Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	HCS12X
Core Size	16-Bit
Speed	80MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	59
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12xdg128caa

The CAN modules may be configured to have their clock sources derived either from the bus clock or directly from the oscillator clock. This allows the user to select its clock based on the required jitter performance. Consult MSCAN block description for more details on the operation and configuration of the CAN blocks.

In order to ensure the presence of the clock the MCU includes an on-chip clock monitor connected to the output of the oscillator. The clock monitor can be configured to invoke the PLL self-clocking mode or to generate a system reset if it is allowed to time out as a result of no oscillator clock being present.

In addition to the clock monitor, the MCU also provides a clock quality checker which performs a more accurate check of the clock. The clock quality checker counts a predetermined number of clock edges within a defined time window to insure that the clock is running. The checker can be invoked following specific events such as on wake-up or clock monitor failure.

1.4 Chip Configuration Summary

CAUTION

Emulation single chip mode, Normal expanded mode, Emulation expanded mode and ROMCTL/EROMCTL functionality is only available on parts with external bus interface in 144 LQFP package. see Appendix E Derivative Differences.

The MCU can operate in six different modes. The different modes, the state of ROMCTL and EROMCTL signal on rising edge of $\overline{\text{RESET}}$, and the security state of the MCU affects the following device characteristics:

- External bus interface configuration
- Flash in memory map, or not
- Debug features enabled or disabled

The operating mode out of reset is determined by the states of the MODC, MODB, and MODA signals during reset (see Table 1-9). The MODC, MODB, and MODA bits in the MODE register show the current operating mode and provide limited mode switching during operation. The states of the MODC, MODB, and MODA signals are latched into these bits on the rising edge of $\overline{\text{RESET}}$.

In normal expanded mode and in emulation modes the ROMON bit and the EROMON bit in the MMCCTL1 register defines if the on chip flash memory is the memory map, or not. (See Table 1-9.) For a detailed description of the ROMON and EROMON bits refer to the S12X_MMC section.

The state of the ROMCTL signal is latched into the ROMON bit in the MMCCTL1 register on the rising edge of $\overline{\text{RESET}}$. The state of the EROMCTL signal is latched into the EROMON bit in the MISC register on the rising edge of $\overline{\text{RESET}}$.

The MCU can operate in two different modes. The operating mode out of reset is determined by the state of the MODC signal during reset. The MODC bit in the MODE register shows the current operating mode and provide limited mode switching during operation. The state of the MODC signal is latched into this bit on the rising edge of $\overline{\text{RESET}}$.

Table 2-5. PLLCTL Field Descriptions (continued)

Field	Description
2 PRE	**RTI Enable during Pseudo Stop Bit** — PRE enables the RTI during pseudo stop mode. Write anytime. 0 RTI stops running during pseudo stop mode. 1 RTI continues running during pseudo stop mode. **Note:** If the PRE bit is cleared the RTI dividers will go static while pseudo stop mode is active. The RTI dividers will not initialize like in wait mode with RTIWAI bit set.
1 PCE	**COP Enable during Pseudo Stop Bit** — PCE enables the COP during pseudo stop mode. Write anytime. 0 COP stops running during pseudo stop mode 1 COP continues running during pseudo stop mode **Note:** If the PCE bit is cleared, the COP dividers will go static while pseudo stop mode is active. The COP dividers will not initialize like in wait mode with COPWAI bit set.
0 SCME	**Self Clock Mode Enable Bit** Normal modes: Write once Special modes: Write anytime SCME can not be cleared while operating in self clock mode (SCM = 1). 0 Detection of crystal clock failure causes clock monitor reset (see Section 2.5.2, “Clock Monitor Reset”). 1 Detection of crystal clock failure forces the MCU in self clock mode (see Section 2.4.2.2, “Self Clock Mode”).

2.3.2.8 CRG RTI Control Register (RTICTL)

This register selects the timeout period for the real time interrupt.

	7	6	5	4	3	2	1	0
R W	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
Reset	0	0	0	0	0	0	0	0

Figure 2-11. CRG RTI Control Register (RTICTL)

Read: Anytime

Write: Anytime

NOTE

A write to this register initializes the RTI counter.

