Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	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	91
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12xdp512val

5.3.2.6 ATD Control Register 5 (ATDCTL5)

This register selects the type of conversion sequence and the analog input channels sampled. Writes to this register will abort current conversion sequence and start a new conversion sequence.

	7	6	5	4	3	2	1	0
R	DJM	DSGN	SCAN	MULT	0	CC	CB	CA
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 5-8. ATD Control Register 5 (ATDCTL5)

Read: Anytime

Write: Anytime

Table 5-13. ATDCTL5 Field Descriptions

Field	Description
7 DJM	**Result Register Data Justification** — This bit controls justification of conversion data in the result registers. See Section 5.3.2.13, “ATD Conversion Result Registers (ATDDRx),” for details. 0 Left justified data in the result registers 1 Right justified data in the result registers
6 DSGN	**Result Register Data Signed or Unsigned Representation** — This bit selects between signed and unsigned conversion data representation in the result registers. Signed data is represented as 2’s complement. Signed data is not available in right justification. See Section 5.3.2.13, “ATD Conversion Result Registers (ATDDRx),” for details. 0 Unsigned data representation in the result registers 1 Signed data representation in the result registers Table 5-14 summarizes the result data formats available and how they are set up using the control bits. Table 5-15 illustrates the difference between the signed and unsigned, left justified output codes for an input signal range between 0 and 5.12 Volts.
5 SCAN	**Continuous Conversion Sequence Mode** — This bit selects whether conversion sequences are performed continuously or only once. 0 Single conversion sequence 1 Continuous conversion sequences (scan mode)
4 MULT	**Multi-Channel Sample Mode** — When MULT is 0, the ATD sequence controller samples only from the specified analog input channel for an entire conversion sequence. The analog channel is selected by channel selection code (control bits CC/CB/CA located in ATDCTL5). When MULT is 1, the ATD sequence controller samples across channels. The number of channels sampled is determined by the sequence length value (S8C, S4C, S2C, S1C). The first analog channel examined is determined by channel selection code (CC, CB, CA control bits); subsequent channels sampled in the sequence are determined by incrementing the channel selection code. 0 Sample only one channel 1 Sample across several channels
2–0 CC, CB, CA	**Analog Input Channel Select Code** — These bits select the analog input channel(s) whose signals are sampled and converted to digital codes. Table 5-16 lists the coding used to select the various analog input channels. In the case of single channel scans (MULT = 0), this selection code specified the channel examined. In the case of multi-channel scans (MULT = 1), this selection code represents the first channel to be examined in the conversion sequence. Subsequent channels are determined by incrementing channel selection code; selection codes that reach the maximum value wrap around to the minimum value.

BHI Branch if Higher BHI

Operation

If C | Z = 0, then PC + $0002 + (REL9 << 1) ⇒ PC

Branch instruction to compare unsigned numbers.

Branch if RS1 > RS2:

```
SUB     R0,RS1,RS2
BHI     REL9
```

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
BHI REL9	REL9	0	0	1	1	0	0	0	REL9	PP/P

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

Table 6-17. Instruction Set Summary (Sheet 3 of 3)

Functionality	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANDH RD, #IMM8	1	0	0	0	1	RD			IMM8							
BITL RD, #IMM8	1	0	0	1	0	RD			IMM8							
BITH RD, #IMM8	1	0	0	1	1	RD			IMM8							
ORL RD, #IMM8	1	0	1	0	0	RD			IMM8							
ORH RD, #IMM8	1	0	1	0	1	RD			IMM8							
XNORL RD, #IMM8	1	0	1	1	0	RD			IMM8							
XNORH RD, #IMM8	1	0	1	1	1	RD			IMM8							
Arithmetic Immediate Instructions																
SUBL RD, #IMM8	1	1	0	0	0	RD			IMM8							
SUBH RD, #IMM8	1	1	0	0	1	RD			IMM8							
CMPL RS, #IMM8	1	1	0	1	0	RS			IMM8							
CPCH RS, #IMM8	1	1	0	1	1	RS			IMM8							
ADDL RD, #IMM8	1	1	1	0	0	RD			IMM8							
ADDH RD, #IMM8	1	1	1	0	1	RD			IMM8							
LDL RD, #IMM8	1	1	1	1	0	RD			IMM8							
LDH RD, #IMM8	1	1	1	1	1	RD			IMM8							

14.2 External Signal Description

Due to the nature of VREG_3V3 being a voltage regulator providing the chip internal power supply voltages, most signals are power supply signals connected to pads.

