MC9S12XEP100
Reference Manual
Covers MC9S12XE Family

HCS12X
Microcontrollers

MC9S12XEP100RMV1

Rev. 1.25

02/2013

freescale.com

To provide the most up-to-date information, the document revision on the World Wide Web is the most current. A printed copy may be an earlier revision. To verif, refer to:

freescale.com

This document contains information for the complete S12XE-Family and thus includes a set of separate FTM module sections to cover the whole family. A full list of family members and options is included in the appendices.

This document contains information for all constituent modules, with the exception of the S12X CPU. For S12X CPU information please refer to CPU12XV2 in the CPU12/CPU12X Reference Manual.

Revision History. Refer to module section revision history tables for more information.

Date	Revision	Description
Sep, 2008	1.18	Updated NVM timing parameter section for brownout case Specified time delay from RESET to start of CPU code execution Added NVM patch Part IDs Enhanced ECT GPIO / timer function transitioning description
Dec, 2008	1.19	Updated 208MAPBGA thermal parameters Revised TIM flag clearing procedure Corrected CRG register address Added maskset identifier suffix for ATMC fab Fixed typos
Aug, 2009	1.20	Added 208MAPBGA disclaimer Added VREAPI to PT5. Added LVR Note to electricals. Updates to TIM/ECT/XGATE/SCI/MSCAN (see embedded rev. history)
Apr, 2010	1.21	FTM section (see FTM revision history) PIM section (see PIM revision history)
May, 2010	1.22	ECT and TIM sections (see ECT, TIM revision history tables) BDM Alternate clock source defined in device overview
Sep, 2010	1.23	Added S12XEG256 option. Updated MSCAN section
Aug, 2012	1.24	Added bandgap voltage to electricals Added new maskset and Part ID numbers Minor updates to MSCAN,SCI and S12XINT sections Removed BGA disclaimer
Feb, 2013	1.25	Updated MSCAN section Formatting updates and minor corrections in PWM, CRG, BDM, DBG sections Updated Ordering Information

Blank Page

Blank Page

Chapter 1
Device Overview MC9S12XE-Family

Chapter 2
Port Integration Module (S12XEPIMV1)

Chapter 3
Memory Mapping Control (S12XMMCV4)

Chapter 4
Memory Protection Unit (S12XMPUV1)

Chapter 5
External Bus Interface (S12XEBIV4)

Chapter 6
Interrupt (S12XINTV2)

Chapter 7
Background Debug Module (S12XBDMV2)

Chapter 8
S12X Debug (S12XDBGV3) Module

Chapter 9
Security (S12XE9SECV2)

Chapter 10
XGATE (S12XGATEV3)

Chapter 11
S12XE Clocks and Reset Generator (S12XECRGV1)

Chapter 12
Pierce Oscillator (S12XOSCLCPV2)

Chapter 13
Analog-to-Digital Converter (ADC12B16CV1)

Chapter 14
Enhanced Capture Timer (ECT16B8CV3)

Chapter 15
Inter-Integrated Circuit (IICV3) Block Description

Chapter 16
Freescale’s Scalable Controller Area Network (S12MSCANV3)

Chapter 17
Periodic Interrupt Timer (S12PIT24B8CV2)

Chapter 18
Periodic Interrupt Timer (S12PIT24B4CV2)

Chapter 19
Pulse-Width Modulator (S12PWM8B8CV1)

Chapter 20
Serial Communication Interface (S12SCIV5)

Chapter 21
Serial Peripheral Interface (S12SPIV5)

Chapter 22
Timer Module (TIM16B8CV2) Block Description

Chapter 23
Voltage Regulator (S12VREGL3V3V1)

Chapter 24
128 KByte Flash Module (S12XFTM128K2V1)

Chapter 25
256 KByte Flash Module (S12XFTM256K2V1)

Chapter 26
384 KByte Flash Module (S12XFTM384K2V1)

Chapter 27
512 KByte Flash Module (S12XFTM512K3V1)

Chapter 28
768 KByte Flash Module (S12XFTM768K4V2)

Chapter 29
1024 KByte Flash Module (S12XFTM1024K5V2)

Appendix A Electrical Characteristics

Appendix B
Package Information

Appendix C
PCB Layout Guidelines

Appendix D
Derivative Differences

Appendix E
Detailed Register Address Map

Appendix F
Ordering Information

Chapter 1
Device Overview MC9S12XE-Family

1.1 Introduction

The MC9S12XE-Family of micro controllers is a further development of the S12XD-Family including new features for enhanced system integrity and greater functionality. These new features include a Memory Protection Unit (MPU) and Error Correction Code (ECC) on the Flash memory together with enhanced EEPROM functionality (EEE), an enhanced XGATE, an Internally filtered, frequency modulated Phase Locked Loop (IPLL) and an enhanced ATD. The E-Family extends the S12X product range up to 1MB of Flash memory with increased I/O capability in the 208-pin version of the flagship MC9S12XE100.

The MC9S12XE-Family delivers 32-bit performance with all the advantages and efficiencies of a 16 bit MCU. It retains the low cost, power consumption, EMC and code-size efficiency advantages currently enjoyed by users of Freescale's existing 16-Bit MC9S12 and S12X MCU families. There is a high level of compatibility between the S12XE and S12XD families.

The MC9S12XE-Family features an enhanced version of the performance-boosting XGATE co-processor which is programmable in "C" language and runs at twice the bus frequency of the S12X with an instruction set optimized for data movement, logic and bit manipulation instructions and which can service any peripheral module on the device. The new enhanced version has improved interrupt handling capability and is fully compatible with the existing XGATE module.

The MC9S12XE-Family is composed of standard on-chip peripherals including up to 64Kbytes of RAM, eight asynchronous serial communications interfaces (SCI), three serial peripheral interfaces (SPI), an 8-channel IC/OC enhanced capture timer (ECT), two 16-channel, 12-bit analog-to-digital converters, an 8-channel pulse-width modulator (PWM), five CAN 2.0 A, B software compatible modules (MSCAN12), two inter-IC bus blocks (IIC), an 8-channel 24-bit periodic interrupt timer (PIT) and an 8-channel 16-bit standard timer module (TIM).

The MC9S12XE-Family uses 16-bit wide accesses without wait states for all peripherals and memories. The non-multiplexed expanded bus interface available on the 144/208-Pin versions allows an easy interface to external memories.

In addition to the I/O ports available in each module, up to 26 further I/O ports are available with interrupt capability allowing Wake-Up from STOP or WAIT modes. The MC9S12XE-Family is available in 208-Pin MAPBGA, 144-Pin LQFP, 112-Pin LQFP or 80-Pin QFP options.

