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	40MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	44
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12xs256j0vaer

Chapter 1
Device Overview S12XS Family

1.1 Introduction

The new S12XS family of 16-bit micro controllers is a compatible, reduced version of the S12XE family. These families provide an easy approach to develop common platforms from low-end to high-end applications, minimizing the redesign of software and hardware.

Targeted at generic automotive applications and CAN nodes, some typical examples of these applications are: Body Controllers, Occupant Detection, Door Modules, RKE Receivers, Smart Actuators, Lighting Modules and Smart Junction Boxes amongst many others.

The S12XS family retains many of the features of the S12XE family including Error Correction Code (ECC) on Flash memory, a separate Data-Flash Module for code or data storage, a Frequency Modulated Locked Loop (IPLL) that improves the EMC performance and a fast ATD converter.

S12XS family delivers 32-bit performance with all the advantages and efficiencies of a 16-bit MCU while retaining the low cost, power consumption, EMC and code-size efficiency advantages currently enjoyed by users of Freescale's existing 16-bit S12 and S12X MCU families. Like members of other S12X families, the S12XS family runs 16-bit wide accesses without wait states for all peripherals and memories.

The S12XS family is available in 112-pin LQFP, 80-pin QFP, 64-pin LQFP package options and maintains a high level of pin compatibility with the S12XE family. In addition to the I/O ports available in each module, up to 18 further I/O ports are available with interrupt capability allowing Wake-Up from stop or wait modes.

The peripheral set includes MSCAN, SPI, two SCIs, an 8-channel 24-bit periodic interrupt timer, 8-channel 16-bit Timer, 8-channel PWM and up to 16- channel 12-bit ATD converter.

Software controlled peripheral-to-port routing enables access to a flexible mix of the peripheral modules in the lower pin count package options.

1.1.1 Features

Features of the S12XS Family are listed here. Please see Table D-1 for memory options and Table D-2 for the peripheral features that are available on the different family members.

- 16-bit CPU12X
  - Upward compatible with S12 instruction set with the exception of five Fuzzy instructions (MEM, WAV, WAVR, REV, REVW) which have been removed
  - Enhanced indexed addressing
  - Access to large data segments independent of PPAGE

The clock generated by the PLL or oscillator provides the main system clock frequencies core clock and bus clock. As shown in Figure 1-6, these system clocks are used throughout the MCU to drive the core, the memories, and the peripherals.

The program Flash memory is supplied by the bus clock and the oscillator clock. The oscillator clock is used as a time base to derive the program and erase times for the NVMs.

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.

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 Modes of Operation

The MCU can operate in different modes. These are described in 1.4.1 Chip Configuration Summary.

The MCU can operate in different power modes to facilitate power saving when full system performance is not required. These are described in 1.4.2 Power Modes.

Some modules feature a software programmable option to freeze the module status whilst the background debug module is active to facilitate debugging. This is described in 1.4.3 Freeze Mode.

1.4.1 Chip Configuration Summary

The different modes and the security state of the MCU affect the debug features (enabled or disabled).

The operating mode out of reset is determined by the state of the MODC signal during reset (see Table 1-8). The MODC bit in the MODE register shows the current operating mode and provides 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 1-8. Chip Modes

Chip Modes	MODC
Normal single chip	1
Special single chip	0

1.4.1.1 Normal Single-Chip Mode

This mode is intended for normal device operation. The opcode from the on-chip memory is being executed after reset (requires the reset vector to be programmed correctly). The processor program is executed from internal memory.

Table 2-2. Block Memory Map (continued)

Port	Offset or Address	Register	Access	Reset Value	Section/Page
T	0x0240	PTT—Port T Data Register	R/W	0x00	2.3.18/2-85
	0x0241	PTIT—Port T Input Register	R	[4]	2.3.19/2-86
	0x0242	DDRT—Port T Data Direction Register	R/W	0x00	2.3.20/2-87
	0x0243	RDRT—Port T Reduced Drive Register	R/W	0x00	2.3.21/2-87
	0x0244	PERT—Port T Pull Device Enable Register	R/W	0x00	2.3.22/2-88
	0x0245	PPST—Port T Polarity Select Register	R/W	0x00	2.3.23/2-88
	0x0246	PIM Reserved	R	0x00	2.3.24/2-89
	0x0247	Port T Routing Register	R/W	0x00	2.3.25/2-89
S	0x0248	PTS—Port S Data Register	R/W	0x00	2.3.26/2-91
	0x0249	PTIS—Port S Input Register	R	[4]	2.3.27/2-92
	0x024A	DDRS—Port S Data Direction Register	R/W	0x00	2.3.28/2-93
	0x024B	RDRS—Port S Reduced Drive Register	R/W	0x00	2.3.29/2-94
	0x024C	PERS—Port S Pull Device Enable Register	R/W	0xFF	2.3.30/2-94
	0x024D	PTPS—Port S Polarity Select Register	R/W	0x00	2.3.31/2-95
	0x024E	WOMS—Port S Wired-Or Mode Register	R/W	0x00	2.3.32/2-95
	0x024F	PIM Reserved	R	0x00	2.3.33/2-96
M	0x0250	PTM—Port M Data Register	R/W	0x00	2.3.34/2-96
	0x0251	PTIM—Port M Input Register	R	[4]	2.3.35/2-98
	0x0252	DDRM—Port M Data Direction Register	R/W	0x00	2.3.36/2-98
	0x0253	RDRM—Port M Reduced Drive Register	R/W	0x00	2.3.37/2-99
	0x0254	PERM—Port M Pull Device Enable Register	R/W	0x00	2.3.38/2-100
	0x0255	PPSM—Port M Polarity Select Register	R/W	0x00	2.3.39/2-100
	0x0256	WOMM—Port M Wired-Or Mode Register	R/W	0x00	2.3.40/2-101
	0x0257	MODRR—Module Routing Register	R/W	0x00	2.3.41/2-101
P	0x0258	PTP—Port P Data Register	R/W	0x00	2.3.42/2-102
	0x0259	PTIP—Port P Input Register	R	[4]	2.3.43/2-104
	0x025A	DDRP—Port P Data Direction Register	R/W	0x00	2.3.44/2-105
	0x025B	RDRP—Port P Reduced Drive Register	R/W	0x00	2.3.45/2-106
	0x025C	PERP—Port P Pull Device Enable Register	R/W	0x00	2.3.46/2-106
	0x025D	PPSP—Port P Polarity Select Register	R/W	0x00	2.3.47/2-107
	0x025E	PIEP—Port P Interrupt Enable Register	R/W	0x00	2.3.48/2-107
	0x025F	PIFP—Port P Interrupt Flag Register	R/W	0x00	2.3.49/2-108

