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, I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	59
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	20K x 8
Voltage - Supply (Vcc/Vdd)	3.15V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°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/s912xdt512j1maa

Chapter 1 Device Overview MC9S12XD-Family

1.1 Introduction

The MC9S12XD family will retain the low cost, power consumption, EMC and code-size efficiency advantages currently enjoyed by users of Freescale's existing 16-Bit MC9S12 MCU Family.

Based around an enhanced S12 core, the MC9S12XD family will deliver 2 to 5 times the performance of a 25-MHz S12 whilst retaining a high degree of pin and code compatibility with the S12.

The MC9S12XD family introduces the performance boosting XGATE module. Using enhanced DMA functionality, this parallel processing module offloads the CPU by providing high-speed data processing and transfer between peripheral modules, RAM, Flash EEPROM and I/O ports. Providing up to 80 MIPS of performance additional to the CPU, the XGATE can access all peripherals, Flash EEPROM and the RAM block.

The MC9S12XD family is composed of standard on-chip peripherals including up to 512 Kbytes of Flash EEPROM, 32 Kbytes of RAM, 4 Kbytes of EEPROM, six asynchronous serial communications interfaces (SCI), three serial peripheral interfaces (SPI), an 8-channel IC/OC enhanced capture timer, an 8-channel, 10-bit analog-to-digital converter, a 16-channel, 10-bit analog-to-digital converter, an 8-channel pulse-width modulator (PWM), five CAN 2.0 A, B software compatible modules (MSCAN12), two inter-IC bus blocks, and a periodic interrupt timer. The MC9S12XD family has full 16-bit data paths throughout.

The non-multiplexed expanded bus interface available on the 144-pin versions allows an easy interface to external memories

The inclusion of a PLL circuit allows power consumption and performance to be adjusted to suit operational requirements. System power consumption can be further improved with the new “fast exit from stop mode” feature.

In addition to the I/O ports available in each module, up to 25 further I/O ports are available with interrupt capability allowing wake-up from stop or wait mode.

Family members in 144-pin LQFP will be available with external bus interface and parts in 112-pin LQFP or 80-pin QFP package without external bus interface. See Appendix E Derivative Differences for package options.

2.4 Functional Description

2.4.1 Functional Blocks

2.4.1.1 Phase Locked Loop (PLL)

The PLL is used to run the MCU from a different time base than the incoming OSCCLK. For increased flexibility, OSCCLK can be divided in a range of 1 to 16 to generate the reference frequency. This offers a finer multiplication granularity. The PLL can multiply this reference clock by a multiple of 2, 4, 6,... 126,128 based on the SYNR register.

$$PLLCLK = 2 \times OSCCLK \times \frac{[SYNR + 1]}{[REFDV + 1]}$$

CAUTION

Although it is possible to set the two dividers to command a very high clock frequency, do not exceed the specified bus frequency limit for the MCU.
If (PLLSEL = 1), Bus Clock = PLLCLK / 2

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}.