Table 14-1 shows all signals of VREG_3V3 associated with pins.

Table 14-1. Signal Properties

Name	Function	Reset State	Pull Up
V_{DDR}	Power input (positive supply)	—	—
V_{DDA}	Quiet input (positive supply)	—	—
V_{SSA}	Quiet input (ground)	—	—
V_{DD}	Primary output (positive supply)	—	—
V_{SS}	Primary output (ground)	—	—
V_{DDPLL}	Secondary output (positive supply)	—	—
V_{SSPLL}	Secondary output (ground)	—	—
V_{REGEN} (optional)	Optional Regulator Enable	—	—

NOTE

Check device level specification for connectivity of the signals.

14.2.1 VDDR — Regulator Power Input Pins

Signal V_{DDR} is the power input of VREG_3V3. All currents sourced into the regulator loads flow through this pin. A chip external decoupling capacitor (>=100 nF, X7R ceramic) between V_{DDR} and V_{SSR} (if V_{SSR} is not available V_{SS}) can smooth ripple on V_{DDR}.

For entering Shutdown Mode, pin V_{DDR} should also be tied to ground on devices without VREGEN pin.

14.2.2 VDDA, VSSA — Regulator Reference Supply Pins

Signals $V_{DDA}/V_{SSA,}$ which are supposed to be relatively quiet, are used to supply the analog parts of the regulator. Internal precision reference circuits are supplied from these signals. A chip external decoupling capacitor (>=100 nF, X7R ceramic) between V_{DDA} and V_{SSA} can further improve the quality of this supply.

14.2.3 VDD, VSS — Regulator Output1 (Core Logic) Pins

Signals V_{DD}/V_{SS} are the primary outputs of VREG_3V3 that provide the power supply for the core logic. These signals are connected to device pins to allow external decoupling capacitors (220 nF, X7R ceramic).

In Shutdown Mode an external supply driving V_{DD}/V_{SS} can replace the voltage regulator.

Figure 15-10 shows the host receiving a logic 0 from the target. Since the host is asynchronous to the target, there is up to a one clock-cycle delay from the host-generated falling edge on BKGD to the start of the bit time as perceived by the target. The host initiates the bit time but the target finishes it. Since the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 target clock cycles then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 target clock cycles after starting the bit time.