Table 2-32. DDRM Register Field Descriptions

Field	Description
7-6 DDRM	**Port M data direction**— This bit determines whether the associated pin is an input or output. 1 Associated pin configured as output 0 Associated pin configured as input
5-2 DDRM	**Port M data direction**— This bit determines whether the associated pin is an input or output. Depending on the configuration of the enabled SPI0 the I/O state will be forced to be input or output. In this case the data direction bit will not change. 1 Associated pin configured as output 0 Associated pin configured as input
1 DDRM	**Port M data direction**— This bit determines whether the associated pin is an input or output. The enabled CAN0 or SCI1 forces the I/O state to be an output. In this case the data direction bit will not change. 1 Associated pin configured as output 0 Associated pin configured as input
0 DDRM	**Port M data direction**— This bit determines whether the associated pin is an input or output. The enabled CAN0 or SCI1 forces the I/O state to be an input. In this case the data direction bit will not change. 1 Associated pin configured as output 0 Associated pin configured as input

2.3.37 Port M Reduced Drive Register (RDRM)

Address 0x0253 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDRM7	RDRM6	RDRM5	RDRM4	RDRM3	RDRM2	RDRM1	RDRM0
Reset	0	0	0	0	0	0	0	0

Figure 2-35. Port M Reduced Drive Register (RDRM)

1 Read: Anytime
Write: Anytime

Table 2-33. RDRM Register Field Descriptions

Field	Description
7-0 RDRM	**Port M reduced drive**—Select reduced drive for output pin This bit configures the drive strength of the associated output pin as either full or reduced. If a pin is used as input this bit has no effect. The reduced drive function is independent of which function is being used on a particular pin. 1 Reduced drive selected (approx. 1/5 of the full drive strength) 0 Full drive strength enabled

NOTE

Using this configuration, a byte access of ADDR[n] can cause a comparator match if the databus low byte by chance contains the same value as ADDR[n+1] because the databus comparator does not feature access size comparison and uses the mask as a “don’t care” function. Thus masked bits do not prevent a match.

Comparators A and C feature an NDB control bit to determine if a match occurs when the data bus differs to comparator register contents or when the data bus is equivalent to the comparator register contents.

6.4.2.2 Exact Address Comparator Match (Comparators B and D)

Comparators B and D feature SZ and SZE control bits. If SZE is clear, then the comparator address match qualification functions the same as for comparators A and C.

If the SZE bit is set the access size (word or byte) is compared with the SZ bit value such that only the specified type of access causes a match. Thus if configured for a byte access of a particular address, a word access covering the same address does not lead to match.

Table 6-37. Comparator Access Size Considerations

Comparator	Address	SZE	SZ8	Condition For Valid Match
Comparators A and C	ADDR[n]	—	—	Word and byte accesses of ADDR[n](1) MOVB #$BYTE ADDR[n] MOVW #$WORD ADDR[n]
Comparators B and D	ADDR[n]	0	X	Word and byte accesses of ADDR[n]1 MOVB #$BYTE ADDR[n] MOVW #$WORD ADDR[n]
Comparators B and D	ADDR[n]	1	0	Word accesses of ADDR[n]1 MOVW #$WORD ADDR[n]
Comparators B and D	ADDR[n]	1	1	Byte accesses of ADDR[n] MOVB #$BYTE ADDR[n]

1. A word access of ADDR[n-1] also accesses ADDR[n] but does not generate a match.
The comparator address register must contain the exact address used in the code.

6.4.2.3 Data Bus Comparison NDB Dependency

Comparators A and C each feature an NDB control bit, which allows data bus comparators to be configured to either trigger on equivalence or trigger on difference. This allows monitoring of a difference in the contents of an address location from an expected value.

When matching on an equivalence (NDB=0), each individual data bus bit position can be masked out by clearing the corresponding mask bit (DBGxDHM/DBGxDLM), so that it is ignored in the comparison. A match occurs when all data bus bits with corresponding mask bits set are equivalent. If all mask register bits are clear, then a match is based on the address bus only, the data bus is ignored.

When matching on a difference, mask bits can be cleared to ignore bit positions. A match occurs when any data bus bit with corresponding mask bit set is different. Clearing all mask bits, causes all bits to be ignored and prevents a match because no difference can be detected. In this case address bus equivalence does not cause a match.