Table 2-6. RTICTL Field Descriptions

Field	Description
7 RTDEC	**Decimal or Binary Divider Select Bit** — RTDEC selects decimal or binary based prescaler values. 0 Binary based divider value. See Table 2-7 1 Decimal based divider value. See Table 2-8
6–4 RTR[6:4]	**Real Time Interrupt Prescale Rate Select Bits** — These bits select the prescale rate for the RTI. See Table 2-7 and Table 2-8.
3–0 RTR[3:0]	**Real Time Interrupt Modulus Counter Select Bits** — These bits select the modulus counter target value to provide additional granularity.Table 2-7 and Table 2-8 show all possible divide values selectable by the RTICTL register. The source clock for the RTI is OSCCLK.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
ATDDR3L	10-BIT	BIT 1	BIT 0	0	0	0	0	0	0
	8-BIT	U	U	0	0	0	0	0	0
	W								
ATDDR4H	10-BIT	BIT 9 MSB	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2
	8-BIT	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	W								
ATDDR4L	10-BIT	BIT 1	BIT 0	0	0	0	0	0	0
	8-BIT	U	U	0	0	0	0	0	0
	W								
ATDD45H	10-BIT	BIT 9 MSB	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2
	8-BIT	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	W								
ATDD45L	10-BIT	BIT 1	BIT 0	0	0	0	0	0	0
	8-BIT	U	U	0	0	0	0	0	0
	W								
ATDD46H	10-BIT	BIT 9 MSB	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2
	8-BIT	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	W								
ATDDR6L	10-BIT	BIT 1	BIT 0	0	0	0	0	0	0
	8-BIT	U	U	0	0	0	0	0	0
	W								
ATDD47H	10-BIT	BIT 9 MSB	BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2
	8-BIT	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
	W								
ATDD47L	10-BIT	BIT 1	BIT 0	0	0	0	0	0	0
	8-BIT	U	U	0	0	0	0	0	0
	W								

Right Justified Result Data

Note: The read portion of the right justified result data registers has been divided to show the bit position when reading 10-bit and 8-bit conversion data. For more detailed information refer to Section 5.3.2.13, “ATD Conversion Result Registers (ATDDRx)”.

ATDDR0H	10-BIT	0	0	0	0	0	0	BIT 9 MSB	BIT 8
	8-BIT	0	0	0	0	0	0	0	0
	W								
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								

= Unimplemented or Reserved

Figure 5-2. ATD Register Summary (Sheet 3 of 5)

RTS — Return to Scheduler — RTS

Operation

Terminates the current thread of program execution and remains idle until a new thread is started by the hardware scheduler.

CCR Effects

N	Z	V	C
—	—	—	—

N: Not affected.
Z: Not affected.
V: Not affected.
C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code															Cycles
RTS	INH	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	PA

7.3.2.10 Timer Interrupt Enable Register (TIE)

	7	6	5	4	3	2	1	0
R W	C7I	C6I	C5I	C4I	C3I	C2I	C1I	C0I
Reset	0	0	0	0	0	0	0	0

Figure 7-15. Timer Interrupt Enable Register (TIE)

Read or write: Anytime

All bits reset to zero.

The bits C7I–C0I correspond bit-for-bit with the flags in the TFLG1 status register.

Table 7-13. TIE Field Descriptions

Field	Description
7:0 C[7:0]I	**Input Capture/Output Compare “x” Interrupt Enable** 0 The corresponding flag is disabled from causing a hardware interrupt. 1 The corresponding flag is enabled to cause an interrupt.

7.3.2.29 8-Bit Pulse Accumulators Holding Registers (PA3H–PA0H)

	7	6	5	4	3	2	1	0
R	PA3H7	PA3H6	PA3H5	PA3H4	PA3H3	PA3H2	PA3H1	PA3H0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-51. 8-Bit Pulse Accumulators Holding Register 3 (PA3H)

	7	6	5	4	3	2	1	0
R	PA2H7	PA2H6	PA2H5	PA2H4	PA2H3	PA2H2	PA2H1	PA2H0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-52. 8-Bit Pulse Accumulators Holding Register 2 (PA2H)

	7	6	5	4	3	2	1	0
R	PA1H7	PA1H6	PA1H5	PA1H4	PA1H3	PA1H2	PA1H1	PA1H0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-53. 8-Bit Pulse Accumulators Holding Register 1 (PA1H)

	7	6	5	4	3	2	1	0
R	PA0H7	PA0H6	PA0H5	PA0H4	PA0H3	PA0H2	PA0H1	PA0H0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-54. 8-Bit Pulse Accumulators Holding Register 0 (PA0H)

Read: Anytime.

Write: Has no effect.

All bits reset to zero.

These registers are used to latch the value of the corresponding pulse accumulator when the related bits in register ICPAR are enabled (see Section 7.4.1.3, “Pulse Accumulators”).

NOTE

Register bits PCLK0 to PCLK7 can be written anytime. If a clock select is changed while a PWM signal is being generated, a truncated or stretched pulse can occur during the transition.