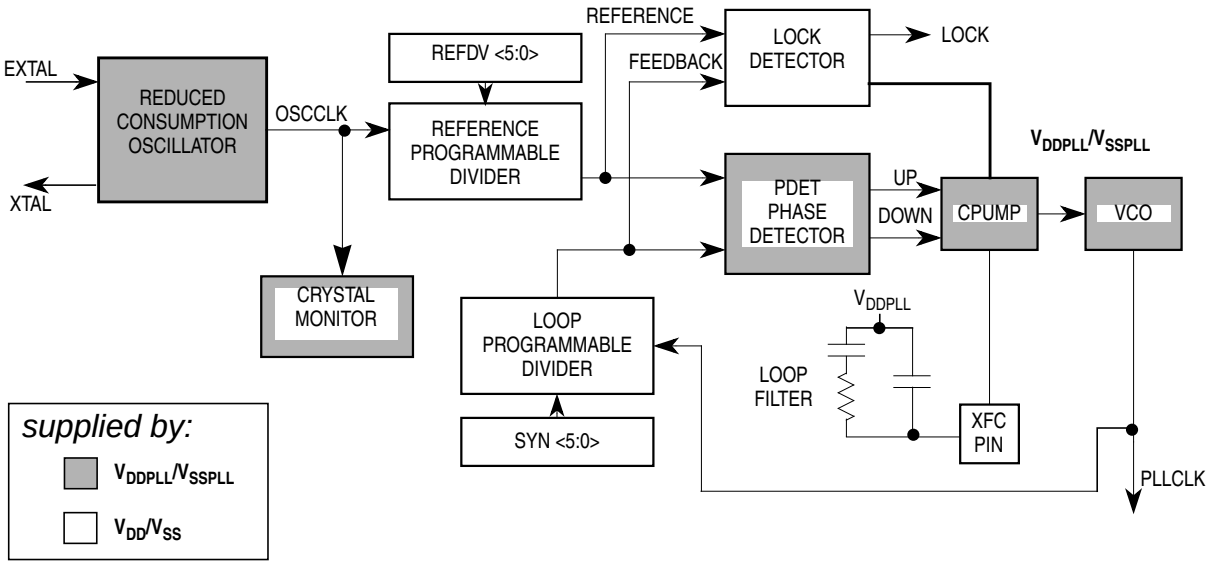

Figure 2-16. PLL Functional Diagram

Table 5-12. Clock Prescaler Values

Prescale Value	Total Divisor Value	Max. Bus Clock[1]	Min. Bus Clock[2]
00000	Divide by 2	4 MHz	1 MHz
00001	Divide by 4	8 MHz	2 MHz
00010	Divide by 6	12 MHz	3 MHz
00011	Divide by 8	16 MHz	4 MHz
00100	Divide by 10	20 MHz	5 MHz
00101	Divide by 12	24 MHz	6 MHz
00110	Divide by 14	28 MHz	7 MHz
00111	Divide by 16	32 MHz	8 MHz
01000	Divide by 18	36 MHz	9 MHz
01001	Divide by 20	40 MHz	10 MHz
01010	Divide by 22	44 MHz	11 MHz
01011	Divide by 24	48 MHz	12 MHz
01100	Divide by 26	52 MHz	13 MHz
01101	Divide by 28	56 MHz	14 MHz
01110	Divide by 30	60 MHz	15 MHz
01111	Divide by 32	64 MHz	16 MHz
10000	Divide by 34	68 MHz	17 MHz
10001	Divide by 36	72 MHz	18 MHz
10010	Divide by 38	76 MHz	19 MHz
10011	Divide by 40	80 MHz	20 MHz
10100	Divide by 42	84 MHz	21 MHz
10101	Divide by 44	88 MHz	22 MHz
10110	Divide by 46	92 MHz	23 MHz
10111	Divide by 48	96 MHz	24 MHz
11000	Divide by 50	100 MHz	25 MHz
11001	Divide by 52	104 MHz	26 MHz
11010	Divide by 54	108 MHz	27 MHz
11011	Divide by 56	112 MHz	28 MHz
11100	Divide by 58	116 MHz	29 MHz
11101	Divide by 60	120 MHz	30 MHz
11110	Divide by 62	124 MHz	31 MHz
11111	Divide by 64	128 MHz	32 MHz

1 Maximum ATD conversion clock frequency is 2 MHz. The maximum allowed bus clock frequency is shown in this column.

2 Minimum ATD conversion clock frequency is 500 kHz. The minimum allowed bus clock frequency is shown in this column.

STB

Store Byte to Memory (Low Byte)

STB

Operation

RS.L ⇒ M[RB, #OFFS5]
RS.L ⇒ M[RB, RI]
RS.L ⇒ M[RB, RI]; RI+1 ⇒ RI;
RI–1 ⇒ RI; RS.L ⇒ M[RB, RI][1]

Stores the low byte of register RD to memory.

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
STB RS, (RB, #OFFS5),	IDO5	0	1	0	1	0	RS	RB	OFFS5			Pw
STB RS, (RB, RI)	IDR	0	1	1	1	0	RS	RB	RI	0	0	Pw
STB RS, (RB, RI+)	IDR+	0	1	1	1	0	RS	RB	RI	0	1	Pw
STB RS, (RB, -RI)	-IDR	0	1	1	1	0	RS	RB	RI	1	0	Pw

1. If the same general purpose register is used as index (RI) and source register (RS), the unmodified content of the source register is written to the memory: RS.L ⇒ M[RB, RS-1]; RS-1 ⇒ RS

7.3.2.23 Input Control Overwrite Register (ICOVW)

	7	6	5	4	3	2	1	0
R W	NOVW7	NOVW6	NOVW5	NOVW4	NOVW3	NOVW2	NOVW1	NOVW0
Reset	0	0	0	0	0	0	0	0

Figure 7-45. Input Control Overwrite Register (ICOVW)

Read: Anytime

Write: Anytime

All bits reset to zero.

Table 7-29. ICOVW Field Descriptions

Field	Description
7:0 NOVW[7:0]	**No Input Capture Overwrite** 0 The contents of the related capture register or holding register can be overwritten when a new input capture or latch occurs. 1 The related capture register or holding register cannot be written by an event unless they are empty (see Section 7.4.1.1, “IC Channels”). This will prevent the captured value being overwritten until it is read or latched in the holding register.

NOTE

Changing the PWM output mode from left aligned to center aligned output (or vice versa) while channels are operating can cause irregularities in the PWM output. It is recommended to program the output mode before enabling the PWM channel.

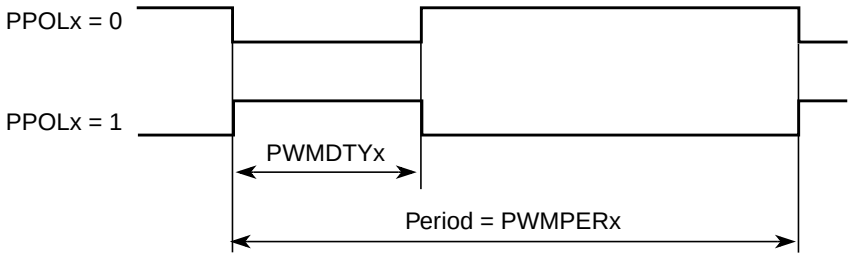

Figure 8-20. PWM Left Aligned Output Waveform

To calculate the output frequency in left aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB) and divide it by the value in the period register for that channel.