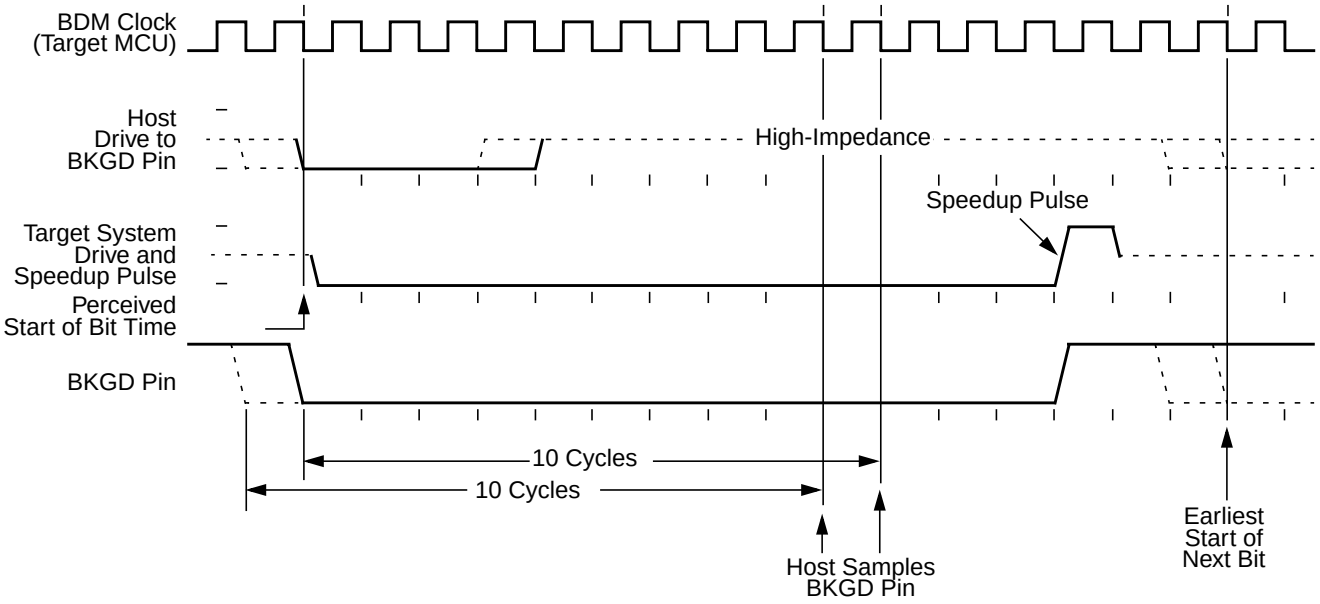

Figure 15-10. BDM Target-to-Host Serial Bit Timing (Logic 0)

15.4.7 Serial Interface Hardware Handshake Protocol

BDM commands that require CPU execution are ultimately treated at the MCU bus rate. Since the BDM clock source can be asynchronously related to the bus frequency, when CLKSW = 0, it is very helpful to provide a handshake protocol in which the host could determine when an issued command is executed by the CPU. The alternative is to always wait the amount of time equal to the appropriate number of cycles at the slowest possible rate the clock could be running. This sub-section will describe the hardware handshake protocol.

The hardware handshake protocol signals to the host controller when an issued command was successfully executed by the target. This protocol is implemented by a 16 serial clock cycle low pulse followed by a brief speedup pulse in the BKGD pin. This pulse is generated by the target MCU when a command, issued by the host, has been successfully executed (see Figure 15-11). This pulse is referred to as the ACK pulse. After the ACK pulse has finished: the host can start the bit retrieval if the last issued command was a read command, or start a new command if the last command was a write command or a control command (BACKGROUND, GO, GO_UNTIL or TRACE1). The ACK pulse is not issued earlier than 32 serial clock cycles after the BDM command was issued. The end of the BDM command is assumed to be the 16th tick of the last bit. This minimum delay assures enough time for the host to perceive the ACK pulse. Note also that, there is no upper limit for the delay between the command and the related ACK pulse, since the command execution depends upon the CPU bus frequency, which in some cases could be very slow

19.3.1.11.6 Debug Comparator Data Low Register (DBGXDL)

0x002D

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

Figure 19-19. Debug Comparator Data Low Register (DBGXDL)

Read: Anytime

Write: Anytime when DBG not armed.

Table 19-33. DBGXDL Field Descriptions

Field	Description
7–0 Bits [7:0]	**Comparator Data Low Compare Bits** — The comparator data low compare bits control whether the selected comparator compares the data bus bits [7:0] to a logic 1 or logic 0. The comparator data compare bits are only used in comparison if the corresponding data mask bit is logic 1. This register is available only for comparators A and C. 0 Compare corresponding data bit to a logic 0 1 Compare corresponding data bit to a logic 1

19.3.1.11.7 Debug Comparator Data High Mask Register (DBGXDHM)

0x002E

	7	6	5	4	3	2	1	0
R W	Bit 15	14	13	12	11	10	9	Bit 8
Reset	0	0	0	0	0	0	0	0

Figure 19-20. Debug Comparator Data High Mask Register (DBGXDHM)

Read: Anytime

Write: Anytime when DBG not armed.

Table 19-34. DBGXDHM Field Descriptions

Field	Description
7–0 Bits [15:8]	**Comparator Data High Mask Bits** — The comparator data high mask bits control whether the selected comparator compares the data bus bits [15:8] to the corresponding comparator data compare bits. This register is available only for comparators A and C. 0 Do not compare corresponding data bit 1 Compare corresponding data bit

22.2 External Signal Description

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

22.2.1 Signal Properties

Table 22-1 shows all the pins and their functions that are controlled by the PIM. *Refer to Section 22.4, "Functional Description" for the availability of the individual pins in the different package options.*

NOTE

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 bottom (lowest priority).