Table 8-3. PWMCLK Field Descriptions

Field	Description
7 PCLK7	**Pulse Width Channel 7 Clock Select** 0 Clock B is the clock source for PWM channel 7. 1 Clock SB is the clock source for PWM channel 7.
6 PCLK6	**Pulse Width Channel 6 Clock Select** 0 Clock B is the clock source for PWM channel 6. 1 Clock SB is the clock source for PWM channel 6.
5 PCLK5	**Pulse Width Channel 5 Clock Select** 0 Clock A is the clock source for PWM channel 5. 1 Clock SA is the clock source for PWM channel 5.
4 PCLK4	**Pulse Width Channel 4 Clock Select** 0 Clock A is the clock source for PWM channel 4. 1 Clock SA is the clock source for PWM channel 4.
3 PCLK3	**Pulse Width Channel 3 Clock Select** 0 Clock B is the clock source for PWM channel 3. 1 Clock SB is the clock source for PWM channel 3.
2 PCLK2	**Pulse Width Channel 2 Clock Select** 0 Clock B is the clock source for PWM channel 2. 1 Clock SB is the clock source for PWM channel 2.
1 PCLK1	**Pulse Width Channel 1 Clock Select** 0 Clock A is the clock source for PWM channel 1. 1 Clock SA is the clock source for PWM channel 1.
0 PCLK0	**Pulse Width Channel 0 Clock Select** 0 Clock A is the clock source for PWM channel 0. 1 Clock SA is the clock source for PWM channel 0.

8.3.2.4 PWM Prescale Clock Select Register (PWMPRCLK)

This register selects the prescale clock source for clocks A and B independently.

	7	6	5	4	3	2	1	0
R	0	PCKB2	PCKB1	PCKB0	0	PCKA2	PCKA1	PCKA0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 8-6. PWM Prescale Clock Select Register (PWMPRCLK)

Read: Anytime

Write: Anytime

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period register will go directly to the latches as well as the buffer.

NOTE

Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active period due to the double buffering scheme.

See Section 8.4.2.3, “PWM Period and Duty” for more information.

To calculate the output period, take the selected clock source period for the channel of interest (A, B, SA, or SB) and multiply it by the value in the period register for that channel:

- Left aligned output (CAEx = 0)
- PWMx Period = Channel Clock Period * PWMPERx Center Aligned Output (CAEx = 1)

  PWMx Period = Channel Clock Period * (2 * PWMPERx)

For boundary case programming values, please refer to Section 8.4.2.8, “PWM Boundary Cases”.

	7	6	5	4	3	2	1	0
R W	Bit 7	6	5	4	3	2	1	Bit 0
Reset	1	1	1	1	1	1	1	1

Figure 8-15. PWM Channel Period Registers (PWMPERx)

Read: Anytime

Write: Anytime

8.3.2.14 PWM Channel Duty Registers (PWMDTYx)

There is a dedicated duty register for each channel. The value in this register determines the duty of the associated PWM channel. The duty value is compared to the counter and if it is equal to the counter value a match occurs and the output changes state.

The duty registers for each channel are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM will always be either the old duty waveform or the new duty waveform, not some variation in between. If the channel is not enabled, then writes to the duty register will go directly to the latches as well as the buffer.

NOTE

The CANTBSEL register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK=1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Read: Find the lowest ordered bit set to 1, all other bits will be read as 0
Write: Anytime when not in initialization mode

Table 10-15. CANTBSEL Register Field Descriptions

Field	Description
2:0 TX[2:0]	**Transmit Buffer Select** — The lowest numbered bit places the respective transmit buffer in the CANTXFG register space (e.g., TX1 = 1 and TX0 = 1 selects transmit buffer TX0; TX1 = 1 and TX0 = 0 selects transmit buffer TX1). Read and write accesses to the selected transmit buffer will be blocked, if the corresponding TXEx bit is cleared and the buffer is scheduled for transmission (see Section 10.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)"). 0 The associated message buffer is deselected 1 The associated message buffer is selected, if lowest numbered bit

The following gives a short programming example of the usage of the CANTBSEL register:

To get the next available transmit buffer, application software must read the CANTFLG register and write this value back into the CANTBSEL register. In this example Tx buffers TX1 and TX2 are available. The value read from CANTFLG is therefore 0b0000_0110. When writing this value back to CANTBSEL, the Tx buffer TX1 is selected in the CANTXFG because the lowest numbered bit set to 1 is at bit position 1. Reading back this value out of CANTBSEL results in 0b0000_0010, because only the lowest numbered bit position set to 1 is presented. This mechanism eases the application software the selection of the next available Tx buffer.

- LDD CANTFLG; value read is 0b0000_0110
- STD CANTBSEL; value written is 0b0000_0110
- LDD CANTBSEL; value read is 0b0000_0010

If all transmit message buffers are deselected, no accesses are allowed to the CANTXFG registers.

10.3.2.12 MSCAN Identifier Acceptance Control Register (CANIDAC)

The CANIDAC register is used for identifier acceptance control as described below.

	7	6	5	4	3	2	1	0
R	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-15. MSCAN Identifier Acceptance Control Register (CANIDAC)

12.4.3 Transmission Formats

During an SPI transmission, data is transmitted (shifted out serially) and received (shifted in serially) simultaneously. The serial clock (SCK) synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows selection of an individual slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. Optionally, on a master SPI device, the slave select line can be used to indicate multiple-master bus contention.