- PWMx Frequency = Clock (A, B, SA, or SB) / PWMPERx
- PWMx Duty Cycle (high time as a% of period):
  - Polarity = 0 (PPOLx = 0)
- Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
  - Polarity = 1 (PPOLx = 1)

Duty Cycle = [PWMDTYx / PWMPERx] * 100%

As an example of a left aligned output, consider the following case:

Clock Source = E, where E = 10 MHz (100 ns period)

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10 MHz/4 = 2.5 MHz

PWMx Period = 400 ns

PWMx Duty Cycle = 3/4 *100% = 75%

The output waveform generated is shown in Figure 8-21.

When the device is operating in expanded modes except emulation single-chip mode, accesses to the global addresses which are not occupied by the on-chip resources (unimplemented areas or external space) result in accesses to the external bus (see Figure 17-23).

In emulation single-chip mode, accesses to the global addresses which are not occupied by the on-chip resources (unimplemented areas) result in accesses to the external bus. CPU accesses to the global addresses which are occupied by the external space result in an illegal access reset (system reset). The BDM accesses to the external space are performed but the data is undefined.

In single-chip modes an access to any of the unimplemented areas (see Figure 17-23) by the CPU (except firmware commands) results in an illegal access reset (system reset). The BDM accesses to the unimplemented areas are performed but the data is undefined.

Misaligned word accesses to the last location (Top address) of any of the on-chip resource blocks (except RAM) by the CPU is performed in expanded modes. In single-chip modes these accesses (except Flash) result in an illegal access reset (except firmware commands).

Misaligned word accesses to the last location (top address) of the on-chip RAM by the CPU is ignored in expanded modes (read of undefined data). In single-chip modes these accesses result in an illegal access reset (except firmware commands).

No misaligned word access from the BDM module will occur. These accesses are blocked in the BDM (Refer to BDM Block Guide).

Misaligned word accesses to the last location of any global page (64 Kbyte) by using global instructions, is performed by accessing the last byte of the page and the first byte of the same page, considering the above mentioned misaligned access cases.

The non internal resources (unimplemented areas or external space) are used to generate the chip selects (CS0,CS1,CS2 and CS3) (see Figure 17-23), which are only active in normal expanded mode, emulation expanded mode, and special test mode (see Section 1.3.2.1, “MMC Control Register (MMCCTL0)”).

Table 1-21 shows the address boundaries of each chip select and the relationship with the implemented resources (internal) parameters.

Table 17-20. Global Chip Selects Memory Space

Chip Selects	Bottom Address	Top Address
$\overline{CS3}$	$00_0800	$0F_FFFF minus RAMSIZE[1]
$\overline{CS2}$	$10_0000	$13_FFFF minus EEPROMSIZE[2]
$\overline{CS2}$[3]	$14_0000	$1F_FFFF
$\overline{CS1}$	$20_0000	$3F_FFFF
$\overline{CS0}$[4]	$40_0000	$7F_FFFF minus FLASHSIZE[5]

1 External RPAGE accesses in (NX, EX and ST)

2 External EPAGE accesses in (NX, EX and ST)

3 When ROMHM is set (see ROMHM in Table 1-19) the $\overline{CS2}$ is asserted in the space occupied by this on-chip memory block.

4 When the internal NVM is enabled (see ROMON in Section 1.3.2.5, “MMC Control Register (MMCCTL1)”) the $\overline{CS0}$ is not asserted in the space occupied by this on-chip memory block.

5 External PPAGE accesses in (NX, EX and ST)

19.3.1.3 Debug Trace Control Register (DBGTCR)

0x0022

	7	6	5	4	3	2	1	0
R W	TSOURCE		TRANGE		TRCMOD		TALIGN	
Reset	0	0	0	0	0	0	0	0

Figure 19-5. Debug Trace Control Register (DBGTCR)

Read: Anytime

Write: Bits 7:6 only when DBG is neither secure nor armed.
Bits 5:0 anytime the module is disarmed.

Table 19-8. DBGTCR Field Descriptions

Field	Description
7–6 TSOURCE	**Trace Source Control Bits** — The TSOURCE bits select the data source for the tracing session. If the MCU system is secured, these bits cannot be set and tracing is inhibited. See Table 19-9.
5–4 TRANGE[5:4]	**Trace Range Bits** —The TRANGE bits allow filtering of trace information from a selected address range when tracing from the CPU in detail mode. The XGATE tracing range cannot be narrowed using these bits. To use a comparator for range filtering, the corresponding COMPE and SRC bits must remain cleared. If the COMPE bit is not clear then the comparator will also be used to generate state sequence triggers or tags. If the SRC bit is set the comparator is mapped to the XGATE busses, corrupting the trace. See Table 19-10.
3–2 TRCMOD[3:2]	**Trace Mode Bits** — See Section 19.4.5.2, “Trace Modes“ for detailed trace mode descriptions. In normal mode, change of flow information is stored. In loop1 mode, change of flow information is stored but redundant entries into trace memory are inhibited. In detail mode, address and data for all memory and register accesses is stored. See Table 19-11
1–0 TALIGN[1:0]	**Trigger Align Bits** — These bits control whether the trigger is aligned to the beginning, end or the middle of a tracing session. See Table 19-12.