1.1.1 Features

Features of the MC9S12XE-Family are listed here. Please see Table D-2.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 MC9S12 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
- INT (interrupt module)
  - Eight levels of nested interrupts
  - Flexible assignment of interrupt sources to each interrupt level.
  - External non-maskable high priority interrupt (XIRQ)
  - Internal non-maskable high priority Memory Protection Unit interrupt
  - Up to 24 pins on ports J, H and P configurable as rising or falling edge sensitive interrupts
- EBI (external bus interface)(available in 208-Pin and 144-Pin packages only)
  - Up to four chip select outputs to select 16K, 1M, 2M and up to 4MByte address spaces
  - Each chip select output can be configured to complete transaction on either the time-out of one of the two wait state generators or the deassertion of EWAIT signal
- MMC (module mapping control)
- DBG (debug module)
  - Monitoring of CPU and/or XGATE busses with tag-type or force-type breakpoint requests
  - 64 x 64-bit circular trace buffer captures change-of-flow or memory access information
- BDM (background debug mode)
- MPU (memory protection unit)
  - 8 address regions definable per active program task
  - Address range granularity as low as 8-bytes
  - No write / No execute Protection Attributes
  - Non-maskable interrupt on access violation
- XGATE
  - Programmable, high performance I/O coprocessor module
  - Transfers data to or from all peripherals and RAM without CPU intervention or CPU wait states
  - Performs logical, shifts, arithmetic, and bit operations on data
  - Can interrupt the HCS12X CPU signalling transfer completion
  - Triggers from any hardware module as well as from the CPU possible
  - Two interrupt levels to service high priority tasks
  - Hardware support for stack pointer initialisation
- OSC_LCP (oscillator)
  - Low power loop control Pierce oscillator utilizing a 4MHz to 16MHz crystal
  - Good noise immunity
  - Full-swing Pierce option utilizing a 2MHz to 40MHz crystal
  - Transconductance sized for optimum start-up margin for typical crystals
- IPLL (Internally filtered, frequency modulated phase-locked-loop clock generation)

  - No external components required
  - Configurable option to spread spectrum for reduced EMC radiation (frequency modulation)
- CRG (clock and reset generation)
  - COP watchdog
  - Real time interrupt
  - Clock monitor
  - Fast wake up from STOP in self clock mode
- Memory Options
  - 128K, 256k, 384K, 512K, 768K and 1M byte Flash
  - 2K, 4K byte emulated EEPROM
  - 12K, 16K, 24K, 32K, 48K and 64K Byte RAM
- Flash General Features
  - 64 data bits plus 8 syndrome ECC (Error Correction Code) bits allow single bit failure correction and double fault detection
  - Erase sector size 1024 bytes
  - Automated program and erase algorithm
- D-Flash Features
  - Up to 32 Kbytes of D-Flash memory with 256 byte sectors for user access.
  - Dedicated commands to control access to the D-Flash memory over EEE operation.
  - Single bit fault correction and double bit fault detection within a word during read operations.
  - Automated program and erase algorithm with verify and generation of ECC parity bits.
  - Fast sector erase and word program operation.
  - Ability to program up to four words in a burst sequence
- Emulated EEPROM Features
  - Automatic EEE file handling using an internal Memory Controller.
  - Automatic transfer of valid EEE data from D-Flash memory to buffer RAM on reset.
  - Ability to monitor the number of outstanding EEE related buffer RAM words left to be programmed into D-Flash memory.
  - Ability to disable EEE operation and allow priority access to the D-Flash memory.
  - Ability to cancel all pending EEE operations and allow priority access to the D-Flash memory.
- Two 16-channel, 12-bit Analog-to-Digital Converters
  - 8/10/12 Bit resolution
  - 3μs, 10-bit single conversion time
  - Left/right, signed/unsigned result data
  - External and internal conversion trigger capability
  - Internal oscillator for conversion in Stop modes
  - Wake from low power modes on analog comparison > or <= match
- Five MSCAN (1 M bit per second, CAN 2.0 A, B software compatible modules)
  - Five receive and three transmit buffers

- Flexible identifier filter programmable as 2 x 32 bit, 4 x 16 bit, or 8 x 8 bit
- Four separate interrupt channels for Rx, Tx, error, and wake-up
- Low-pass filter wake-up function
- Loop-back for self-test operation

- ECT (enhanced capture timer)
  - 8 x 16-bit channels for input capture or output compare
  - 16-bit free-running counter with 8-bit precision prescaler
  - 16-bit modulus down counter with 8-bit precision prescaler
  - Four 8-bit or two 16-bit pulse accumulators
- TIM (standard timer module)
  - 8 x 16-bit channels for input capture or output compare
  - 16-bit free-running counter with 8-bit precision prescaler
  - 1 x 16-bit pulse accumulator
- PIT (periodic interrupt timer)
  - Up to eight timers with independent time-out periods
  - Time-out periods selectable between 1 and 2^{24} bus clock cycles
  - Time-out interrupt and peripheral triggers
- 8 PWM (pulse-width modulator) channels
  - 8 channel x 8-bit or 4 channel x 16-bit Pulse Width Modulator
  - programmable period and duty cycle per channel
  - Center- or left-aligned outputs
  - Programmable clock select logic with a wide range of frequencies
  - Fast emergency shutdown input
- Three Serial Peripheral Interface Modules (SPI)
  - Configurable for 8 or 16-bit data size
- Eight Serial Communication Interfaces (SCI)
  - Standard mark/space non-return-to-zero (NRZ) format
  - Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with programmable pulse widths
- Two Inter-IC bus (IIC) Modules
  - Multi-master operation
  - Software programmable for one of 256 different serial clock frequencies
  - Broadcast mode support
  - 10-bit address support
- On-Chip Voltage Regulator
  - Two parallel, linear voltage regulators with bandgap reference
  - Low-voltage detect (LVD) with low-voltage interrupt (LVI)
  - Power-on reset (POR) circuit
  - 3.3V and 5V range operation
  - Low-voltage reset (LVR)

- Low-power wake-up timer (API)
  - Available in all modes including Full Stop Mode
  - Trimmable to +-5% accuracy
  - Time-out periods range from 0.2ms to ~13s with a 0.2ms resolution
- Input/Output
  - Up to 152 general-purpose input/output (I/O) pins plus 2 input-only pins
  - Hysteresis and configurable pull up/pull down device on all input pins
  - Configurable drive strength on all output pins
- Package Options
  - 208-pin MAPBGA
  - 144-pin low-profile quad flat-pack (LQFP)
  - 112-pin low-profile quad flat-pack (LQFP)
  - 80-pin quad flat-pack (QFP)
- 50MHz maximum CPU bus frequency, 100MHz maximum XGATE bus frequency

1.1.2 Modes of Operation

Memory map and bus interface modes:

- Normal and emulation operating modes
  - Normal single-chip mode
  - Normal expanded mode
  - Emulation of single-chip mode
  - Emulation of expanded mode
- Special Operating Modes
  - Special single-chip mode with active background debug mode
  - Special test mode (**Freescale use only**)

Low-power modes:

- System stop modes
  - Pseudo stop mode
  - Full stop mode with fast wake-up option
- System wait mode