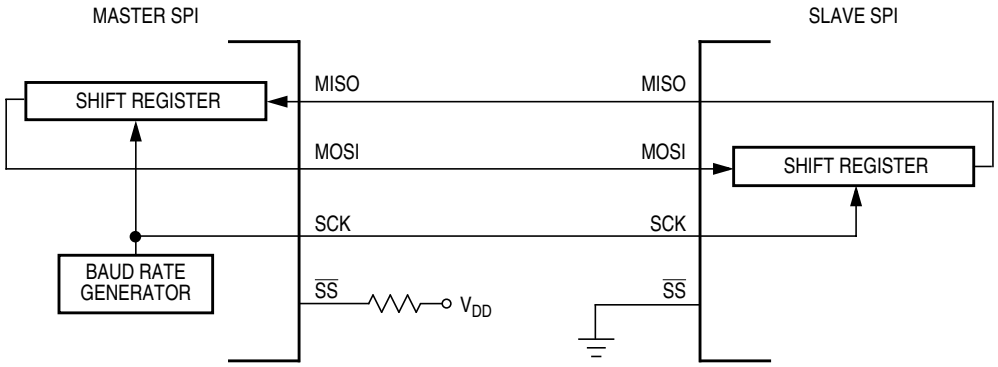

Figure 12-10. Master/Slave Transfer Block Diagram

12.4.3.1 Clock Phase and Polarity Controls

Using two bits in the SPI control register 1, software selects one of four combinations of serial clock phase and polarity.

The CPOL clock polarity control bit specifies an active high or low clock and has no significant effect on the transmission format.

The CPHA clock phase control bit selects one of two fundamentally different transmission formats.

Clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

12.4.3.2 CPHA = 0 Transfer Format

The first edge on the SCK line is used to clock the first data bit of the slave into the master and the first data bit of the master into the slave. In some peripherals, the first bit of the slave's data is available at the slave's data out pin as soon as the slave is selected. In this format, the first SCK edge is issued a half cycle after $\overline{SS}$ has become low.

A half SCK cycle later, the second edge appears on the SCK line. When this second edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the shift register, depending on LSBFE bit.

After this second edge, the next bit of the SPI master data is transmitted out of the serial data output pin of the master to the serial input pin on the slave. This process continues for a total of 16 edges on the SCK line, with data being latched on odd numbered edges and shifted on even numbered edges.

If the interrupt source is unknown (for example, in the case where an interrupt request becomes inactive after the interrupt has been recognized, but prior to the vector request), the vector address supplied to the CPU will default to that of the spurious interrupt vector.

NOTE

Care must be taken to ensure that all exception requests remain active until the system begins execution of the applicable service routine; otherwise, the exception request may not get processed at all or the result may be a spurious interrupt request (vector at address (vector base + 0x0010)).

16.4.5 Reset Exception Requests

The XINT supports three system reset exception request types (please refer to CRG for details):

1. Pin reset, power-on reset, low-voltage reset, or illegal address reset
2. Clock monitor reset request
3. COP watchdog reset request

16.4.6 Exception Priority

The priority (from highest to lowest) and address of all exception vectors issued by the XINT upon request by the CPU is shown in Table 16-8.

Table 16-8. Exception Vector Map and Priority

Vector Address[1]	Source
0xFFFE	Pin reset, power-on reset, low-voltage reset, illegal address reset
0xFFFC	Clock monitor reset
0xFFFA	COP watchdog reset
(Vector base + 0x00F8)	Unimplemented opcode trap
(Vector base + 0x00F6)	Software interrupt instruction (SWI) or BDM vector request
(Vector base + 0x00F4)	$\overline{\text{XIRQ}}$ interrupt request
(Vector base + 0x00F2)	$\overline{\text{IRQ}}$ interrupt request
(Vector base + 0x00F0–0x0012)	Device specific I bit maskable interrupt sources (priority determined by the associated configuration registers, in descending order)
(Vector base + 0x0010)	Spurious interrupt

1 16 bits vector address based

18.3.2.7 EEPROM Page Index Register (EPAGE)

Address: 0x0017

	7	6	5	4	3	2	1	0
R W	EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
Reset	1	1	1	1	1	1	1	0

Figure 18-13. EEPROM Page Index Register (EPAGE)

Read: Anytime

Write: Anytime

These eight index bits are used to page 1 KByte blocks into the EEPROM page window located in the local (CPU or BDM) memory map from address 0x0800 to address 0x0BFF (see Figure 18-14). This supports accessing up to 256 Kbytes of EEPROM (in the Global map) within the 64 KByte Local map. The EEPROM page index register is effectively used to construct paged EEPROM addresses in the Local map format.

CAUTION

XGATE write access to this register during an CPU access which makes use of this register could lead to unexpected results.