Table 22-1. Pin Functions and Priorities (Sheet 1 of 7)

Port	Pin Name	Pin Function and Priority	I/O	Description	Pin Function after Reset
—	BKGD	MODC[1]	I	MODC input during $\overline{\text{RESET}}$	BKGD
		BKGD	I/O	S12X_BDM communication pin	
A	PA[7:0]	ADDR[15:8] mux IVD[15:8][2]	O	High-order external bus address output (multiplexed with IVIS data)	Mode dependent[3]
		GPIO	I/O	General-purpose I/O	
B	PB[7:1]	ADDR[7:1] mux IVD[7:1][2]	O	Low-order external bus address output (multiplexed with IVIS data)	Mode dependent[3]
		GPIO	I/O	General-purpose I/O	
	PB[0]	ADDR[0] mux IVD0[2]	O	Low-order external bus address output (multiplexed with IVIS data)	
		$\overline{\text{UDS}}$	O	Upper data strobe	
		GPIO	I/O	General-purpose I/O	
C	PC[7:0]	DATA[15:8]	I/O	High-order bidirectional data input/output Configurable for reduced input threshold	Mode dependent[3]
		GPIO	I/O	General-purpose I/O	
D	PD[7:0]	DATA[7:0]	I/O	Low-order bidirectional data input/output Configurable for reduced input threshold	Mode dependent[3]
		GPIO	I/O	General-purpose I/O	

Table 22-1. Pin Functions and Priorities (Sheet 2 of 7)

Port	Pin Name	Pin Function and Priority	I/O	Description	Pin Function after Reset
E	PE[7]	$\overline{\text{XCLKS}}$[1]	I	External clock selection input during $\overline{\text{RESET}}$	Mode dependent[3]
		ECLKX2	I	Free-running clock output at Core Clock rate (ECLK x 2)	
		GPIO	I/O	General-purpose I/O	
	PE[6]	MODB[1]	I	MODB input during $\overline{\text{RESET}}$	
		$\overline{\text{TAGHI}}$	I	Instruction tagging low pin Configurable for reduced input threshold	
		GPIO	I/O	General-purpose I/O	
	PE[5]	MODA[1]	I	MODA input during $\overline{\text{RESET}}$	
		$\overline{\text{RE}}$	O	Read enable signal	
		$\overline{\text{TAGLO}}$	I	Instruction tagging low pin Configurable for reduced input threshold	
		GPIO	I/O	General-purpose I/O	
	PE[4]	ECLK	O	Free-running clock output at the Bus Clock rate or programmable divided in normal modes	
		GPIO	I/O	General-purpose I/O	
	PE[3]	EROMCTL[1]	I	EROMON bit control input during $\overline{\text{RESET}}$	
		$\overline{\text{LSTRB}}$	O	Low strobe bar output	
		$\overline{\text{LDS}}$	O	Lower data strobe	
		GPIO	I/O	General-purpose I/O	
	PE[2]	R/$\overline{\text{W}}$	O	Read/write output for external bus	
		$\overline{\text{WE}}$	O	Write enable signal	
		GPIO	I/O	General-purpose I/O	
	PE[1]	$\overline{\text{IRQ}}$	I	Maskable level- or falling edge-sensitive interrupt input	
		GPIO	I/O	General-purpose I/O	
	PE[0]	$\overline{\text{XIRQ}}$	I	Non-maskable level-sensitive interrupt input	
		GPIO	I/O	General-purpose I/O	

22.3.2.12 S12X_EBI Ports Reduced Drive Register (RDRIV)

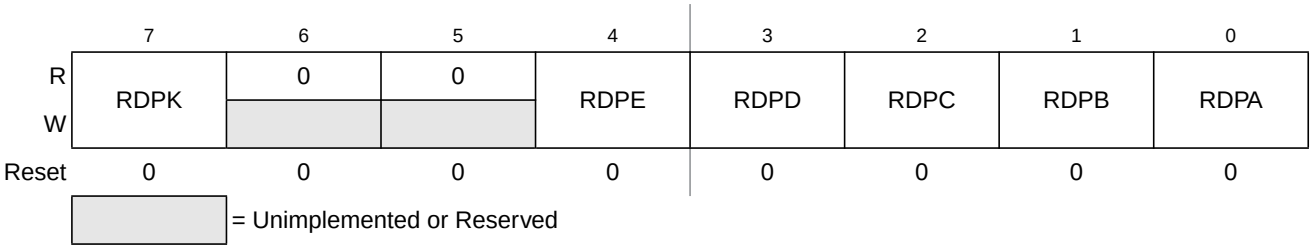

Figure 22-14. S12X_EBI Ports Reduced Drive Register (RDRIV)