Operating system states

- Supervisor state
- User state

1.1.3 Block Diagram

Figure 1-1 shows a block diagram of the MC9S12XE-Family devices

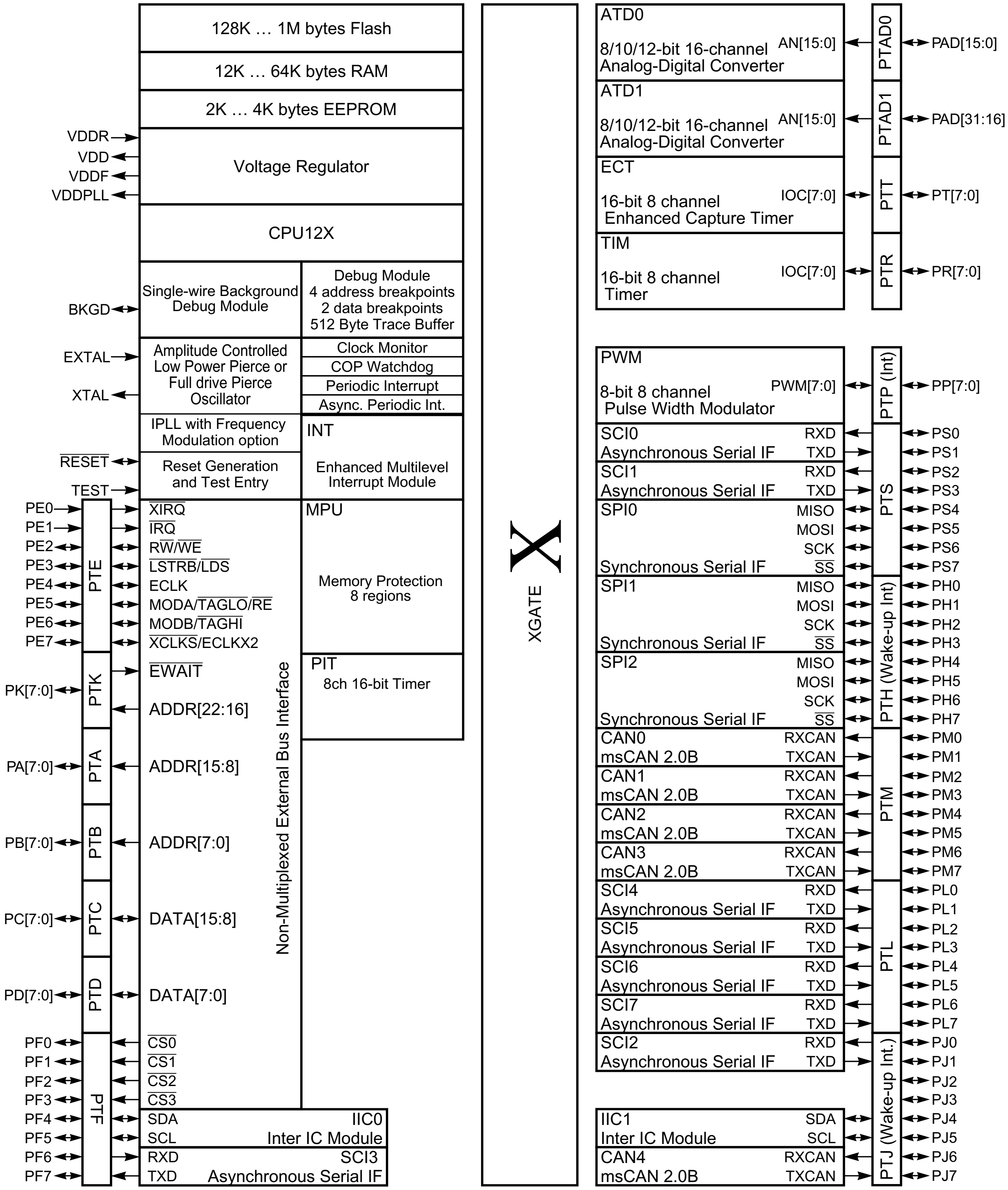


Figure 1-1. MC9S12XE-Family Block Diagram

1.1.4 Device Memory Map

Table 1-1 shows the device register memory map.

Table 1-1. Device Register Memory Map

Address	Module	Size (Bytes)
0x0000–0x0009	PIM (port integration module)	10
0x000A–0x000B	MMC (memory map control)	2
0x000C–0x000D	PIM (port integration module)	2
0x000E–0x000F	EBI (external bus interface)	2
0x0010–0x0017	MMC (memory map control)	8
0x0018–0x0019	Reserved	2
0x001A–0x001B	Device ID register	2
0x001C–0x001F	PIM (port integration module)	4
0x0020–0x002F	DBG (debug module)	16
0x0030–0x0031	Reserved	2
0x0032–0x0033	PIM (port integration module)	2
0x0034–0x003F	ECRG (clock and reset generator)	12
0x0040–0x007F	ECT (enhanced capture timer 16-bit 8-channel)s	64
0x0080–0x00AF	ATD1 (analog-to-digital converter 12-bit 16-channel)	48
0x00B0–0x00B7	IIC1 (inter IC bus)	8
0x00B8–0x00BF	SCI2 (serial communications interface)	8
0x00C0–0x00C7	SCI3 (serial communications interface)	8
0x00C8–0x00CF	SCI0 (serial communications interface)	8
0x00D0–0x00D7	SCI1 (serial communications interface)	8
0x00D8–0x00DF	SPI0 (serial peripheral interface)	8
0x00E0–0x00E7	IIC0 (inter IC bus)	8
0x00E8–0x00EF	Reserved	8
0x00F0–0x00F7	SPI1 (serial peripheral interface)	8
0x00F8–0x00FF	SPI2 (serial peripheral interface)	8
0x0100–0x0113	FTM control registers	20
0x0114–0x011F	MPU (memory protection unit)	12
0x0120–0x012F	INT (interrupt module)	16
0x0130–0x0137	SCI4 (serial communications interface)	8
0x0138–0x013F	SCI5 (serial communications interface)	8
0x0140–0x017F	CAN0	64
0x0180–0x01BF	CAN1	64
0x01C0–0x01FF	CAN2	64

Table 1-1. Device Register Memory Map (continued)

Address	Module	Size (Bytes)
0x0200–0x023F	CAN3	64
0x0240–0x027F	PIM (port integration module)	64
0x0280–0x02BF	CAN4	64
0x02C0–0x02EF	ATD0 (analog-to-digital converter 12 bit 16-channel)	48
0x02F0–0x02F7	Voltage regulator	8
0x02F8–0x02FF	Reserved	8
0x0300–0x0327	PWM (pulse-width modulator 8 channels)	40
0x0328–0x032F	Reserved	8
0x0330–0x0337	SCI6 (serial communications interface)	8
0x0338–0x033F	SCI7 (serial communications interface)	8
0x0340–0x0367	PIT (periodic interrupt timer)	40
0x0368–0x037F	PIM (port integration module)	24
0x0380–0x03BF	XGATE	64
0x03C0–0x03CF	Reserved	16
0x03D0–0x03FF	TIM (timer module)	48
0x0400–0x07FF	Reserved	1024

NOTE

Reserved register space shown in Table 1-1 is not allocated to any module. This register space is reserved for future use. Writing to these locations have no effect. Read access to these locations returns zero.

1.1.5 Address Mapping

Figure 1-2 shows S12XE CPU & BDM local address translation to the global memory map. It indicates also the location of the internal resources in the memory map.

EEEPROM size is presented like a fixed 256 KByte in the memory map.