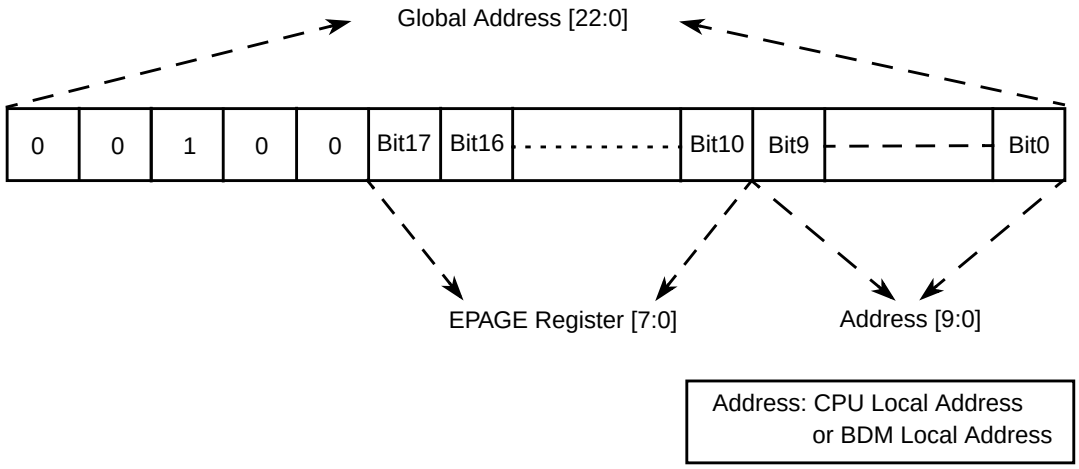

Figure 18-14. EPAGE Address Mapping

Table 18-13. EPAGE Field Descriptions

Field	Description
7–0 EP[7:0]	**EEPROM Page Index Bits 7–0** — These page index bits are used to select which of the 256 EEPROM array pages is to be accessed in the EEPROM Page Window.

The reset value of 0xFE ensures that there is a linear EEPROM space available between addresses 0x0800 and 0x0FFF out of reset.

The fixed 1K page 0x0C00–0x0FFF of EEPROM is equivalent to page 255 (page number 0xFF).

When program control returns from a tagged breakpoint using an RTI or BDM GO command without program counter modification it will return to the instruction whose tag generated the breakpoint. Thus care must be taken to avoid re triggering a breakpoint at the same location. This can be done by reconfiguring the DBG module in the SWI routine, (SWI configuration), or by executing a TRACE command before the GO (BDM configuration) to increment the program flow past the tagged instruction.

Comparators should not be configured for the vector address range while tagging, since these addresses are not opcode addresses

22.4.1.2 Input Register

This is a read-only register and always returns the buffered state of the pin (Figure 22-76).

22.4.1.3 Data Direction Register

This register defines whether the pin is used as an input or an output.

If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 22-76).

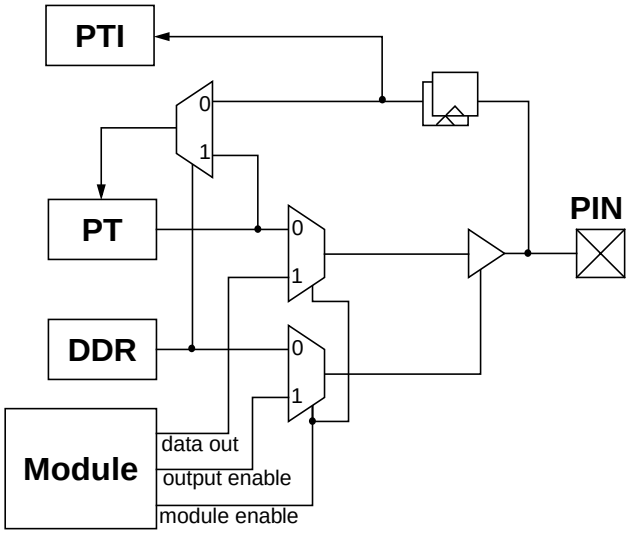

Figure 22-76. Illustration of I/O Pin Functionality

22.4.1.4 Reduced Drive Register

If the pin is used as an output this register allows the configuration of the drive strength.

22.4.1.5 Pull Device Enable Register

This register turns on a pull-up or pull-down device. It becomes active only if the pin is used as an input or as a wired-OR output.

22.4.1.6 Polarity Select Register

This register selects either a pull-up or pull-down device if enabled. It becomes active only if the pin is used as an input. A pull-up device can be activated if the pin is used as a wired-OR output. If the pin is used as an interrupt input this register selects the active interrupt edge.

22.4.1.7 Wired-OR Mode Register

If the pin is used as an output this register turns off the active high drive. This allows wired-OR type connections of outputs.

22.4.2.6 Port T

This port is associated with the ECT module. Port T pins PT[7:0] can be used for either general-purpose I/O, or with the channels of the enhanced capture timer.

22.4.2.7 Port S

This port is associated with SCI0, SCI1 and SPI0. Port S pins PS[7:0] can be used either for general-purpose I/O, or with the SCI and SPI subsystems.

The SPI0 pins can be re-routed. *Refer to Section 22.3.2.37, "Module Routing Register (MODRR)".*

NOTE

PS[7:4] are not available in 80-pin packages.