8.3.2.2 S12XECRG Reference Divider Register (REFDV)

The REFDV register provides a finer granularity for the IPLL multiplier steps.

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R W	REFFRQ[1:0]		REFDIV[5:0]					
Reset	0	0	0	0	0	0	0	0

Figure 8-4. S12XECRG Reference Divider Register (REFDV)

Read: Anytime

Write: Anytime except when PLLSEL = 1

NOTE

Write to this register initializes the lock detector bit.

$$f_{REF} = \frac{f_{OSC}}{(REFDIV + 1)}$$

The REFFRQ[1:0] bit are used to configure the internal PLL filter for optimal stability and lock time. For correct IPLL operation the REFFRQ[1:0] bits have to be selected according to the actual REFCLK frequency as shown in Figure 8-3. Setting the REFFRQ[1:0] bits wrong can result in a non functional IPLL (no locking and/or insufficient stability).

Table 8-3. Reference Clock Frequency Selection

REFCLK Frequency Ranges	REFFRQ[1:0]
1MHz <= f_{REF} <= 2MHz	00
2MHz < f_{REF} <= 6MHz	01
6MHz < f_{REF} <= 12MHz	10
f_{REF} >12MHz	11

8.3.2.3 S12XECRG Post Divider Register (POSTDIV)

The POSTDIV register controls the frequency ratio between the VCOCLK and PLLCLK. The count in the final divider divides VCOCLK frequency by 1 or 2*POSTDIV. Note that if POSTDIV = $00 f_{PLL}= f_{VCO} (divide by one).

11.4 Functional Description

11.4.1 General

This section provides a complete functional description of the MSCAN.

11.4.2 Message Storage

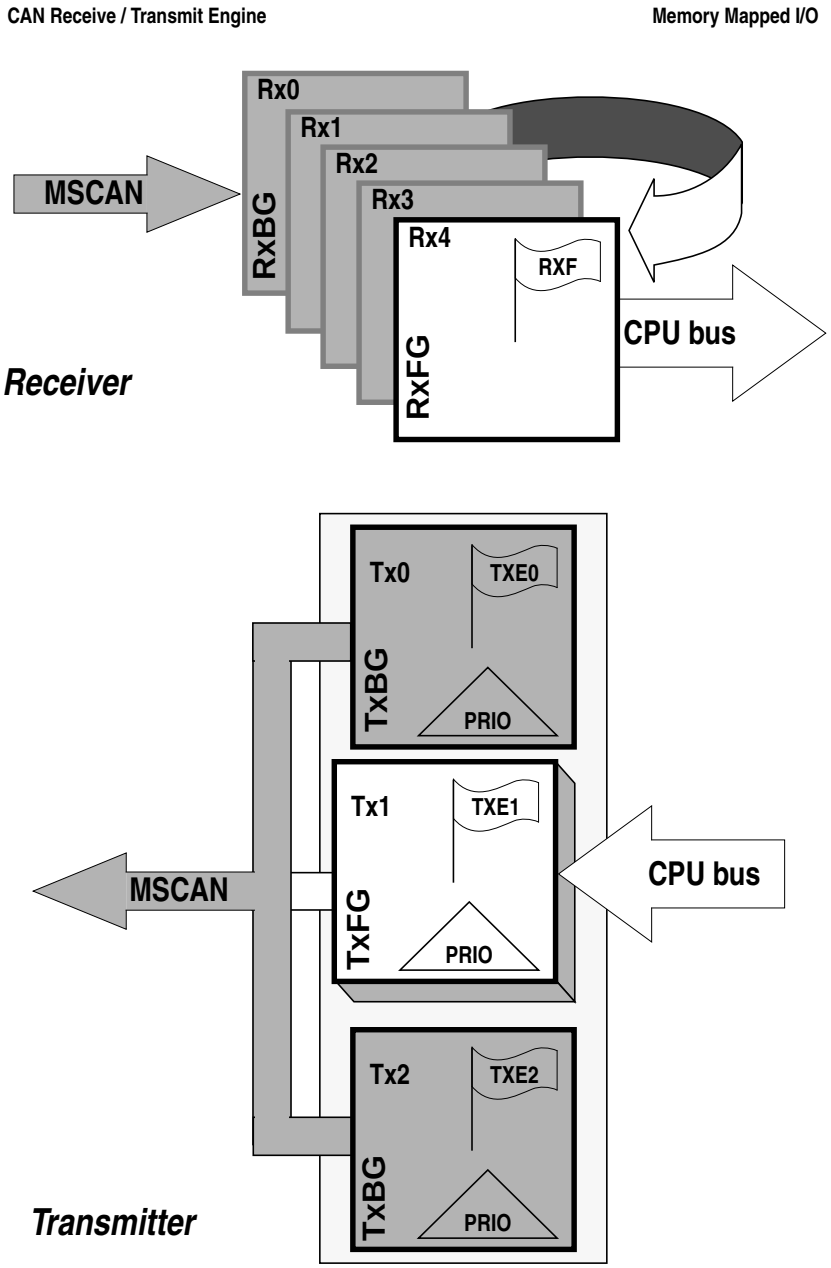

Figure 11-39. User Model for Message Buffer Organization

Table 12-8. PITMTLD0–1 Field Descriptions

Field	Description
7:0 PMTLD[7:0]	**PIT Micro Timer Load Bits 7:0** — These bits set the 8-bit modulus down-counter load value of the micro timers. Writing a new value into the PITMTLD register will not restart the timer. When the micro timer has counted down to zero, the PMTLD register value will be loaded. The PFLMT bits in the PITCFLMT register can be used to immediately update the count register with the new value if an immediate load is desired.