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.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
PTIP	R	PTIP7	PTIP6	PTIP5	PTIP4	PTIP3	PTIP2	PTIP1	PTIP0
	W								
DDRP	R	DDRP7	DDRP6	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0
	W								
RDRP	R	RDRP7	RDRP6	RDRP5	RDRP4	RDRP3	RDRP2	RDRP1	RDRP0
	W								
PERP	R	PERP7	PERP6	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0
	W								
PPSP	R	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
	W								
PIEP	R	PIEP7	PIEP6	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
	W								
PIFP	R	PIFP7	PIFP6	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
	W								
PTH	R	PTH7	PTH6	PTH5	PTH4	PTH3	PTH2	PTH1	PTH0
	W								
PTIH	R	PTIH7	PTIH6	PTIH5	PTIH4	PTIH3	PTIH2	PTIH1	PTIH0
	W								
DDRH	R	DDRH7	DDRH6	DDRH5	DDRH4	DDRH3	DDRH2	DDRH1	DDRH0
	W								
RDRH	R	RDRH7	RDRH6	RDRH5	RDRH4	RDRH3	RDRH2	RDRH1	RDRH0
	W								
PERH	R	PERH7	PERH6	PERH5	PERH4	PERH3	PERH2	PERH1	PERH0
	W								
PPSH	R	PPSH7	PPSH6	PPSH5	PPSH4	PPSH3	PPSH2	PPSH1	PPSH0
	W								
PIEH	R	PIEH7	PIEH6	PIEH5	PIEH4	PIEH3	PIEH2	PIEH1	PIEH0
	W								
PIFH	R	PIFH7	PIFH6	PIFH5	PIFH4	PIFH3	PIFH2	PIFH1	PIFH0
	W								

= Unimplemented or Reserved

Figure 24-2. PIM Register Summary (Sheet 5 of 7)

24.0.5.36 Port P Data Direction Register (DDRP)

	7	6	5	4	3	2	1	0
R W	DDRP7	DDRP6	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0
Reset	0	0	0	0	0	0	0	0

Figure 24-38. Port P Data Direction Register (DDRP)

Read: Anytime.

Write: Anytime.

This register configures each port P pin as either input or output.

If the associated PWM channel or SPI module is enabled this register has no effect on the pins.

The PWM forces the I/O state to be an output for each port line associated with an enabled PWM7–0 channel. Channel 7 can force the pin to input if the shutdown feature is enabled. *Refer to PWM section for details.*

If SPI is enabled, the SPI determines the pin direction. *Refer to SPI section for details.*

The DDRP bits revert to controlling the I/O direction of a pin when the associated peripherals are disabled.

Table 24-35. DDRP Field Descriptions

Field	Description
7–0 DDRP[7:0]	**Data Direction Port P** 0 Associated pin is configured as input. 1 Associated pin is configured as output. **Note:** Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTP or PTIP registers, when changing the DDRP register.

24.0.5.37 Port P Reduced Drive Register (RDRP)

	7	6	5	4	3	2	1	0
R W	RDRP7	RDRP6	RDRP5	RDRP4	RDRP3	RDRP2	RDRP1	RDRP0
Reset	0	0	0	0	0	0	0	0

Figure 24-39. Port P Reduced Drive Register (RDRP)

Read: Anytime.

Write: Anytime.

This register configures the drive strength of each port P output pin as either full or reduced. If the port is used as input this bit is ignored.

The EADDRHI and EADDRLO registers are the EEPROM address registers.