Table 19-9. TSOURCE Trace Source Bit Encoding

TSOURCE	Tracing Source
00	No tracing requested
01	CPU
10[1]	XGATE
11[1, 2]	Both CPU and XGATE

1 No range limitations are allowed. Thus tracing operates as if TRANGE = 00.

2 No detail mode tracing supported. If TRCMOD =10, no information is stored.

Table 19-10. TRANGE Trace Range Encoding

TRANGE	Tracing Source
00	Trace from all addresses (No filter)
01	Trace only in address range from 0x0000 to comparator D
10	Trace only in address range from comparator C to 0x7FFFFF

20.3.2.7.2 Debug State Control Register 2 (DBGSCR2)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	SC3	SC2	SC1	SC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 20-10. Debug State Control Register 2 (DBGSCR2)

Read: Anytime

Write: Anytime when S12XDBG not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 01. The state control register 2 selects the targeted next state whilst in State2. The matches refer to the match channels of the comparator match control logic as depicted in Figure 20-1 and described in Section 20.3.2.8.1". Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 20-22. DBGSCR2 Field Descriptions

Field	Description
3–0 SC[3:0]	These bits select the targeted next state whilst in State2, based upon the match event.

Table 20-23. State2 —Sequencer Next State Selection

SC[3:0]	Description
0000	Any match triggers to state1
0001	Any match triggers to state3
0010	Any match triggers to Final State
0011	Match3 triggers to State1..... Other matches have no effect
0100	Match3 triggers to State3..... Other matches have no effect
0101	Match3 triggers to Final State..... Other matches have no effect
0110	Match0 triggers to State1..... Match1 triggers to State3..... Other matches have no effect
0111	Match1 triggers to State3..... Match0 triggers Final State..... Other matches have no effect
1000	Match0 triggers to State1..... Match2 triggers to State3..... Other matches have no effect
1001	Match2 triggers to State3..... Match0 triggers Final State..... Other matches have no effect
1010	Match1 triggers to State1..... Match3 triggers to State3..... Other matches have no effect
1011	Match3 triggers to State3..... Match1 triggers Final State..... Other matches have no effect
1100	Match2 triggers to State1..... Match3 trigger to Final State
1101	Match2 has no affect, all other matches (M0,M1,M3) trigger to Final State
1110	Reserved
1111	Reserved

The trigger priorities described in Table 20-38 dictate that in the case of simultaneous matches, the match on the lower channel number (0,1,2,3) has priority. The SC[3:0] encoding ensures that a match leading to final state has priority over all other matches.

22.3.2.12 S12X_EBI Ports Reduced Drive Register (RDRIV)

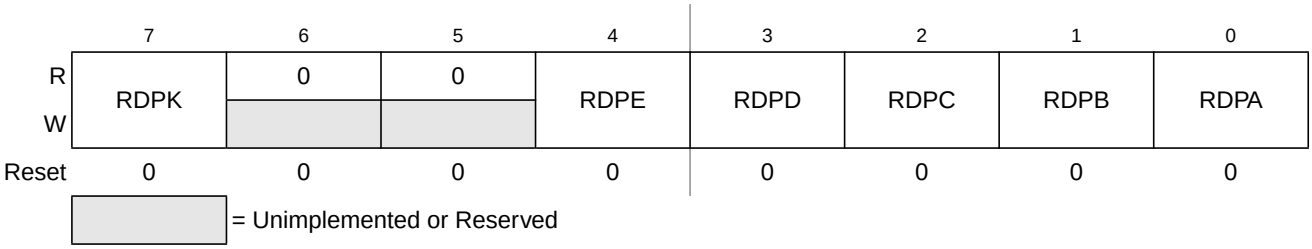

Figure 22-14. S12X_EBI Ports Reduced Drive Register (RDRIV)

Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data are read from this register.

Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

This register is used to select reduced drive for the pins associated with the S12X_EBI ports A, B, C, D, E, and K. If enabled, the pins drive at about 1/6 of the full drive strength. The reduced drive function is independent of which function is being used on a particular pin.

The reduced drive functionality does not take effect on the pins in emulation modes.

Table 22-15. RDRIV Field Descriptions

Field	Description
7 RDPK	**Reduced Drive of Port K** 0 All port K output pins have full drive enabled. 1 All port K output pins have reduced drive enabled.
4 RDPE	**Reduced Drive of Port E** 0 All port E output pins have full drive enabled. 1 All port E output pins have reduced drive enabled.
3 RDPD	**Reduced Drive of Port D** 0 All port D output pins have full drive enabled. 1 All port D output pins have reduced drive enabled.
2 RDPC	**Reduced Drive of Port C** 0 All port C output pins have full drive enabled. 1 All port C output pins have reduced drive enabled.
1 RDPB	**Reduced Drive of Port B** 0 All port B output pins have full drive enabled. 1 All port B output pins have reduced drive enabled.
0 RDPA	**Reduced Drive of Ports A** 0 All Port A output pins have full drive enabled. 1 All port A output pins have reduced drive enabled.