12.4 Functional Description

Figure 12-19 shows a detailed block diagram of the PIT module. The main parts of the PIT are status, control and data registers, two 8-bit down-counters, four 16-bit down-counters and an interrupt/trigger interface.

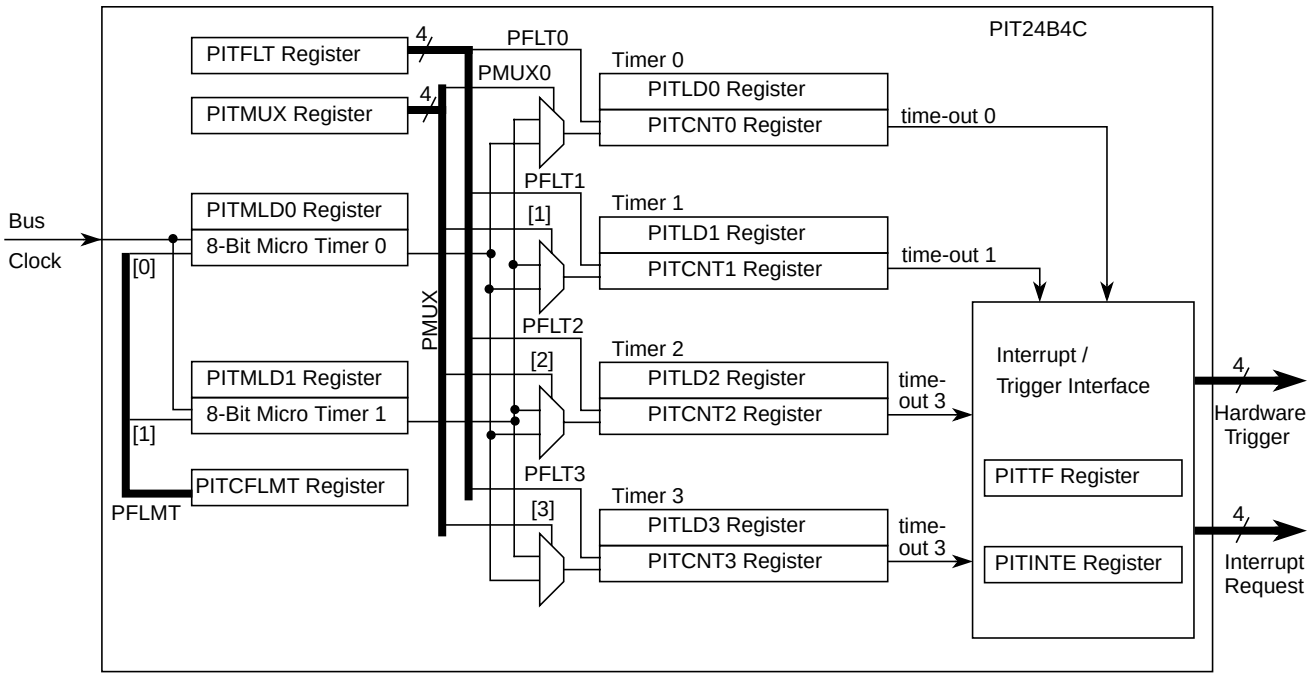

Figure 12-19. PIT24B4C Detailed Block Diagram

12.4.1 Timer

As shown in Figure 12-1 and Figure 12-19, the 24-bit timers are built in a two-stage architecture with four 16-bit modulus down-counters and two 8-bit modulus down-counters. The 16-bit timers are clocked with two selectable micro time bases which are generated with 8-bit modulus down-counters. Each 16-bit timer is connected to micro time base 0 or 1 via the PMUX[3:0] bit setting in the PIT Multiplex (PITMUX) register.

A timer channel is enabled if the module enable bit PITE in the PIT control and force load micro timer (PITCFLMT) register is set and if the corresponding PCE bit in the PIT channel enable (PITCE) register is set. Two 8-bit modulus down-counters are used to generate two micro time bases. As soon as a micro time base is selected for an enabled timer channel, the corresponding micro timer modulus down-counter will load its start value as specified in the PITMTLD0 or PITMTLD1 register and will start down-counting. Whenever the micro timer down-counter has counted to zero the PITMTLD register is reloaded and the connected 16-bit modulus down-counters count one cycle.

13.1.2 Modes of Operation

There is a software programmable option for low power consumption in wait mode that disables the input clock to the prescaler.

In freeze mode there is a software programmable option to disable the input clock to the prescaler. This is useful for emulation.

13.1.3 Block Diagram

Figure 13-1 shows the block diagram for the 8-bit 8-channel PWM block.