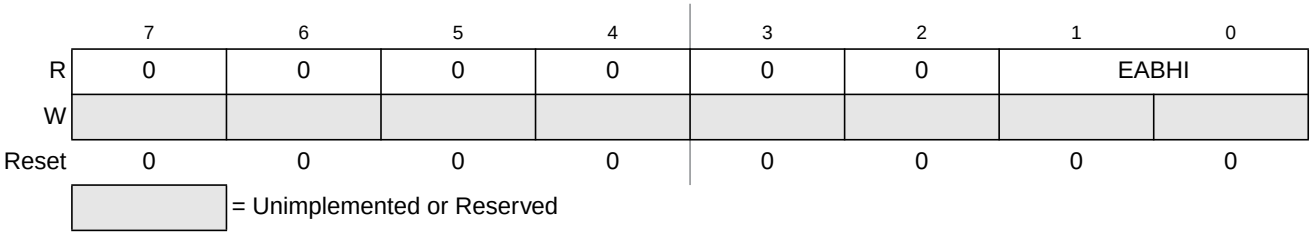

Figure 25-13. EEPROM Address High Register (EADDRHI)

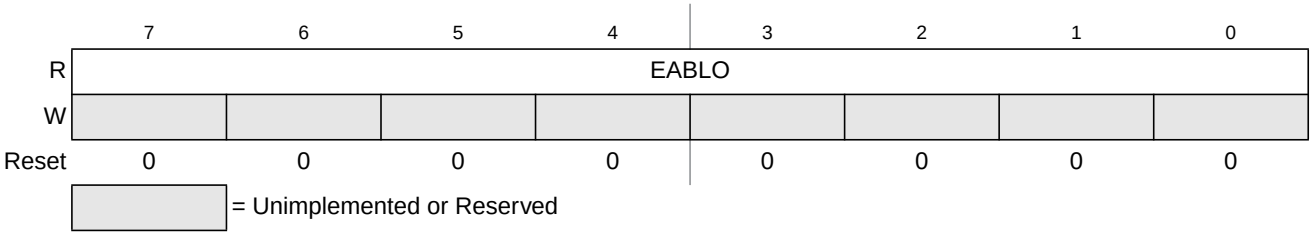

Figure 25-14. EEPROM Address Low Register (EADDRLO)

All EABHI and EABLO bits read 0 and are not writable in normal modes.

All EABHI and EABLO bits are readable and writable in special modes.

The MCU address bit AB0 is not stored in the EADDR registers since the EEPROM block is not byte addressable.

25.3.2.9 EEPROM Data Registers (EDATA)

The EDATAHI and EDATALO registers are the EEPROM data registers.

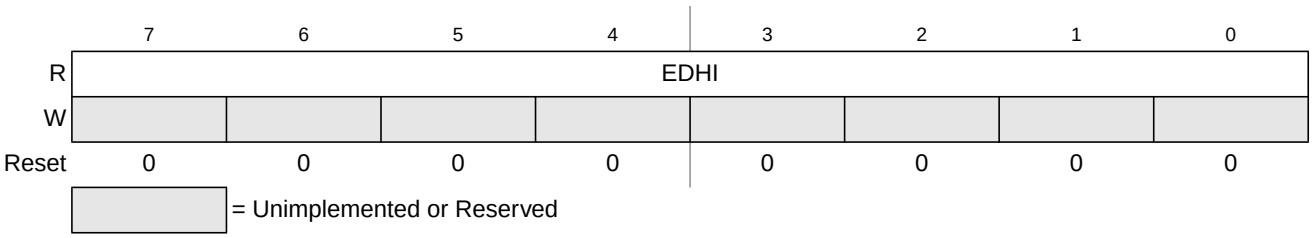

Figure 25-15. EEPROM Data High Register (EDATAHI)

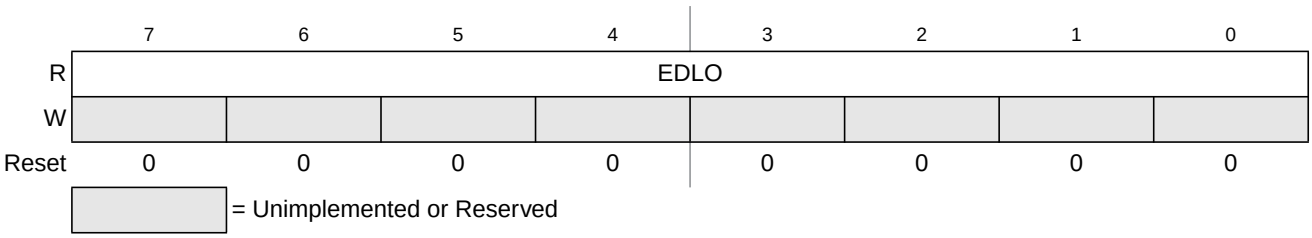

Figure 25-16. EEPROM Data Low Register (EDATALO)

All EDHI and EDLO bits read 0 and are not writable in normal modes.