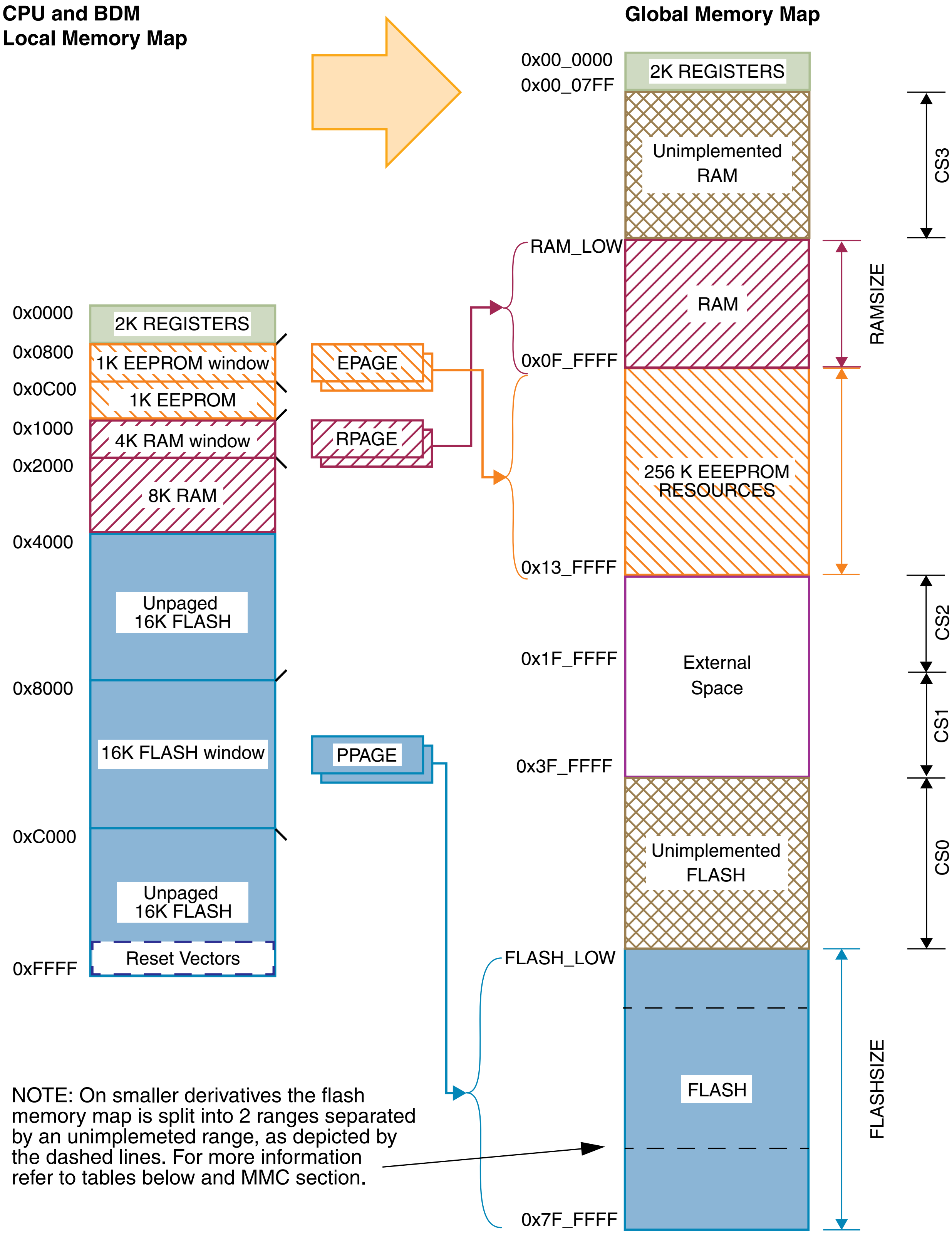


Figure 1-2. MC9S12XE100 Global Memory Map

Unimplemented RAM pages are mapped externally in expanded modes. Accessing unimplemented RAM pages in single chip modes causes an illegal address reset if the MPU is not configured to flag an MPU protection error in that range.

Accessing unimplemented FLASH pages in single chip modes causes an illegal address reset if the MPU is not configured to flag an MPU protection error in that range.

The PARTID value should be referenced regarding the specific memory map for any given device. For devices sharing the same part ID, the memory regions which are implemented on the larger device but not supported on the smaller device are implemented but untested on that smaller device. These regions do not appear as unimplemented in the memory map and do not result in an illegal address reset if erroneously accessed.

Table 1-2. Unimplemented Range Mapping to Part ID

Part ID	RAM_LOW	EE_LOW	Flash Blocks	Registers
0xCC8x	0x0F_0000	0x13_F000	B3, B2, B1S, B1N, B0	2K
0xCC9x	0x0F_0000	0x13_F000	B3, B2, B1S, B1N, B0	2K
0xC48x	0x0F_8000	0x13_F000	B1N, B1S, B0	2K
0xC08x	0x0F_C000	0x13_F000	B1S, B0(128K)	2K

From the above the following examples can be derived.

The 9S12XEP768 is currently only available as a 9S12XEP100 die, thus the unimplemented FLASH pages are those of the 9S12XEP100 device map.

The 9S12XEQ384, 9S12XEG384, 9S12XES384 are currently only available as a 9S12XEQ512 die, thus the unimplemented FLASH pages are those of the 9S12XEQ512 device map.

The 9S12XEG128 is currently only available as a 9S12XET256 die, thus the unimplemented FLASH pages are those of the 9S12XET256 device map.

The range between 0x10_0000 and 0x13_FFFF is mapped to EEPROM resources. The actual EEPROM and dataflash block sizes are listed in Table 1-4. Within EEPROM resource range an address range exists which is neither used by EEPROM resources nor remapped to external resources via chip selects (see the FTM/MMC descriptions for details). These ranges do not constitute unimplemented areas.

Accessing reserved registers within the 2K register space does not generate an illegal address reset.

The fixed 8K RAM default location in the global map is 0x0F_E000- 0x0F_FFFF. This is subject to remapping when configuring the local address map for a larger RAM access range.

Figure 1-3 shows XGATE local address translation to the global memory map. It indicates also the location of used internal resources in the memory map.

Table 1-3. XGATE Resources

Internal Resource	Size /KByte	$Address
XGATE RAM	32K	XGRAM_LOW = 0x0F_8000
FLASH	30K[1]	XGFLASH_HIGH = 0x78_8000

[1] This value is calculated by the following formula: (64K -2K- XGRAMSIZE)

Table 1-4. Derivative Dependent Memory Parameters

Device	FLASH_LOW	PPAGE [1]	RAM_LOW	RPAGE [2]	EE_LOW	EPAGE
9S12XEP100	0x70_0000	64	0x0F_0000	16	0x13_F000	4[3] + 32[4]
9S12XEP768	0x74_0000	48	0x0F_4000	12	0x13_F000	4 + 32
9S12XEQ512	0x78_0000	32	0x0F_8000	8	0x13_F000	4 + 32
9S12XEx384	0x78_0000[5]	24	0x0F_A000	6	0x13_F000	4 + 32
9S12XET256 9S12XEA256 [6]	0x78_0000[7]	16	0x0F_C000	4	0x13_F000	4 + 32
9S12XEG128 9S12XEA128[6]	0x78_0000[8]	8	0x0F_D000	3	0x13_F800	2 + 32

[1] Number of 16K pages addressable via PPAGE register
[2] Number of 4K pages addressing the RAM. RAM can also be mapped to 0x4000 - 0x7FFF
[3] Number of 1K pages addressing the Cache RAM via the EPAGE register counting downwards from 0xFF
[4] Number of 1K pages addressing the Data flash via the EPAGE register starting upwards from 0x00
[5] The 384K memory map is split into a 128K block from 0x78_0000 to 0x79_FFFF and a 256K block from 0x7C_0000 to 0x7F_FFFF
[6] The 9S12XEA devices are a special bondout for access to extra ADC channels in 80QFP. Available in 80QFP only. WARNING: NOT PIN-COMPATIBLE WITH REST OF FAMILY.
[7] The 256K memory map is split into a 128K block from 0x78_0000 to 0x79_FFFF and a 128K block from 0x7E_0000 to 0x7F_FFFF
[8] The 128K memory map is split into a 64K block from 0x78_0000 to 0x78_FFFF and a 64K block from 0x7F_0000 to 0x7F_FFFF

Table 1-5. Derivative Dependent Flash Block Mapping

Device	0x70_0000	0x74_0000	0x78_0000	0x7A_0000	0x7C_0000	0x7E_0000
9S12XEP100	B3	B2	B1S	B1N	B0	
9S12XEP768	—	B2	B1S	B1N	B0	
9S12XEQ512	—	—	B1S	B1N	B0	
9S12XEx384	—	—	B1S	—	B0	

Table 1-5. Derivative Dependent Flash Block Mapping (continued)

Device	0x70_0000	0x74_0000	0x78_0000	0x7A_0000	0x7C_0000	0x7E_0000
9S12XET256 9S12XEA256 (1)	—	—	B1S	—	—	B0(128K)
9S12XEG128 9S12XEA128[1]	—	—	B1S (64K)	—	—	B0 (64K)

[1] The 9S12XEA devices are special bondouts for access to extra ADC channels in 80QFP. Available in 80QFP only. WARNING: NOT PIN-COMPATIBLE WITH REST OF FAMILY.