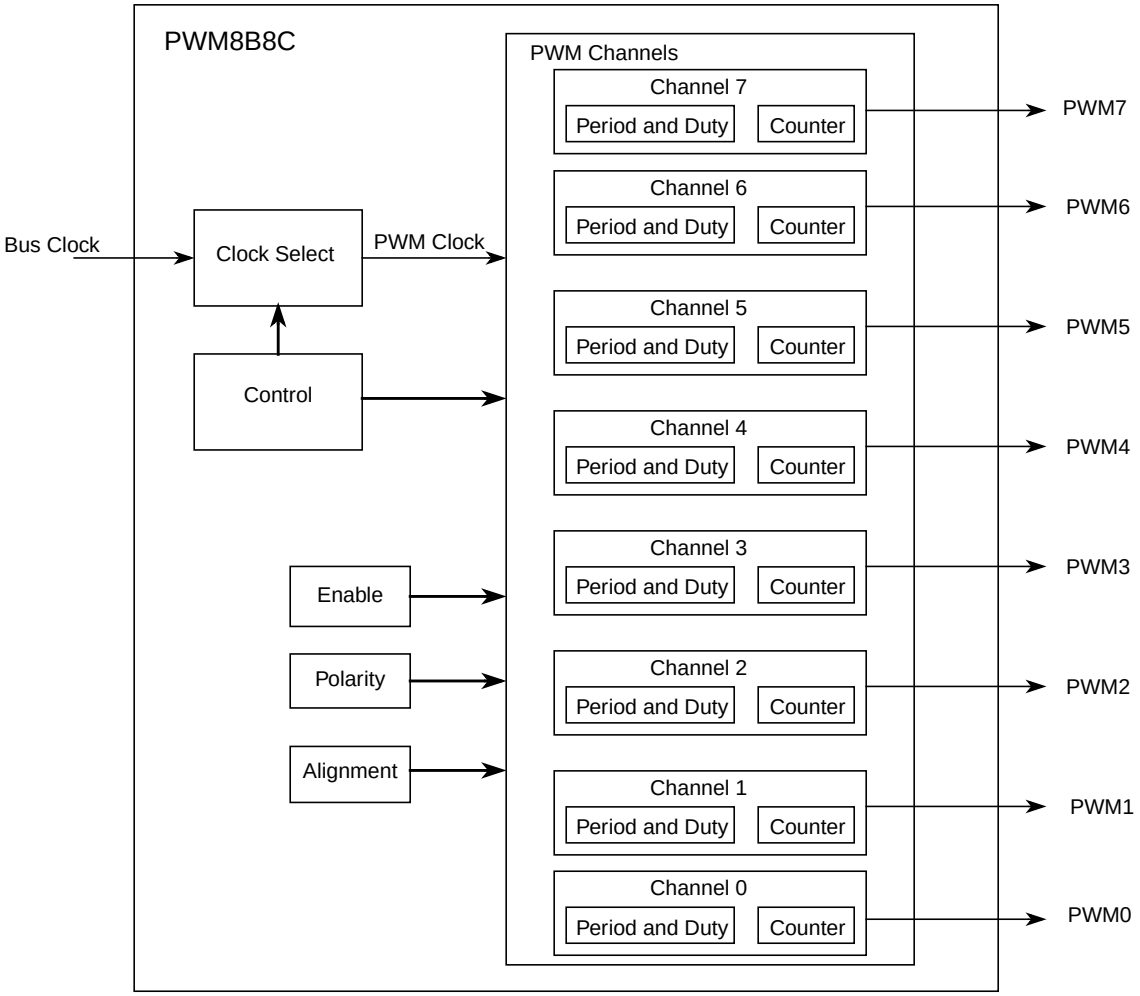

Figure 13-1. PWM Block Diagram

13.2 External Signal Description

The PWM module has a total of 8 external pins.

2 registers become the high order bytes of the double byte channel. When channels 0 and 1 are concatenated, channel 0 registers become the high order bytes of the double byte channel.

See Section 13.4.2.7, “PWM 16-Bit Functions” for a more detailed description of the concatenation PWM Function.

NOTE

Change these bits only when both corresponding channels are disabled.

Table 13-8. PWMCTL Field Descriptions

Field	Description
7 CON67	**Concatenate Channels 6 and 7** 0 Channels 6 and 7 are separate 8-bit PWMs. 1 Channels 6 and 7 are concatenated to create one 16-bit PWM channel. Channel 6 becomes the high order byte and channel 7 becomes the low order byte. Channel 7 output pin is used as the output for this 16-bit PWM (bit 7 of port PWMP). Channel 7 clock select control-bit determines the clock source, channel 7 polarity bit determines the polarity, channel 7 enable bit enables the output and channel 7 center aligned enable bit determines the output mode.
6 CON45	**Concatenate Channels 4 and 5** 0 Channels 4 and 5 are separate 8-bit PWMs. 1 Channels 4 and 5 are concatenated to create one 16-bit PWM channel. Channel 4 becomes the high order byte and channel 5 becomes the low order byte. Channel 5 output pin is used as the output for this 16-bit PWM (bit 5 of port PWMP). Channel 5 clock select control-bit determines the clock source, channel 5 polarity bit determines the polarity, channel 5 enable bit enables the output and channel 5 center aligned enable bit determines the output mode.
5 CON23	**Concatenate Channels 2 and 3** 0 Channels 2 and 3 are separate 8-bit PWMs. 1 Channels 2 and 3 are concatenated to create one 16-bit PWM channel. Channel 2 becomes the high order byte and channel 3 becomes the low order byte. Channel 3 output pin is used as the output for this 16-bit PWM (bit 3 of port PWMP). Channel 3 clock select control-bit determines the clock source, channel 3 polarity bit determines the polarity, channel 3 enable bit enables the output and channel 3 center aligned enable bit determines the output mode.
4 CON01	**Concatenate Channels 0 and 1** 0 Channels 0 and 1 are separate 8-bit PWMs. 1 Channels 0 and 1 are concatenated to create one 16-bit PWM channel. Channel 0 becomes the high order byte and channel 1 becomes the low order byte. Channel 1 output pin is used as the output for this 16-bit PWM (bit 1 of port PWMP). Channel 1 clock select control-bit determines the clock source, channel 1 polarity bit determines the polarity, channel 1 enable bit enables the output and channel 1 center aligned enable bit determines the output mode.
3 PSWAI	**PWM Stops in Wait Mode —** Enabling this bit allows for lower power consumption in wait mode by disabling the input clock to the prescaler. 0 Allow the clock to the prescaler to continue while in wait mode. 1 Stop the input clock to the prescaler whenever the MCU is in wait mode.
2 PFREZ	**PWM Counters Stop in Freeze Mode** — In freeze mode, there is an option to disable the input clock to the prescaler by setting the PFRZ bit in the PWMCTL register. If this bit is set, whenever the MCU is in freeze mode, the input clock to the prescaler is disabled. This feature is useful during emulation as it allows the PWM function to be suspended. In this way, the counters of the PWM can be stopped while in freeze mode so that once normal program flow is continued, the counters are re-enabled to simulate real-time operations. Since the registers can still be accessed in this mode, to re-enable the prescaler clock, either disable the PFRZ bit or exit freeze mode. 0 Allow PWM to continue while in freeze mode. 1 Disable PWM input clock to the prescaler whenever the part is in freeze mode. This is useful for emulation.