26.1.4 Block Diagram

A block diagram of the EEPROM module is shown in Figure 26-1.

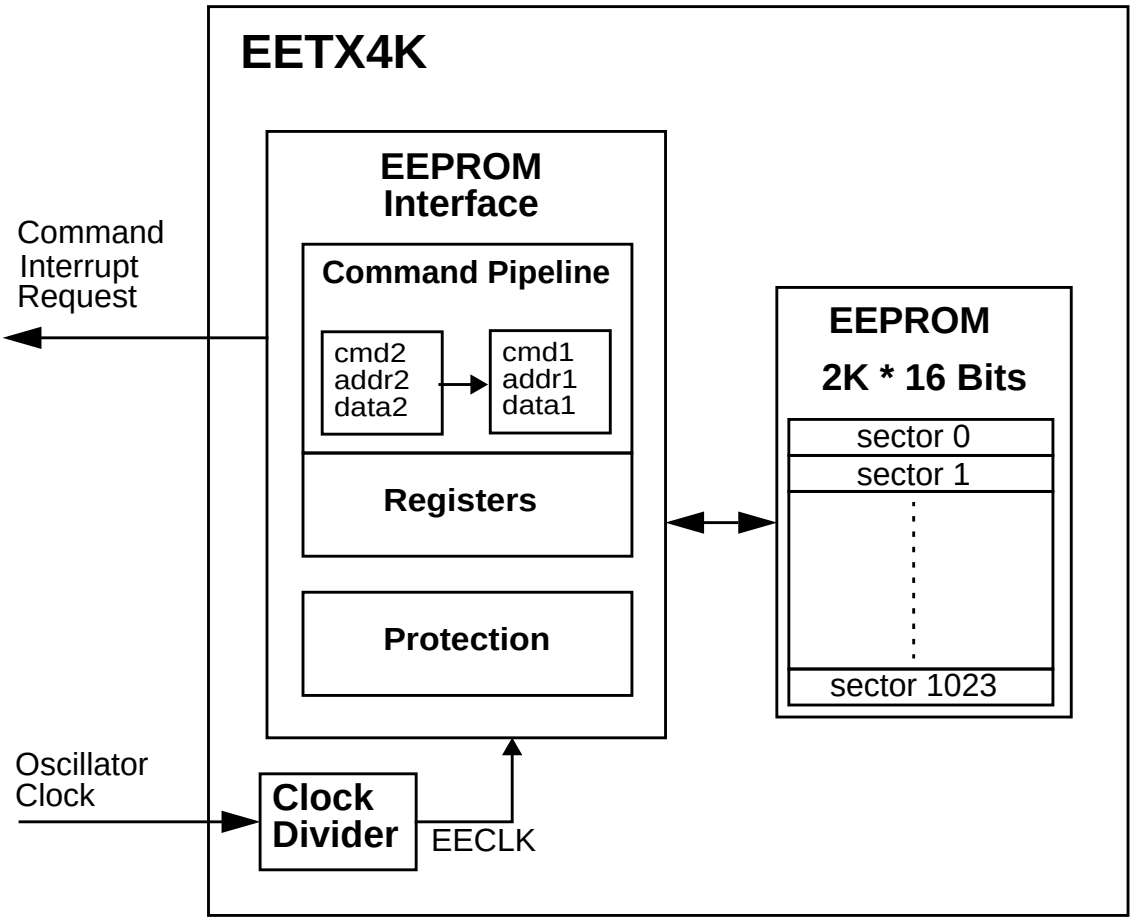

Figure 26-1. EETX4K Block Diagram

26.2 External Signal Description

The EEPROM module contains no signals that connect off-chip.