Block B1 is divided into two 128K blocks. The XGATE is always mapped to block B1S.

On the 9S12XEG128 the flash is divided into two 64K blocks B0 and B1S, the B1S range extending from 0x78_0000 to 0x78_FFFF, the B0 range extending from 0x7F_0000 to 0x7F_FFFF.

The block B0 is a reduced size 128K block on the 256K derivative. On the larger derivatives B0 is a 256K block. The block B0 is a reduced size 64K block on the 128K derivative.

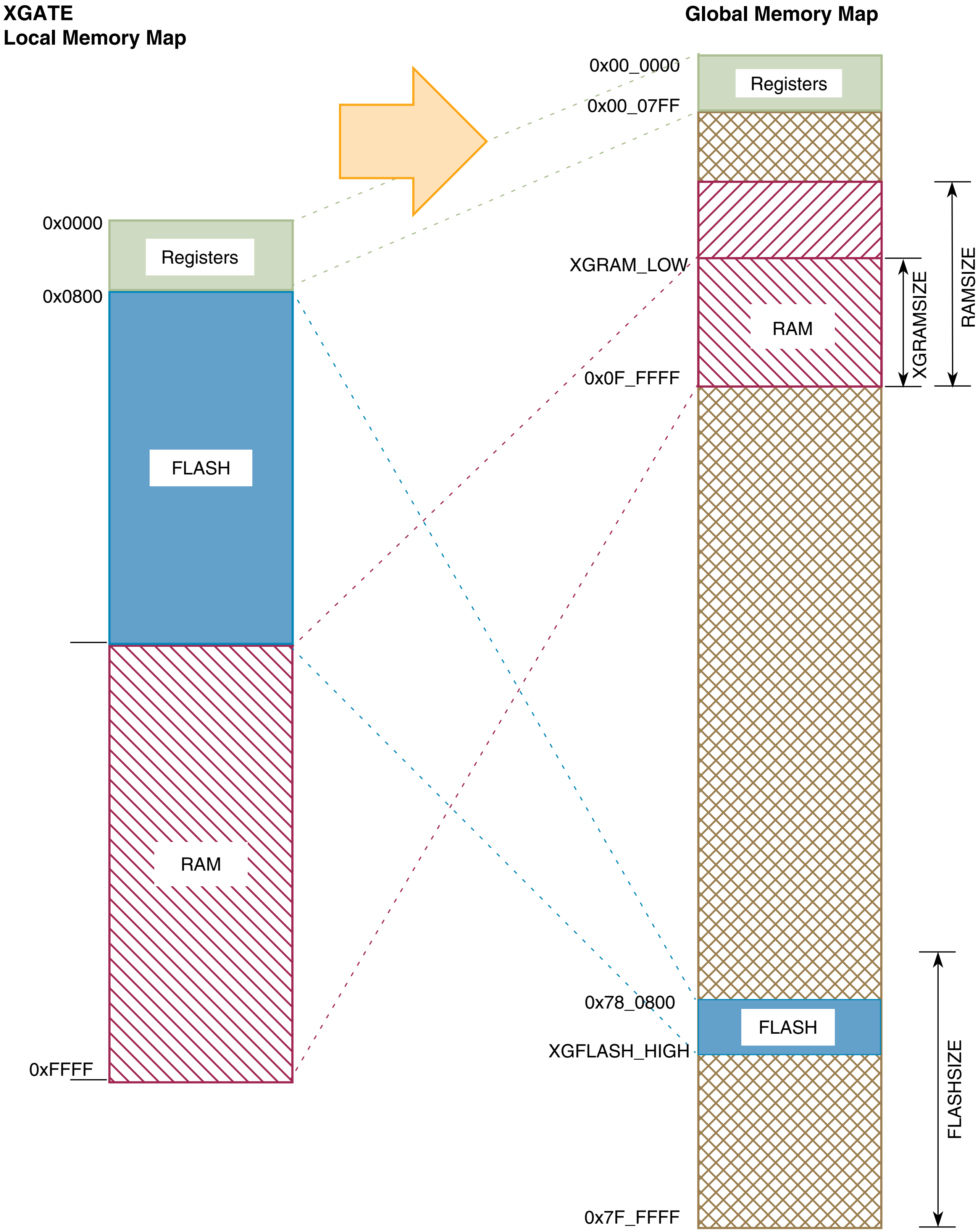


Figure 1-3. XGATE Global Address Mapping

1.1.6 Detailed Register Map

The detailed register map is listed in Appendix A.

1.1.7 Part ID Assignments

The part ID is located in two 8-bit registers PARTIDH and PARTIDL (addresses 0x001A and 0x001B). The read-only value is a unique part ID for each revision of the chip. Table 1-6 shows the assigned part ID number and Mask Set number.

The Version ID is a word located in a flash information row at 0x40_00E8. The version ID number indicates a specific version of internal NVM variables used to patch NVM errata.
The default is no patch (0xFFFF).

Table 1-6. Assigned Part ID Numbers

Device	Mask Set Number	Part ID[1]	Version ID
MC9S12XEP100	0M22E	0xCC80	0xFFFF
MC9S12XEP100	1M22E	0xCC80	0xFFFF
MC9S12XEP100	2M22E	0xCC82	0xFFFF
MC9S12XEP100	0M48H	0xCC90	0xFFFF
MC9S12XEP100	1M48H	0xCC91	0xFFFF
MC9S12XEP100	2M48H	0xCC92	0xFFFF
MC9S12XEP100	3M48H	0xCC93	0xFFFF
MC9S12XEP100, MC9S12XEP768[2]	4M48H	0xCC94	0xFFFF
MC9S12XEP100, MC9S12XEP768[2]	5M48H	0xCC94	0x0004
MC9S12XEP100, MC9S12XEP768[2]	0N35H	0xCC95	0xFFFF
MC9S12XEP100, MC9S12XEP768[2]	1N35H	0xCC95	0x0004
MC9S12XEQ512	0M25J	0xC480	0xFFFF
MC9S12XEQ512	1M25J	0xC481	0xFFFF
MC9S12XEQ512, MC9S12XET512	2M25J	0xC482	0xFFFF
MC9S12XEQ512, MC9S12XET512	3M25J	0xC482	0x0004
MC9S12XEQ384[3], MC9S12XEG384[3], MC9S12XES384[3]	2M25J	0xC482	0xFFFF
MC9S12XEQ384[3], MC9S12XEG384[3], MC9S12XES384[3]	3M25J	0xC482	0x0004
MC9S12XEQ512, MC9S12XET512	0M12S	0xC483	0xFFFF
MC9S12XEQ512, MC9S12XET512	1M12S	0xC483	0x0004
MC9S12XEQ384[3], MC9S12XEG384[3], MC9S12XES384[3]	0M12S	0xC483	0xFFFF
MC9S12XEQ384[3], MC9S12XEG384[3], MC9S12XES384[3]	1M12S	0xC483	0x0004
MC9S12XET256, MC9S12XEG256	0M53J	0xC080	0xFFFF
MC9S12XET256, MC9S12XEG256, MC9S12XEA256	1M53J	0xC081	0xFFFF
MC9S12XET256, MC9S12XEG256, MC9S12XEA256	2M53J	0xC081	0x0004
MC9S12XEG128[4], MC9S12XEA128[4]	1M53J	0xC081	0xFFFF
MC9S12XEG128[4], MC9S12XEA128[4]	2M53J	0xC081	0x0004
MC9S12XET256, MC9S12XEG256, MC9S12XEA256	0N36H	0xC082	0xFFFF
MC9S12XET256, MC9S12XEG256, MC9S12XEA256	1N36H	0xC082	0x0004
MC9S12XEG128[4], MC9S12XEA128[4]	0N36H	0xC082	0xFFFF
MC9S12XEG128[4], MC9S12XEA128[4]	1N36H	0xC082	0x0004

