVF bit is set when the accumulator rolls over from 0xFFFF to 0x0000. The pulse accumulator overflow interrupt enable bit, PAOVI, enables the PAOVF flag to generate interrupt requests.

NOTE

The pulse accumulator counter can operate in event counter mode even when the timer enable bit, TEN, is clear.

Table 16-2. BDMSTS Field Descriptions

Field	Description
7 ENBDM	**Enable BDM** — This bit controls whether the BDM is enabled or disabled. When enabled, BDM can be made active to allow firmware commands to be executed. When disabled, BDM cannot be made active but BDM hardware commands are allowed. 0 BDM disabled 1 BDM enabled **Note:** ENBDM is set by the firmware immediately out of reset in special single-chip mode. In secure mode, this bit will not be set by the firmware until after the EEPROM and FLASH erase verify tests are complete.
6 BDMACT	**BDM Active Status** — This bit becomes set upon entering BDM. The standard BDM firmware lookup table is then enabled and put into the memory map. BDMACT is cleared by a carefully timed store instruction in the standard BDM firmware as part of the exit sequence to return to user code and remove the BDM memory from the map. 0 BDM not active 1 BDM active
5 ENTAG	**Tagging Enable** — This bit indicates whether instruction tagging in enabled or disabled. It is set when the TAGGO command is executed and cleared when BDM is entered. The serial system is disabled and the tag function enabled 16 cycles after this bit is written. BDM cannot process serial commands while tagging is active. 0 Tagging not enabled or BDM active 1 Tagging enabled
4 SDV	**Shift Data Valid** — This bit is set and cleared by the BDM hardware. It is set after data has been transmitted as part of a firmware read command or after data has been received as part of a firmware write command. It is cleared when the next BDM command has been received or BDM is exited. SDV is used by the standard BDM firmware to control program flow execution. 0 Data phase of command not complete 1 Data phase of command is complete
3 TRACE	**TRACE1 BDM Firmware Command is Being Executed** — This bit gets set when a BDM TRACE1 firmware command is first recognized. It will stay set as long as continuous back-to-back TRACE1 commands are executed. This bit will get cleared when the next command that is not a TRACE1 command is recognized. 0 TRACE1 command is not being executed 1 TRACE1 command is being executed

Name[1]		Bit 7	6	5	4	3	2	1	Bit 0
0x0022 DBGTBH	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
0x0023 DBGTBL	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
0x0024 DBGCNT	R	TBF	0	CNT					
	W								
0x0025 DBGCCX(2)	R	PAGSEL		EXTCMP					
	W								
0x0026 DBGCCH(2)	R	Bit 15	14	13	12	11	10	9	Bit 8
	W								
0x0027 DBGCCL(2)	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0028 DBGC2 BKPCT0	R	BKABEN	FULL	BDM	TAGAB	BKCEN	TAGC	RWCEN	RWC
	W								
0x0029 DBGC3 BKPCT1	R	BKAMBH	BKAMBL	BKBMBH	BKBMBL	RWAEN	RWA	RWBEN	RWB
	W								
0x002A DBGCAX BKP0X	R	PAGSEL		EXTCMP					
	W								
0x002B DBGCAH BKP0H	R	Bit 15	14	13	12	11	10	9	Bit 8
	W								
0x002C DBGCAL BKP0L	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x002D DBGCBX BKP1X	R	PAGSEL		EXTCMP					
	W								
0x002E DBGCBH BKP1H	R	Bit 15	14	13	12	11	10	9	Bit 8
	W								
0x002F DBGCBL BKP1L	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								

= Unimplemented or Reserved

Figure 17-3. DBG Register Summary (continued)

17.3.2.2 Debug Status and Control Register (DBGSC)

Module Base + 0x0021
Starting address location affected by INITRG register setting.

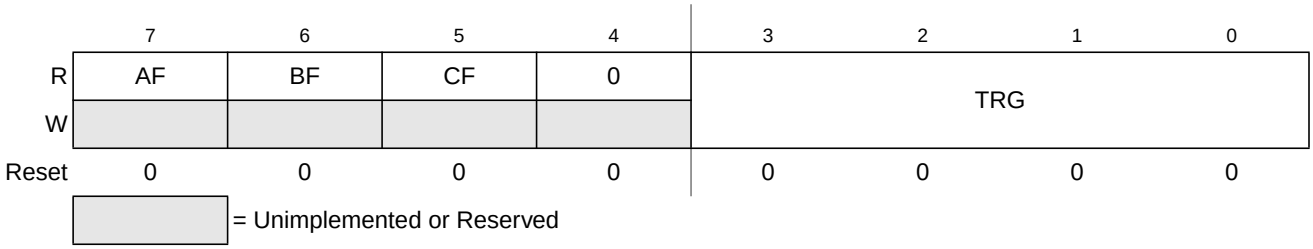


Figure 17-5. Debug Status and Control Register (DBGSC)