NOTE

When PWMSCLA = $00, PWMSCLA value is considered a full scale value of 256. Clock A is thus divided by 512.

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLA).

Module Base + 0x0008

	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 13-11. PWM Scale A Register (PWMSCLA)

Read: Anytime

Write: Anytime (causes the scale counter to load the PWMSCLA value)

13.3.2.10 PWM Scale B Register (PWMSCLB)

PWMSCLB is the programmable scale value used in scaling clock B to generate clock SB. Clock SB is generated by taking clock B, dividing it by the value in the PWMSCLB register and dividing that by two.

Clock SB = Clock B / (2 * PWMSCLB)

NOTE

When PWMSCLB = $00, PWMSCLB value is considered a full scale value of 256. Clock B is thus divided by 512.

Any value written to this register will cause the scale counter to load the new scale value (PWMSCLB).

Module Base + 0x0009

	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 13-12. PWM Scale B Register (PWMSCLB)

Read: Anytime

Write: Anytime (causes the scale counter to load the PWMSCLB value).

13.3.2.11 Reserved Registers (PWMSCNTx)

The registers PWMSCNTA and PWMSCNTB are reserved for factory testing of the PWM module and are not available in normal modes.

17.4 Functional Description

17.4.1 General

Module VREG_3V3 is a voltage regulator, as depicted in Figure 17-1. The regulator functional elements are the regulator core (REG), a low-voltage detect module (LVD), a control block (CTRL), a power-on reset module (POR), and a low-voltage reset module (LVR)and a high temperature sensor (HTD).

17.4.2 Regulator Core (REG)

Respectively its regulator core has three parallel, independent regulation loops (REG1,REG2 and REG3). REG1 and REG3 differ only in the amount of current that can be delivered.

The regulators are linear regulator with a bandgap reference when operated in Full Performance Mode. They act as a voltage clamp in Reduced Power Mode. All load currents flow from input VDDR to VSS or VSSPLL. The reference circuits are supplied by VDDA and VSSA.

17.4.2.1 Full Performance Mode

In Full Performance Mode, the output voltage is compared with a reference voltage by an operational amplifier. The amplified input voltage difference drives the gate of an output transistor.

17.4.2.2 Reduced Power Mode

In Reduced Power Mode, the gate of the output transistor is connected directly to a reference voltage to reduce power consumption. Mode switching from reduced power to full performance requires a transition time of t_{vup}, if the voltage regulator is enabled.

17.4.3 Low-Voltage Detect (LVD)

Subblock LVD is responsible for generating the low-voltage interrupt (LVI). LVD monitors the input voltage (V_{DDA}–V_{SSA}) and continuously updates the status flag LVDS. Interrupt flag LVIF is set whenever status flag LVDS changes its value. The LVD is available in FPM and is inactive in Reduced Power Mode or Shutdown Mode.

17.4.4 Power-On Reset (POR)

This functional block monitors VDD. If V_{DD} is below V_{PORD}, POR is asserted; if V_{DD} exceeds V_{PORD}, the POR is deasserted. POR asserted forces the MCU into Reset. POR Deasserted will trigger the power-on sequence.

17.4.5 Low-Voltage Reset (LVR)

Block LVR monitors the supplies VDD, VDDX and VDDF. If one (or more) drops below it's corresponding assertion level, signal LVR asserts; if all VDD,VDDX and VDDF supplies are above their

Table 18-16. FERSTAT Field Descriptions

Field	Description
1 DFDIF	**Double Bit Fault Detect Interrupt Flag** — The setting of the DFDIF flag indicates that a double bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation was attempted on a Flash block that was under a Flash command operation. The DFDIF flag is cleared by writing a 1 to DFDIF. Writing a 0 to DFDIF has no effect on DFDIF. 0 No double bit fault detected 1 Double bit fault detected or an invalid Flash array read operation attempted
0 SFDIF	**Single Bit Fault Detect Interrupt Flag** — With the IGNSF bit in the FCNFG register clear, the SFDIF flag indicates that a single bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation was attempted on a Flash block that was under a Flash command operation. The SFDIF flag is cleared by writing a 1 to SFDIF. Writing a 0 to SFDIF has no effect on SFDIF. 0 No single bit fault detected 1 Single bit fault detected and corrected or an invalid Flash array read operation attempted

18.3.2.9 P-Flash Protection Register (FPROT)

The FPROT register defines which P-Flash sectors are protected against program and erase operations.