26.3 Memory Map and Register Definition

This section describes the memory map and registers for the EEPROM module.

26.3.1 Module Memory Map

The EEPROM memory map is shown in Figure 26-2. The HCS12X architecture places the EEPROM memory addresses between global addresses 0x13_F000 and 0x13_FFFF. The EPROT register, described in Section 26.3.2.5, “EEPROM Protection Register (EPROT)”, can be set to protect the upper region in the EEPROM memory from accidental program or erase. The EEPROM addresses covered by this protectable

27.4.2.2 Data Compress Command

The data compress operation will check Flash code integrity by compressing data from a selected portion of the Flash memory into a signature analyzer.

An example flow to execute the data compress operation is shown in Figure 27-26. The data compress command write sequence is as follows:

1. Write to a Flash block address to start the command write sequence for the data compress command. The address written determines the starting address for the data compress operation and the data written determines the number of consecutive words to compress. If the data value written is 0x0000, 64K addresses or 128 Kbytes will be compressed. Multiple Flash blocks can be simultaneously compressed by writing to the same relative address in each Flash block. If more than one Flash block is written to in this step, the first data written will determine the number of consecutive words to compress in each selected Flash block.
2. Write the data compress command, 0x06, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the data compress command.

After launching the data compress command, the CCIF flag in the FSTAT register will set after the data compress operation has completed. The number of bus cycles required to execute the data compress operation is equal to two times the number of consecutive words to compress plus the number of Flash blocks simultaneously compressed plus 18 bus cycles as measured from the time the CBEIF flag is cleared until the CCIF flag is set. Once the CCIF flag is set, the signature generated by the data compress operation is available in the FDATA registers. The signature in the FDATA registers can be compared to the expected signature to determine the integrity of the selected data stored in the selected Flash memory. If the last address of a Flash block is reached during the data compress operation, data compression will continue with the starting address of the same Flash block. The MRDS bits in the FTSTMOD register will determine the sense-amp margin setting during the data compress operation.

NOTE

Since the FDATA registers (or data buffer) are written to as part of the data compress operation, a command write sequence is not allowed to be buffered behind a data compress command write sequence. The CBEIF flag will not set after launching the data compress command to indicate that a command should not be buffered behind it. If an attempt is made to start a new command write sequence with a data compress operation active, the ACCERR flag in the FSTAT register will be set. A new command write sequence should only be started after reading the signature stored in the FDATA registers.

In order to take corrective action, it is recommended that the data compress command be executed on a Flash sector or subset of a Flash sector. If the data compress operation on a Flash sector returns an invalid signature, the Flash sector should be erased using the sector erase command and then reprogrammed using the program command.

The data compress command can be used to verify that a sector or sequential set of sectors are erased.

28.4.2.3 Program Command

The program operation will program a previously erased word in the Flash memory using an embedded algorithm.

An example flow to execute the program operation is shown in Figure 28-28. The program command write sequence is as follows:

1. Write to a Flash block address to start the command write sequence for the program command. The data written will be programmed to the address written. Multiple Flash blocks can be simultaneously programmed by writing to the same relative address in each Flash block.
2. Write the program command, 0x20, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the program command.

If a word to be programmed is in a protected area of the Flash block, the PVIOL flag in the FSTAT register will set and the program command will not launch. Once the program command has successfully launched, the CCIF flag in the FSTAT register will set after the program operation has completed unless a new command write sequence has been buffered. By executing a new program command write sequence on sequential words after the CBEIF flag in the FSTAT register has been set, up to 55% faster programming time per word can be effectively achieved than by waiting for the CCIF flag to set after each program operation.

29.4.2.4 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 29-27. 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.
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.

A.3 NVM, Flash, and EEPROM

NOTE

Unless otherwise noted the abbreviation NVM (nonvolatile memory) is used for both Flash and EEPROM.

A.3.1 NVM Timing

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

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

The minimum program and erase times shown in Table A-17 are calculated for maximum f_{NVMOP} and maximum f_{bus}. The maximum times are calculated for minimum f_{NVMOP} and a f_{bus} of 2 MHz.

A.3.1.1 Single Word Programming

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

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

A.3.1.2 Burst Programming

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

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

The time to program a whole row is:

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

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