Table 17-5. DBGSC Field Descriptions

Field	Description
7 AF	**Trigger A Match Flag** — The AF bit indicates if trigger A match condition was met since arming. This bit is cleared when ARM in DBGC1 is written to a 1 or on any write to this register. 0 Trigger A did not match 1 Trigger A match
6 BF	**Trigger B Match Flag** — The BF bit indicates if trigger B match condition was met since arming.This bit is cleared when ARM in DBGC1 is written to a 1 or on any write to this register. 0 Trigger B did not match 1 Trigger B match
5 CF	**Comparator C Match Flag** — The CF bit indicates if comparator C match condition was met since arming.This bit is cleared when ARM in DBGC1 is written to a 1 or on any write to this register. 0 Comparator C did not match 1 Comparator C match
3:0 TRG	**Trigger Mode Bits** — The TRG bits select the trigger mode of the DBG module as shown Table 17-6. See Section 17.4.2.5, “Trigger Modes,” for more detail.

Table 17-6. Trigger Mode Encoding

TRG Value	Meaning
0000	A only
0001	A or B
0010	A then B
0011	Event only B
0100	A then event only B
0101	A and B (full mode)
0110	A and Not B (full mode)
0111	Inside range
1000	Outside range
1001 ↓ 1111	Reserved (Defaults to A only)

The PE4/ECLK pin is initially configured as ECLK output with stretch. The E clock output function depends upon the settings of the NECLK bit in the PEAR register, the IVIS bit in the MODE register and the ESTR bit in the EBICTL register. In normal expanded narrow mode, the E clock is available for use in external select decode logic or as a constant speed clock for use in the external application system.

The PE2/R/W pin is initially configured as a general purpose input with an internal pull resistor enabled but this pin can be reconfigured as the R/$\overline{W}$ bus control signal by writing “1” to the RDWE bit in PEAR. If the expanded narrow system includes external devices that can be written such as RAM, the RDWE bit would need to be set before any attempt to write to an external location. If there are no writable resources in the external system, PE2 can be left as a general purpose I/O pin.

19.4.3.1.4 Emulation Expanded Wide Mode

In expanded wide modes, Ports A and B are configured as a 16-bit multiplexed address and data bus and Port E provides bus control and status signals. These signals allow external memory and peripheral devices to be interfaced to the MCU. These signals can also be used by a logic analyzer to monitor the progress of application programs.

The bus control related pins in Port E (PE7/NOACC, PE6/MODB/IPIPE1, PE5/MODA/IPIPE0, PE4/ECLK, PE3/$\overline{LSTRB}$/$\overline{TAGLO}$, and PE2/R/$\overline{W}$) are all configured to serve their bus control output functions rather than general purpose I/O. Notice that writes to the bus control enable bits in the PEAR register in emulation mode are restricted.

19.4.3.1.5 Emulation Expanded Narrow Mode

Expanded narrow modes are intended to allow connection of single 8-bit external memory devices for lower cost systems that do not need the performance of a full 16-bit external data bus. Accesses to internal resources that have been mapped external (i.e. PORTA, PORTB, DDRA, DDRB, PORTE, DDRE, PEAR, PUCR, RDRIV) will be accessed with a 16-bit data bus on Ports A and B. Accesses of 16-bit external words to addresses which are normally mapped external will be broken into two separate 8-bit accesses using Port A as an 8-bit data bus. Internal operations continue to use full 16-bit data paths. They are only visible externally as 16-bit information if IVIS=1.

Ports A and B are configured as multiplexed address and data output ports. During external accesses, address A15, data D15 and D7 are associated with PA7, address A0 is associated with PB0 and data D8 and D0 are associated with PA0. During internal visible accesses and accesses to internal resources that have been mapped external, address A15 and data D15 is associated with PA7 and address A0 and data D0 is associated with PB0.

The bus control related pins in Port E (PE7/NOACC, PE6/MODB/IPIPE1, PE5/MODA/IPIPE0, PE4/ECLK, PE3/$\overline{LSTRB}$/$\overline{TAGLO}$, and PE2/R/$\overline{W}$) are all configured to serve their bus control output functions rather than general purpose I/O. Notice that writes to the bus control enable bits in the PEAR register in emulation mode are restricted.

The main difference between special modes and normal modes is that some of the bus control and system control signals cannot be written in emulation modes.