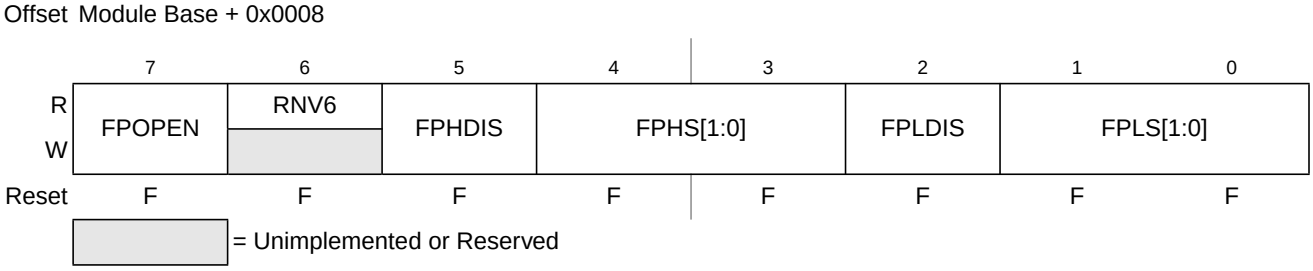

Figure 18-13. Flash Protection Register (FPROT)

The (unreserved) bits of the FPROT register are writable with the restriction that the size of the protected region can only be increased (see Section 18.3.2.9.1, “P-Flash Protection Restrictions,” and Table 18-21).

During the reset sequence, the FPROT register is loaded with the contents of the P-Flash protection byte in the Flash configuration field at global address 0x7F_FF0C located in P-Flash memory (see Table 18-3) as indicated by reset condition ‘F’ in Figure 18-13. To change the P-Flash protection that will be loaded during the reset sequence, the upper sector of the P-Flash memory must be unprotected, then the P-Flash protection byte must be reprogrammed. If a double bit fault is detected while reading the P-Flash phrase containing the P-Flash protection byte during the reset sequence, the FPOPEN bit will be cleared and remaining bits in the FPROT register will be set to leave the P-Flash memory fully protected.

Trying to alter data in any protected area in the P-Flash memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. The block erase of a P-Flash block is not possible if any of the P-Flash sectors contained in the same P-Flash block are protected.

Table 18-64. Erase D-Flash Sector Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 18-28)
		Set if an invalid global address [22:0] is supplied
		Set if a misaligned word address is supplied (global address [0] != 0)
	FPVIOL	Set if the selected area of the D-Flash memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

18.4.3 Interrupts

The Flash module can generate an interrupt when a Flash command operation has completed or when a Flash command operation has detected an ECC fault.

Table 18-65. Flash Interrupt Sources

Interrupt Source	Interrupt Flag	Local Enable	Global (CCR) Mask
Flash Command Complete	CCIF (FSTAT register)	CCIE (FCNFG register)	I Bit
ECC Double Bit Fault on Flash Read	DFDIF (FERSTAT register)	DFDIE (FERCNFG register)	I Bit
ECC Single Bit Fault on Flash Read	SFDIF (FERSTAT register)	SFDIE (FERCNFG register)	I Bit

NOTE

Vector addresses and their relative interrupt priority are determined at the MCU level.

18.4.3.1 Description of Flash Interrupt Operation

The Flash module uses the CCIF flag in combination with the CCIE interrupt enable bit to generate the Flash command interrupt request. The Flash module uses the DFDIF and SFDIF flags in combination with the DFDIE and SFDIE interrupt enable bits to generate the Flash error interrupt request. For a detailed description of the register bits involved, refer to Section 18.3.2.5, “Flash Configuration Register (FCNFG)”, Section 18.3.2.6, “Flash Error Configuration Register (FERCNFG)”, Section 18.3.2.7, “Flash Status Register (FSTAT)”, and Section 18.3.2.8, “Flash Error Status Register (FERSTAT)”.

The logic used for generating the Flash module interrupts is shown in Figure 18-27.

Chapter 19
128 KByte Flash Module (S12XFTMR128K1V1)

Table 19-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V01.04	03 Jan 2008		- Cosmetic changes
V01.05	19 Dec 2008	19.1/19-557 19.4.2.4/19-592 19.4.2.6/19-594 19.4.2.11/19-597 19.4.2.11/19-597 19.4.2.11/19-597	- Clarify single bit fault correction for P-Flash phrase - Add statement concerning code runaway when executing Read Once, Program Once, and Verify Backdoor Access Key commands from Flash block containing associated fields - Relate Key 0 to associated Backdoor Comparison Key address - Change “power down reset” to “reset” in Section 19.4.2.11
V01.06	25 Sep 2009	19.3.2/19-564 19.3.2.1/19-566 19.4.1.2/19-586 19.6/19-606	The following changes were made to clarify module behavior related to Flash register access during reset sequence and while Flash commands are active: - Add caution concerning register writes while command is active - Writes to FCLKDIV are allowed during reset sequence while CCIF is clear - Add caution concerning register writes while command is active - Writes to FCCOBIX, FCCOBHI, FCCOBLO registers are ignored during reset sequence

19.1 Introduction

The FTMR128K1 module implements the following:

- 128 Kbytes of P-Flash (Program Flash) memory
- 8 Kbytes of D-Flash (Data Flash) memory

The Flash memory is ideal for single-supply applications allowing for field reprogramming without requiring external high voltage sources for program or erase operations. The Flash module includes a memory controller that executes commands to modify Flash memory contents. The user interface to the memory controller consists of the indexed Flash Common Command Object (FCCOB) register which is written to with the command, global address, data, and any required command parameters. The memory controller must complete the execution of a command before the FCCOB register can be written to with a new command.