[1] The coding is as follows:
Bit 15-12: Major family identifier
Bit 11-6: Minor family identifier
Bit 5-4: Major mask set revision number including FAB transfers
Bit 3-0: Minor — non full — mask set revision

[2] Currently available as MC9S12XEP100 die only

[3] Currently available as MC9S12XEQ512 die only

[4] Currently available as MC9S12XET256 die only

1.2 Signal Description

This section describes signals that connect off-chip. It includes a pinout diagram, a table of signal properties, and detailed discussion of signals. It is built from the signal description sections of the Block User Guides of the individual IP blocks on the device.

1.2.1 Device Pinout

The MC9S12XE-Family offers pin-compatible packaged devices to assist with system development and accommodate expansion of the application.

NOTE

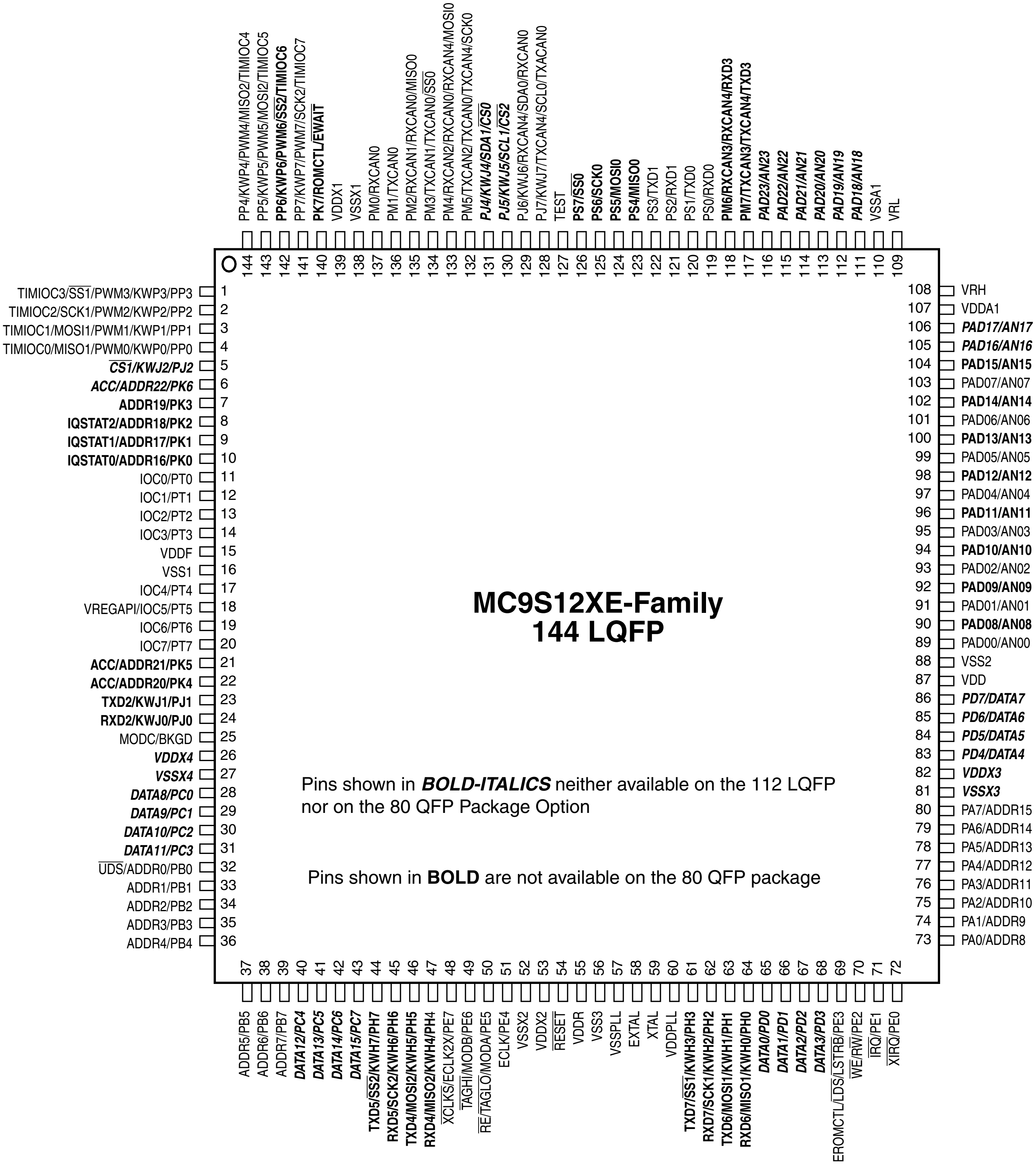
Smaller derivatives within the MC9S12XE-Family feature a subset of the listed modules. Refer to Appendix D Derivative Differences for more information about derivative device module subset and to Table 1-7. Port Availability by Package Option and Table 1-9. Pin-Out Summary for details of pins available in different package options.

The MC9S12XE-Family devices are offered in the following package options:

- 208-pin MAPBGA package with an external bus interface (address/data bus)
- 144-pin LQFP package with an external bus interface (address/data bus)
- 112-pin LQFP without external bus interface
- 80-pin QFP without external bus interface

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	N.C.	N.C.	PP7	PM0	PM1	PF5	PF3	PF1	PJ6	PS6	PS5	PS3	PM6	PAD19	N.C.	N.C.
B	N.C.	PP2	PP6	PF7	PF6	PF4	PF2	PF0	TEST	PS4	PS1	PAD23	PAD21	PAD18	PAD31	N.C.
C	PJ2	PP1	PP4	PP5	PK7	PM2	PM4	PJ5	PS7	PS2	PM7	PAD20	VRL	PAD16	PAD07	PAD14
D	PK1	PJ3	PP0	PP3	VDDX	PM3	PM5	PJ4	PJ7	VDDX	PS0	PAD22	VRH	PAD17	PAD30	PAD29
E	PK0	PK3	PK2	PK6									VSSA	PAD15	PAD06	PAD28
F	PR1	PR0	PT0	VDDX									VDDA	PAD05	PAD13	PAD27
G	PT2	PT3	PR2	PT1			VSSX	VSSX	VSSX	VSSX			VDDA	PAD12	PAD04	PAD11
H	PR3	PR4	PT4	VDDF			VSSX	VSSX	VSSX	VSSX			VSSA	PAD26	PAD03	PAD10
J	PT5	PR5	PT6	VSS1			VSSX	VSSX	VSSX	VSSX			VSS2	PAD09	PAD25	PAD02
K	PR6	PT7	PK4	PR7			VSSX	VSSX	VSSX	VSSX			VDD	PD7	PAD24	PAD01
L	PK5	PJ1	BKGD	VDDX									VDDX	PD4	PAD00	PAD08
M	PJ0	PC0	PB1	PC1									PA6	PA2	PD5	PD6
N	PC2	PC3	PB2	PC7	PL1	PE6	VDDX	VDDR	VSS3	PH3	PH1	VDDX	PE1	PA1	PA5	PA7
P	PB0	PB3	PB4	PC4	PL2	PL0	PE4	RESET	PL7	PL6	PH0	PE2	PE0	PA0	PA3	PA4
R	N.C.	PB5	PB6	PB7	PC6	PH6	PH4	PE5	VSS PLL	VDD PLL	PH2	PL4	PD1	PD3	PE3	N.C.
T	N.C.	N.C.	PC5	PL3	PH7	PH5	PE7	VSS PLL	EXTAL	XTAL	VDD PLL	PL5	PD0	PD2	N.C.	N.C.