22.4.2.3 Port C and D

Port C pins PC[7:0] and port D pins PD[7:0] can be used for either general-purpose I/O, or, in 144-pin packages, also with the external bus interface. In this case port C and port D are associated with the external data bus inputs/outputs DATA15–DATA8 and DATA7–DATA0, respectively.

These pins are configured for reduced input threshold in certain operating modes (refer to S12X_EBI section).

NOTE

Port C and D are neither available in 112-pin nor in 80-pin packages.

22.4.2.4 Port E

Port E is associated with the external bus control outputs R/$\overline{W}$, $\overline{LSTRB}$, $\overline{LDS}$ and $\overline{RE}$, the free-running clock outputs ECLK and ECLK2X, as well as with the $\overline{TAGHI}$, $\overline{TAGLO}$, MODA and MODB and interrupt inputs $\overline{IRQ}$ and $\overline{XIRQ}$.

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 $\overline{IRQ}$ interrupt input. $\overline{IRQ}$ will be enabled by setting the IRQEN configuration bit (Section 22.3.2.14, “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 $\overline{XIRQ}$ interrupt input. $\overline{XIRQ}$ 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.

Port E pins PE[5] and PE[6] are configured for reduced input threshold in certain modes (refer to S12X_EBI section).

22.4.2.5 Port K

Port K pins PK[7:0] can be used for either general-purpose I/O, or, in 144-pin packages, also with the external bus interface. In this case port K pins PK[6:0] are associated with the external address bus outputs ADDR22–ADDR16 and PK7 is associated to the $\overline{EWAIT}$ input.

Port K pin PE[7] is configured for reduced input threshold in certain modes (refer to S12X_EBI section).

NOTE

Port K is not available in 80-pin packages. PK[6] is not available in 112-pin packages.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
PTJ	R	PTJ7	PTJ6	PTJ5	PTJ4	0	PTJ2	PTJ1	PTJ0
	W								
PTIJ	R	PTIJ7	PTIJ6	PTIJ5	PTIJ4	0	PTIJ2	PTIJ1	PTIJ0
	W								
DDRJ	R	DDRJ7	DDRJ6	DDRJ5	DDRJ4	0	DDRJ2	DDRJ1	DDRJ0
	W								
RDRJ	R	RDRJ7	RDRJ6	RDRJ5	RDRJ4	0	RDRJ2	RDRJ1	RDRJ0
	W								
PERJ	R	PERJ7	PERJ6	PERJ5	PERJ4	0	PERJ2	PERJ1	PERJ0
	W								
PPSJ	R	PPSJ7	PPSJ6	PPSJ5	PPSJ4	0	PPSJ2	PPSJ1	PPSJ0
	W								
PIEJ	R	PIEJ7	PIEJ6	PIEJ5	PIEJ4	0	PIEJ2	PIEJ1	PIEJ0
	W								
PIFJ	R	PPSJ7	PPSJ6	PPSJ5	PPSJ4	0	PPSJ2	PPSJ1	PPSJ0
	W								
Reserved	R	0	0	0	0	0	0	0	0
	W								
PT1AD0	R	PT1AD07	PT1AD06	PT1AD05	PT1AD04	PT1AD03	PT1AD02	PT1AD01	PT1AD00
	W								
Reserved	R	0	0	0	0	0	0	0	0
	W								
DDR1AD0	R	DDR1AD07	DDR1AD06	DDR1AD05	DDR1AD04	DDR1AD03	DDR1AD02	DDR1AD01	DDR1AD00
	W								
Reserved	R	0	0	0	0	0	0	0	0
	W								
RDR1AD0	R	RDR1AD07	RDR1AD06	RDR1AD05	RDR1AD04	RDR1AD03	RDR1AD02	RDR1AD01	RDR1AD00
	W								
Reserved	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 23-2. PIM Register Summary (Sheet 6 of 7)

Table 24-13. Free-Running ECLK Clock Rate

EDIV[1:0]	Rate of Free-Running ECLK
00	ECLK = Bus clock rate
01	ECLK = Bus clock rate divided by 2
10	ECLK = Bus clock rate divided by 3
11	ECLK = Bus clock rate divided by 4

24.0.5.10 IRQ Control Register (IRQCR)

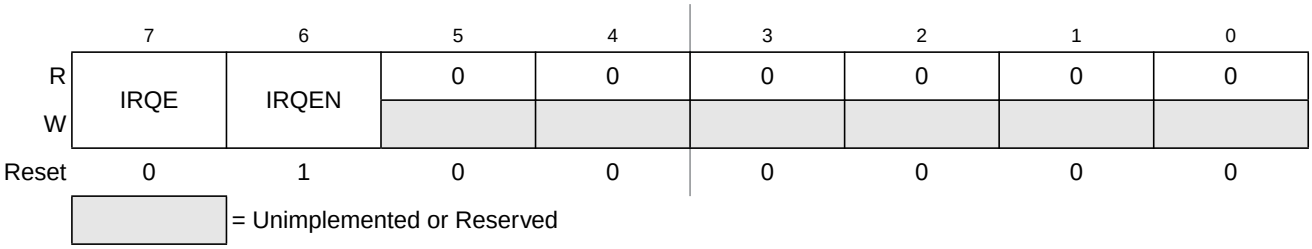

Figure 24-12. IRQ Control Register (IRQCR)

Read: See individual bit descriptions below.