22.4.2.8 Port M

This port is associated with the SCI3, CAN4–0 and SPI0. Port M pins PM[7:0] can be used for either general purpose I/O, or with the CAN, SCI and SPI subsystems.

The CAN0, CAN4 and SPI0 pins can be re-routed. *Refer to Section 22.3.2.37, "Module Routing Register (MODRR)".*

NOTE

PM[7:6] are not available in 80-pin packages.

22.4.2.9 Port P

This port is associated with the PWM, SPI1 and SPI2. Port P pins PP[7:0] can be used for either general purpose I/O, or with the PWM and SPI subsystems.

The pins are shared between the PWM channels and the SPI1 and SPI2 modules. If the PWM is enabled the pins become PWM output channels with the exception of pin 7 which can be PWM input or output. If SPI1 or SPI2 are enabled and PWM is disabled, the respective pin configuration is determined by status bits in the SPI modules.

The SPI1 and SPI2 pins can be re-routed. *Refer to Section 22.3.2.37, "Module Routing Register (MODRR)".*

Port P offers 8 I/O pins with edge triggered interrupt capability in wired-OR fashion (Section 22.4.3, "Pin Interrupts").

NOTE

PP[6] is not available in 80-pin packages.

Table 24-61. Module Implementations on Derivatives

Number of Modules	MSCAN Modules					SPI Modules		
	CAN0	CAN1	CAN2	CAN3	CAN4	SPI0	SPI1	SPI2
5	yes	yes	yes	yes	yes	—	—	—
4	yes	yes	yes	—	yes	—	—	—
3	yes	yes	—	—	yes	yes	yes	yes
2	yes	—	—	—	yes	yes	yes	—
1	yes	—	—	—	—	yes	—	—

24.0.7 Ports

24.0.7.1 BKGD Pin

The BKGD pin is associated with the S12X_BDM and S12X_EBI modules. During reset, the BKGD pin is used as MODC input.

24.0.7.2 Port A and B

Port A pins PA[7:0] and Port B pins PB[7:0] can be used for either general-purpose I/O.

24.0.7.3 Port E

Port E pins PE[7:2] can be used for either general-purpose I/O or with the alternative functions.

Port E pin PE[7] an be used for either general-purpose I/O or as the free-running clock ECLKX2 output running at the core clock rate. The clock output is always enabled in emulation modes.

Port E pin PE[4] an be used for either general-purpose I/O or as the free-running clock ECLK output running at the bus clock rate or at the programmed divided clock rate. The clock output is always enabled in emulation modes.

Port E pin PE[1] can be used for either general-purpose input or as the level- or falling edge-sensitive $\overline{\text{IRQ}}$ interrupt input. $\overline{\text{IRQ}}$ will be enabled by setting the IRQEN configuration bit (Section 24.0.5.10, “IRQ Control Register (IRQCR)”) and clearing the I-bit in the CPU’s condition code register. It is inhibited at reset so this pin is initially configured as a simple input with a pull-up.

Port E pin PE[0] can be used for either general-purpose input or as the level-sensitive $\overline{\text{XIRQ}}$ interrupt input. $\overline{\text{XIRQ}}$ can be enabled by clearing the X-bit in the CPU’s condition code register. It is inhibited at reset so this pin is initially configured as a high-impedance input with a pull-up.

24.0.7.4 Port K

Port K pins PK[7:0] can be used for either general-purpose I/O.

25.4.2.3 Sector Erase Command

The sector erase operation will erase both words in a sector of EEPROM memory using an embedded algorithm.

An example flow to execute the sector erase operation is shown in Figure 25-20. The sector erase command write sequence is as follows:

1. Write to an EEPROM memory address to start the command write sequence for the sector erase command. The EEPROM address written determines the sector to be erased while global address bits [1:0] and the data written are ignored.
2. Write the sector erase command, 0x40, to the ECMD register.
3. Clear the CBEIF flag in the ESTAT register by writing a 1 to CBEIF to launch the sector erase command.

If an EEPROM sector to be erased is in a protected area of the EEPROM memory, the PVIOL flag in the ESTAT register will set and the sector erase command will not launch. Once the sector erase command has successfully launched, the CCIF flag in the ESTAT register will set after the sector erase operation has completed unless a new command write sequence has been buffered.