Figure 1-4. - Pin Assignments, 208 MAPBGA Package



Pins shown in ***BOLD-ITALICS*** neither available on the 112 LQFP nor on the 80 QFP Package Option

Pins shown in **BOLD** are not available on the 80 QFP package

Figure 1-5. MC9S12XE-Family Pin Assignments 144-pin LQFP Package

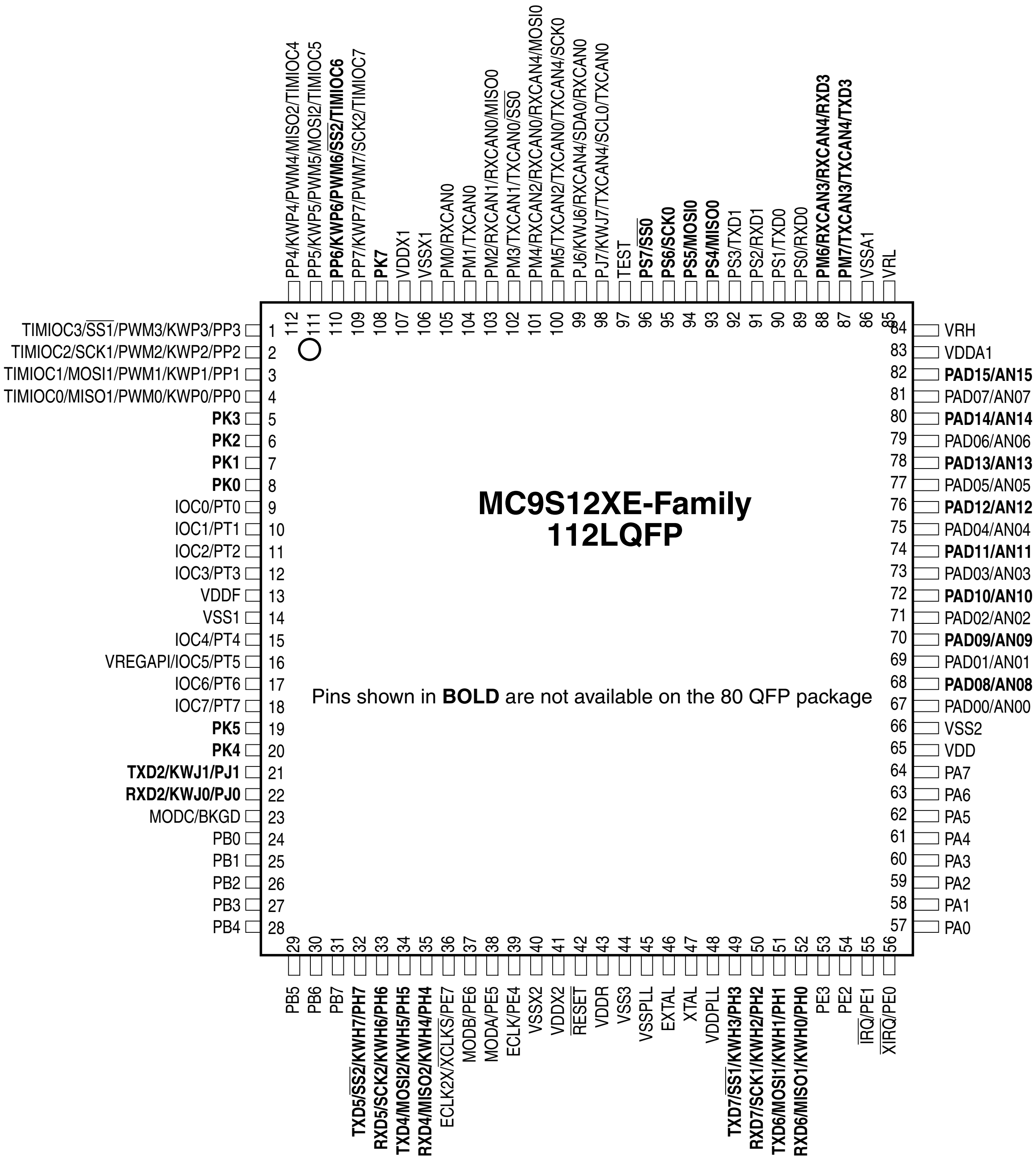


Figure 1-6. MC9S12XE-Family Pin Assignments 112-pin LQFP Package

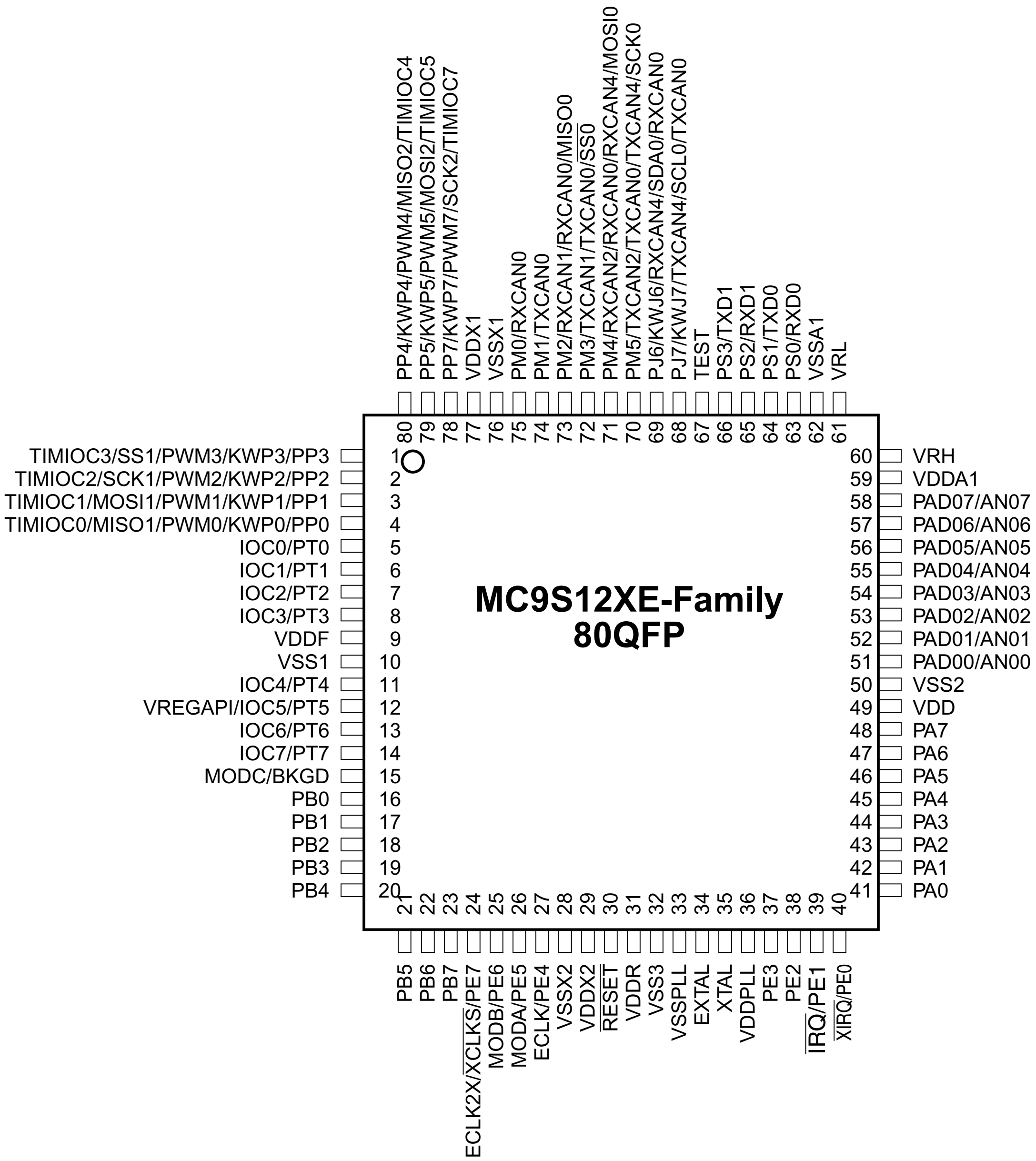


Figure 1-7. MC9S12XE-Family Pin Assignments 80-pin QFP Package

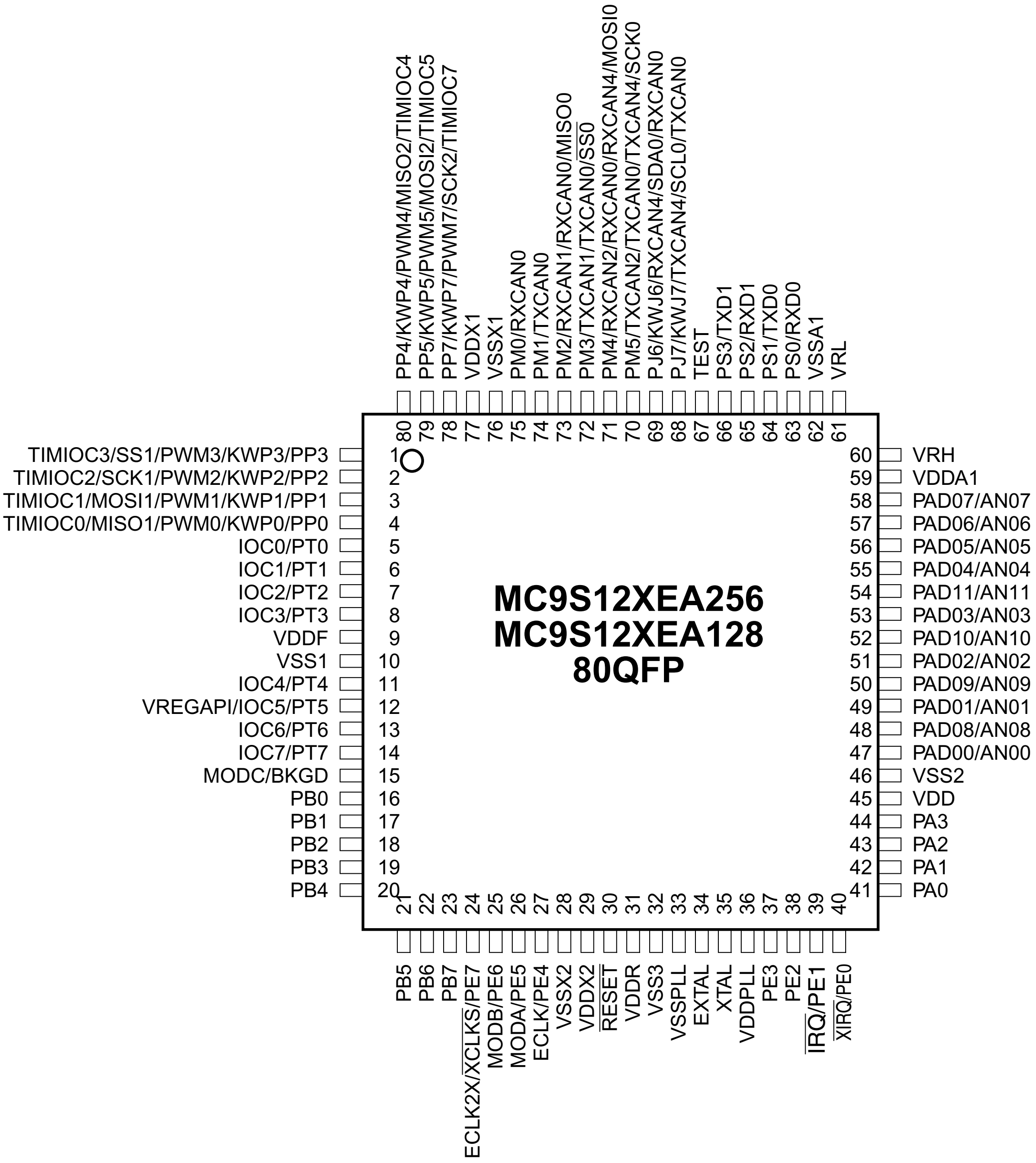


Figure 1-8. MC9S12XEA256/MC9S12XEA128 80-pin QFP Package Pin Assignment

NOTE

SPECIAL BOND-OUT TO PROVIDE ACCESS TO EXTRA ADC CHANNELS IN 80QFP. WARNING: NOT PIN-COMPATIBLE WITH REST OF FAMILY. THE MC9S12XET256 AND MC9S12XEG128 USE THE STANDARD 80QFP BOND-OUT, COMPATIBLE WITH OTHER FAMILY MEMBERS.

1.2.2 Pin Assignment Overview

Table 1-7 provides a summary of which Ports are available for each package option.

Routing of pin functions is summarized in Table 1-8.

Table 1-9 provides a pin out summary listing the availability of individual pins for each package option.