Write: See individual bit descriptions below.

Table 24-14. IRQCR Field Descriptions

Field	Description
7 IRQE	**IRQ Select Edge Sensitive Only** Special modes: Read or write anytime. Normal and emulation modes: Read anytime, write once. 0 $\overline{\text{IRQ}}$ configured for low level recognition. 1 $\overline{\text{IRQ}}$ configured to respond only to falling edges. Falling edges on the $\overline{\text{IRQ}}$ pin will be detected anytime IRQE = 1 and will be cleared only upon a reset or the servicing of the $\overline{\text{IRQ}}$ interrupt.
6 IRQEN	**External IRQ Enable** Read or write anytime. 0 External $\overline{\text{IRQ}}$ pin is disconnected from interrupt logic. 1 External $\overline{\text{IRQ}}$ pin is connected to interrupt logic.

24.0.5.11 Port K Data Register (PORTK)

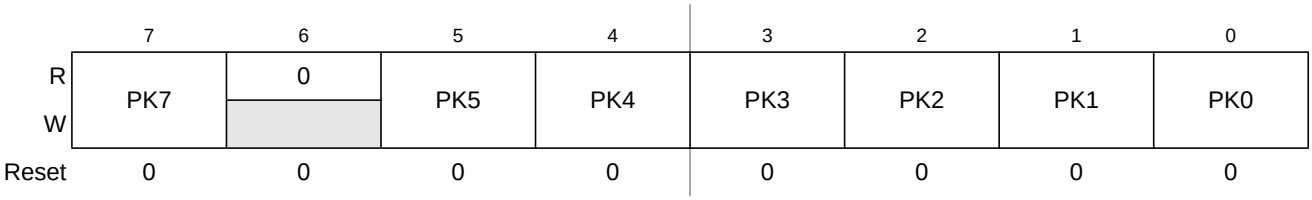

Figure 24-13. Port K Data Register (PORTK)

Read: Anytime.

Write: Anytime.

26.3.2.5 EEPROM Protection Register (EPROT)

The EPROT register defines which EEPROM sectors are protected against program or erase operations.

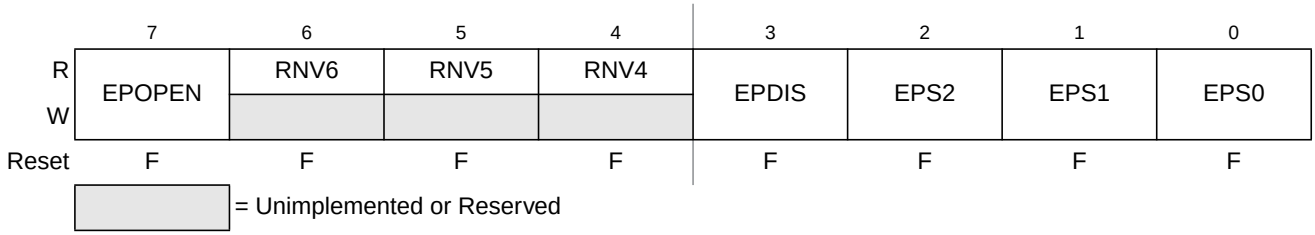

Figure 26-8. EEPROM Protection Register (EPROT)

During the reset sequence, the EPROT register is loaded from the EEPROM Protection byte at address offset 0x0FFD (see Table 26-1).All bits in the EPROT register are readable and writable except for RNV[6:4] which are only readable. The EPOPEN and EPDIS bits can only be written to the protected state. The EPS bits can be written anytime until bit EPDIS is cleared. If the EPOPEN bit is cleared, the state of the EPDIS and EPS bits is irrelevant.

To change the EEPROM protection that will be loaded during the reset sequence, the EEPROM memory must be unprotected, then the EEPROM Protection byte must be reprogrammed. Trying to alter data in any protected area in the EEPROM memory will result in a protection violation error and the PVIOL flag will be set in the ESTAT register. The mass erase of an EEPROM block is possible only when protection is fully disabled by setting the EPOPEN and EPDIS bits.

Table 26-5. EPROT Field Descriptions

Field	Description
7 EPOPEN	**Opens the EEPROM for Program or Erase** 0 The entire EEPROM memory is protected from program and erase. 1 The EEPROM sectors not protected are enabled for program or erase.
6–4 RNV[6:4]	**Reserved Nonvolatile Bits** — The RNV[6:4] bits should remain in the erased state “1” for future enhancements.
3 EPDIS	**EEPROM Protection Address Range Disable** — The EPDIS bit determines whether there is a protected area in a specific region of the EEPROM memory ending with address offset 0x0FFF. 0 Protection enabled. 1 Protection disabled.
2–0 EPS[2:0]	**EEPROM Protection Address Size** — The EPS[2:0] bits determine the size of the protected area as shown inTable 26-6. The EPS bits can only be written to while the EPDIS bit is set.