Address & Name		7	6	5	4	3	2	1	0
0x0010 FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
	W								
0x0011 FRSV2	R	0	0	0	0	0	0	0	0
	W								
0x0012 FRSV3	R	0	0	0	0	0	0	0	0
	W								
0x0013 FRSV4	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 19-4. FTMR128K1 Register Summary (continued)

19.3.2.1 Flash Clock Divider Register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

Offset Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	FDIVLD	FDIV[6:0]						
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 19-5. Flash Clock Divider Register (FCLKDIV)

All bits in the FCLKDIV register are readable, bits 6–0 are write once and bit 7 is not writable.

Table 19-6. FCLKDIV Field Descriptions

Field	Description
7 FDIVLD	**Clock Divider Loaded** 0 FCLKDIV register has not been written 1 FCLKDIV register has been written since the last reset
6–0 FDIV[6:0]	**Clock Divider Bits** — FDIV[6:0] must be set to effectively divide OSCCLK down to generate an internal Flash clock, FCLK, with a target frequency of 1 MHz for use by the Flash module to control timed events during program and erase algorithms. Table 19-7 shows recommended values for FDIV[6:0] based on OSCCLK frequency. Please refer to Section 19.4.1, “Flash Command Operations,” for more information.

Offset Module Base + 0x0005

	7	6	5	4	3	2	1	0
R			0				DFDIE	SFDIE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 19-10. Flash Error Configuration Register (FERCNFG)

All assigned bits in the FERCNFG register are readable and writable.

Table 19-14. FERCNFG Field Descriptions

Field	Description
1 DFDIE	**Double Bit Fault Detect Interrupt Enable** — The DFDIE bit controls interrupt generation when a double bit fault is detected during a Flash block read operation. 0 DFDIF interrupt disabled 1 An interrupt will be requested whenever the DFDIF flag is set (see Section 19.3.2.8)
0 SFDIE	**Single Bit Fault Detect Interrupt Enable** — The SFDIE bit controls interrupt generation when a single bit fault is detected during a Flash block read operation. 0 SFDIF interrupt disabled whenever the SFDIF flag is set (see Section 19.3.2.8) 1 An interrupt will be requested whenever the SFDIF flag is set (see Section 19.3.2.8)

19.3.2.7 Flash Status Register (FSTAT)

The FSTAT register reports the operational status of the Flash module.

Offset Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT[1:0]	
W								
Reset	1	0	0	0	0	0	0[1]	0[1]

= Unimplemented or Reserved

Figure 19-11. Flash Status Register (FSTAT)

[1] Reset value can deviate from the value shown if a double bit fault is detected during the reset sequence (see Section 19.6).

CCIF, ACCERR, and FPVIOL bits are readable and writable, MGBUSY and MGSTAT bits are readable but not writable, while remaining bits read 0 and are not writable.

CAUTION

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

Table 20-61. Program D-Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x11	Global address [22:16] to identify the D-Flash block
001	Global address [15:0] of word to be programmed	
010	Word 0 program value	
011	Word 1 program value, if desired	
100	Word 2 program value, if desired	
101	Word 3 program value, if desired	

Upon clearing CCIF to launch the Program D-Flash command, the user-supplied words will be transferred to the Memory Controller and be programmed if the area is unprotected. The CCOBIX index value at Program D-Flash command launch determines how many words will be programmed in the D-Flash block. The CCIF flag is set when the operation has completed.

Table 20-62. Program D-Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] < 010 at command launch
		Set if CCOBIX[2:0] > 101 at command launch
		Set if command not available in current mode (see Table 20-28)
		Set if an invalid global address [22:0] is supplied
		Set if a misaligned word address is supplied (global address [0] != 0)
		Set if the requested group of words breaches the end of the D-Flash block
	FPVIOL	Set if the selected area of the D-Flash memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

20.4.2.16 Erase D-Flash Sector Command

The Erase D-Flash Sector operation will erase all addresses in a sector of the D-Flash block.

Table 20-63. Erase D-Flash Sector Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x12	Global address [22:16] to identify D-Flash block

Appendix E
Detailed Register Address Map

E.1 Detailed Register Map

The following tables show the detailed register map of the S12XS family.

0x0000–0x0009 Port Integration Module (PIM) Map 1 of 5

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0000	PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA 0
		W								
0x0001	PORTB	R	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
		W								
0x0002	DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
0x0003	DDRB	R	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		W								
0x0004	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0005	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0006	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0007	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0008	PORTE	R	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
		W								
0x0009	DDRE	R	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	0	0
		W								

0x000A–0x000B Module Mapping Control (S12XMMC) Map 1 of 2

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x000A	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x000B	MODE	R	MODC	0	0	0	0	0	0	0
		W								

0x000C–0x000D Port Integration Module (PIM) Map 2 of 5

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x000C	PUCR	R	PUPKE	BKPUE	0	PUPEE	0	0	PUPBE	PUPAE
		W								
0x000D	RDRIV	R	RDPK	0	0	RDPE	0	0	RDPB	RDPA
		W								