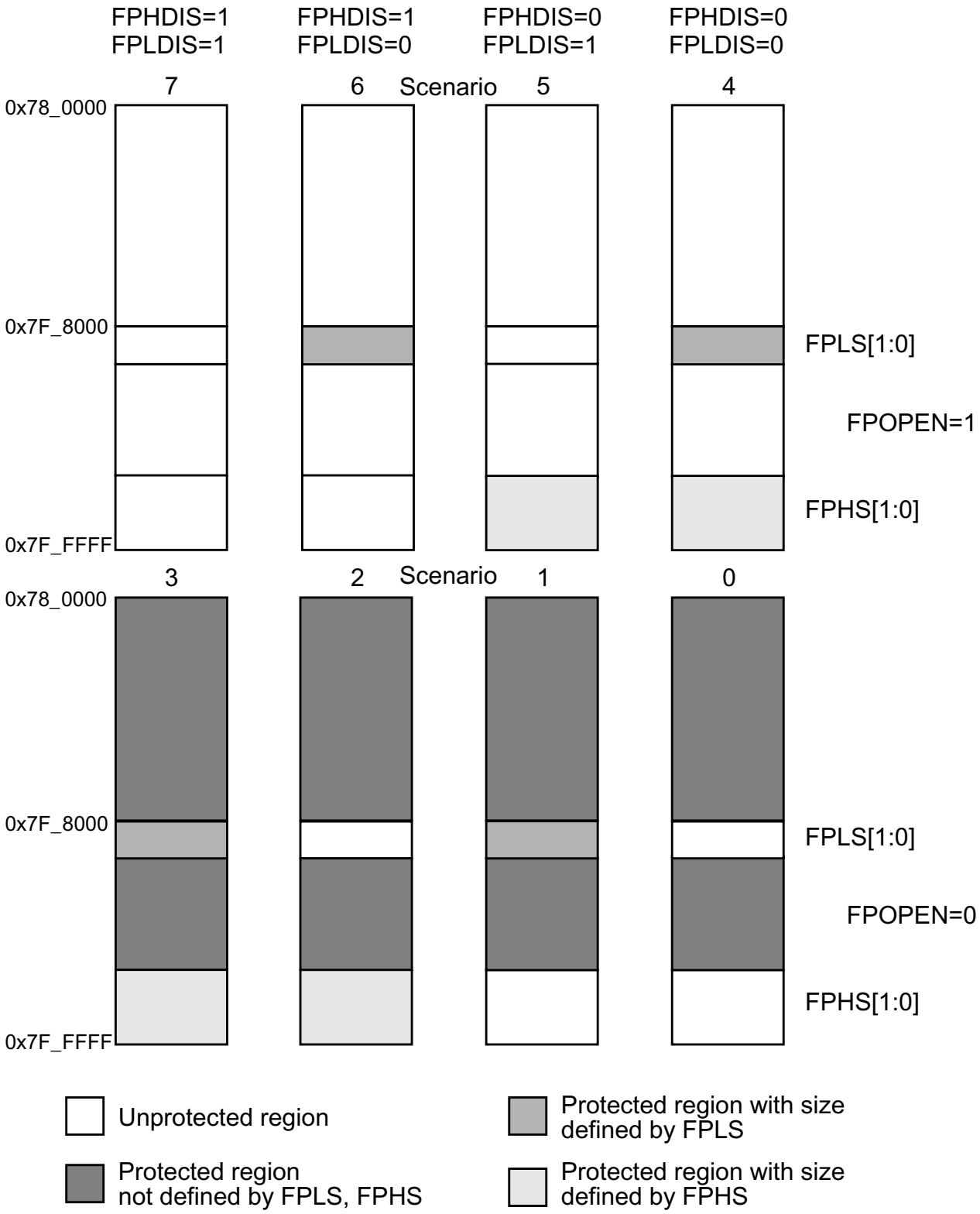

Figure 27-11. Flash Protection Scenarios

29.3.2.7 Flash Command Register (FCMD)

The FCMD register is the Flash command register.

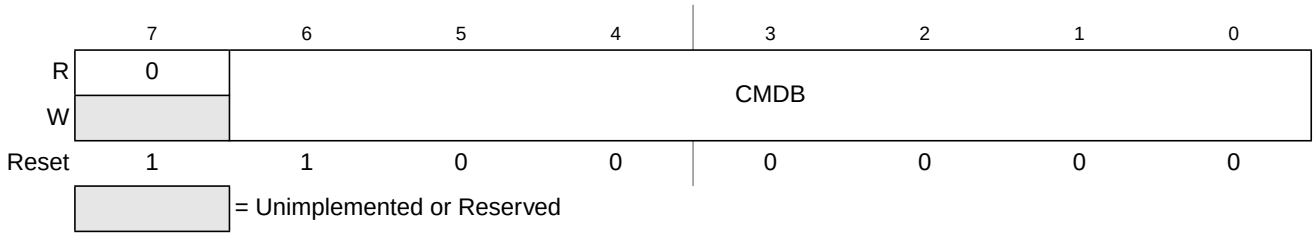

Figure 29-12. Flash Command Register (FCMD)

All CMDB bits are readable and writable during a command write sequence while bit 7 reads 0 and is not writable.

Table 29-15. FCMD Field Descriptions

Field	Description
6:0 CMDB[6:0]	**Flash Command** — Valid Flash commands are shown in Table 29-16. Writing any command other than those listed in Table 29-16 sets the ACCERR flag in the FSTAT register.

Table 29-16. Valid Flash Command List

CMDB[6:0]	NVM Command
0x05	Erase Verify
0x06	Data Compress
0x20	Word Program
0x40	Sector Erase
0x41	Mass Erase
0x47	Sector Erase Abort

29.3.2.8 Flash Control Register (FCTL)

The FCTL register is the Flash control register.