Table 26-6. EEPROM Protection Address Range

EPS[2:0]	Address Offset Range	Protected Size
000	0x0FC0 – 0x0FFF	64 bytes
001	0x0F80 – 0x0FFF	128 bytes
010	0x0F40 – 0x0FFF	192 bytes
011	0x0F00 – 0x0FFF	256 bytes
100	0x0EC0 – 0x0FFF	320 bytes
101	0x0E80 – 0x0FFF	384 bytes
110	0x0E40 – 0x0FFF	448 bytes
111	0x0E00 – 0x0FFF	512 bytes

26.3.2.6 EEPROM Status Register (ESTAT)

The ESTAT register defines the operational status of the module.

	7	6	5	4	3	2	1	0
R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	0	0
W								
Reset	1	1	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 26-9. EEPROM Status Register (ESTAT — Normal Mode)

	7	6	5	4	3	2	1	0
R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	FAIL	0
W								
Reset	1	1	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 26-10. EEPROM Status Register (ESTAT — Special Mode)

CBEIF, PVIOL, and ACCERR are readable and writable, CCIF and BLANK are readable and not writable, remaining bits read 0 and are not writable in normal mode. FAIL is readable and writable in special mode.

27.4.2.4 Sector Erase Command

The sector erase operation will erase all addresses in a 1 Kbyte sector of Flash memory using an embedded algorithm.

An example flow to execute the sector erase operation is shown in Figure 27-29. The sector erase command write sequence is as follows:

1. Write to a Flash block address to start the command write sequence for the sector erase command. The Flash address written determines the sector to be erased while global address bits [9:0] and the data written are ignored. Multiple Flash sectors can be simultaneously erased by writing to the same relative address in each Flash block.
2. Write the sector erase command, 0x40, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the sector erase command.

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

27.6.1 Unsecuring the MCU using Backdoor Key Access

The MCU may be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor keys (four 16-bit words programmed at addresses 0x7F_FF00–0x7F_FF07). If the KEYEN[1:0] bits are in the enabled state (see Section 27.3.2.2, “Flash Security Register (FSEC)”) and the KEYACC bit is set, a write to a backdoor key address in the Flash memory triggers a comparison between the written data and the backdoor key data stored in the Flash memory. If all four words of data are written to the correct addresses in the correct order and the data matches the backdoor keys stored in the Flash memory, the MCU will be unsecured. The data must be written to the backdoor keys sequentially starting with 0x7F_FF00–1 and ending with 0x7F_FF06–7. 0x0000 and 0xFFFF are not permitted as backdoor keys. While the KEYACC bit is set, reads of the Flash memory will return invalid data.

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

If the KEYEN[1:0] bits are in the enabled state (see Section 27.3.2.2, “Flash Security Register (FSEC)”), the MCU can be unsecured by the backdoor key access sequence described below:

1. Set the KEYACC bit in the Flash Configuration Register (FCNFG).
2. Write the correct four 16-bit words to Flash addresses 0xFF00–0xFF07 sequentially starting with 0x7F_FF00.
3. Clear the KEYACC bit. Depending on the user code used to write the backdoor keys, a wait cycle (NOP) may be required before clearing the KEYACC bit.
4. If all four 16-bit words match the backdoor keys stored in Flash addresses 0x7F_FF00–0x7F_FF07, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 1:0.

The backdoor key access sequence is monitored by an internal security state machine. An illegal operation during the backdoor key access sequence will cause the security state machine to lock, leaving the MCU in the secured state. A reset of the MCU will cause the security state machine to exit the lock state and allow a new backdoor key access sequence to be attempted. The following operations during the backdoor key access sequence will lock the security state machine:

1. If any of the four 16-bit words does not match the backdoor keys programmed in the Flash array.
2. If the four 16-bit words are written in the wrong sequence.
3. If more than four 16-bit words are written.
4. If any of the four 16-bit words written are 0x0000 or 0xFFFF.
5. If the KEYACC bit does not remain set while the four 16-bit words are written.
6. If any two of the four 16-bit words are written on successive MCU clock cycles.

After the backdoor keys have been correctly matched, the MCU will be unsecured. Once the MCU is unsecured, the Flash security byte can be programmed to the unsecure state, if desired.

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

The security as defined in the Flash security byte (0x7F_FF0F) is not changed by using the backdoor key access sequence to unsecure. The backdoor keys stored in addresses 0x7F_FF00–0x7F_FF07 are

29.3.2.7 Flash Command Register (FCMD)

The FCMD register is the Flash command register.

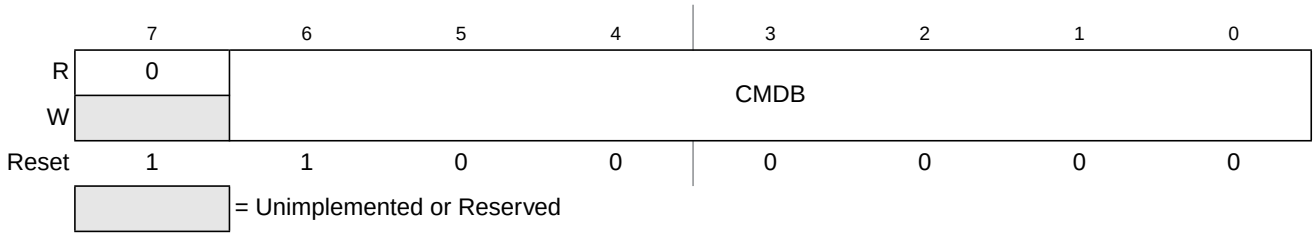

Figure 29-12. Flash Command Register (FCMD)

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.

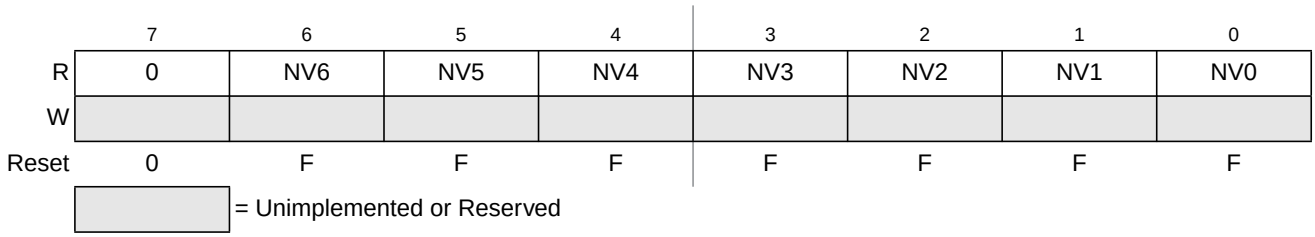

Figure 29-13. Flash Control Register (FCTL)

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.

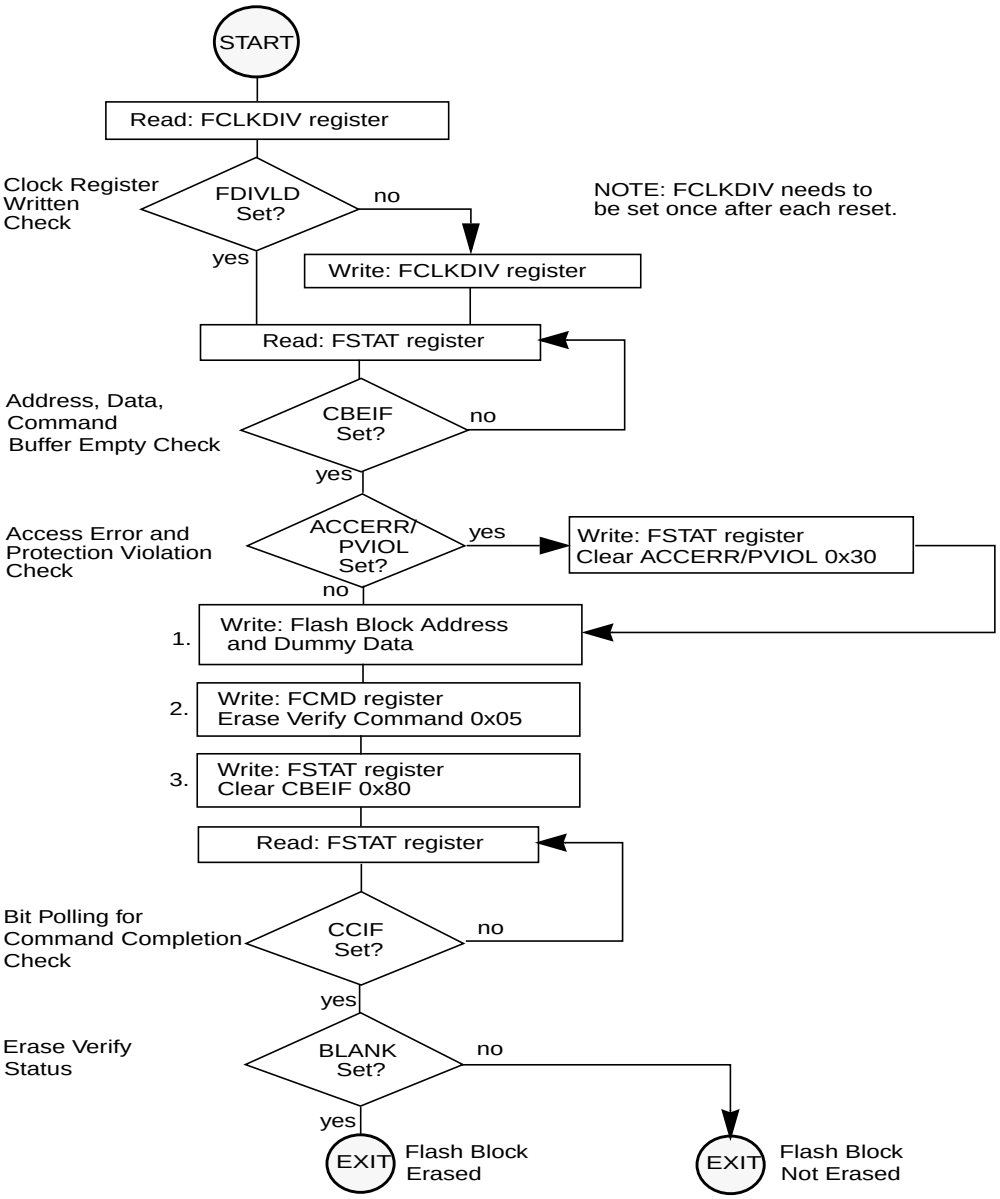

Figure 29-23. Example Erase Verify Command Flow