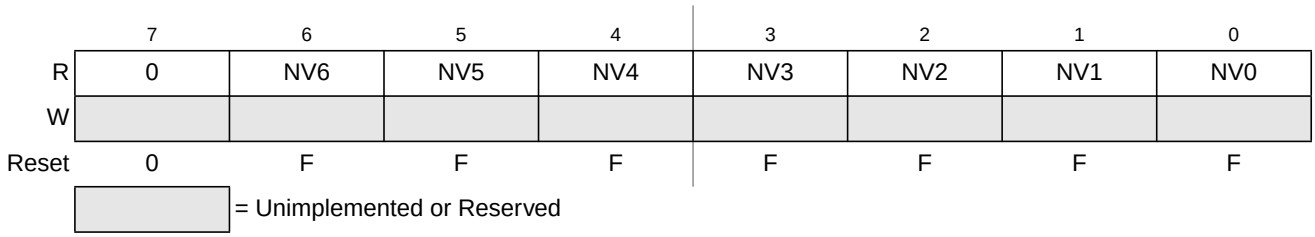

Figure 29-13. Flash Control Register (FCTL)

All bits in the FCTL register are readable but are not writable.

The FCTL NV bits are loaded from the Flash nonvolatile byte located at global address 0x7F_FF0E during the reset sequence, indicated by F in Figure 29-13.

0x0240–0x027F Port Integration Module PIM_9DX (PIM) Map (Sheet 3 of 4)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0260	PTH	R	PTH7	PTH6	PTH5	PTH4	PTH3	PTH2	PTH1	PTH0
		W								
0x0261	PTIH	R	PTIH7	PTIH6	PTIH5	PTIH4	PTIH3	PTIH2	PTIH1	PTIH0
		W								
0x0262	DDRH	R	DDRH7	DDRH7	DDRH5	DDRH4	DDRH3	DDRH2	DDRH1	DDRH0
		W								
0x0263	RDRH	R	RDRH7	RDRH6	RDRH5	RDRH4	RDRH3	RDRH2	RDRH1	RDRH0
		W								
0x0264	PERH	R	PERH7	PERH6	PERH5	PERH4	PERH3	PERH2	PERH1	PERH0
		W								
0x0265	PPSH	R	PPSH7	PPSH6	PPSH5	PPSH4	PPSH3	PPSH2	PPSH1	PPSH0
		W								
0x0266	PIEH	R	PIEH7	PIEH6	PIEH5	PIEH4	PIEH3	PIEH2	PIEH1	PIEH0
		W								
0x0267	PIFH	R	PIFH7	PIFH6	PIFH5	PIFH4	PIFH3	PIFH2	PIFH1	PIFH0
		W								
0x0268	PTJ	R	PTJ7	PTJ6	PTJ5	PTJ4	0	PTJ2	PTJ1	PTJ0
		W								
0x0269	PTIJ	R	PTIJ7	PTIJ6	PTIJ5	PTIJ4	0	PTIJ2	PTIJ1	PTIJ0
		W								
0x026A	DDRJ	R	DDRJ7	DDRJ7	DDRJ5	DDRJ4	0	DDRJ2	DDRJ1	DDRJ0
		W								
0x026B	RDRJ	R	RDRJ7	RDRJ6	RDRJ5	RDRJ4	0	RDRJ2	RDRJ1	RDRJ0
		W								
0x026C	PERJ	R	PERJ7	PERJ6	PERJ5	PERJ4	0	PERJ2	PERJ1	PERJ0
		W								
0x026D	PPSJ	R	PPSJ7	PPSJ6	PPSJ5	PPSJ4	0	PPSJ2	PPSJ1	PPSJ0
		W								
0x026E	PIEJ	R	PIEJ7	PIEJ6	PIEJ5	PIEJ4	0	PIEJ2	PIEJ1	PIEJ0
		W								
0x026F	PIFJ	R	PIFJ7	PIFJ6	PIFJ5	PIFJ4	0	PIFJ2	PIFJ1	PIFJ0
		W								
0x0270	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0271	PT1AD0	R	PT1AD07	PT1AD06	PT1AD05	PT1AD04	PT1AD03	PT1AD02	PT1AD01	PT1AD00
		W								
0x0272	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0273	DDR1AD0	R	DDR1AD07	DDR1AD06	DDR1AD05	DDR1AD04	DDR1AD03	DDR1AD02	DDR1AD01	DDR1AD01
		W								
0x0274	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0275	RDR1AD0	R	RDR1AD07	RDR1AD06	RDR1AD05	RDR1AD04	RDR1AD03	RDR1AD02	RDR1AD01	RDR1AD00
		W								
0x0276	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0277	PER1AD0	R	PER1AD07	PER1AD06	PER1AD05	PER1AD04	PER1AD03	PER1AD02	PER1AD01	PER1AD00
		W								