Table 1-10 provides a list of individual pin functionality

Table 1-7. Port Availability by Package Option

Port	208 MAPBGA	144 LQFP	112 LQFP	Standard 80 QFP	XEA256[1] 80 QFP
Port AD/ADC Channels	32/32	24/24	16/16	8/8	12/12
Port A pins	8	8	8	8	4
Port B pins	8	8	8	8	8
Port C pins	8	8	0	0	0
Port D pins	8	8	0	0	0
Port E pins inc. IRQ/XIRQ input only	8	8	8	8	8
Port F	8	0	0	0	0
Port H	8	8	8	0	0
Port J	8	7	4	2	2
Port K	8	8	7	0	0
Port L	8	0	0	0	0
Port M	8	8	8	6	6
Port P	8	8	8	7	7
Port R	8	0	0	0	0
Port S	8	8	8	4	4
Port T	8	8	8	8	8
Sum of Ports	**152**	**119**	**91**	**59**	**59**
I/O Power Pairs VDDX/VSSX	7/7	4/4	2/2	2/2	2/2

[1] The 9S12XEA256 is a special bondout for access to extra ADC channels in 80QFP.
Available in 80QFP / 256K memory size only. WARNING: NOT PIN-COMPATIBLE WITH REST OF FAMILY.
The 9S12XET256 is the standard 256K/80QFP bondout, compatible with other family members.

Table 1-8. Peripheral - Port Routing Options[1]

	CAN0	CAN1	CAN2	CAN3	CAN4	SCI0	SCI1	SCI2	SCI3	SCI4	SCI5	SCI6	SCI7	SPI0	SPI1	SPI2	IIC0	IIC1	$\overline{\text{CS0}}$	$\overline{\text{CS1}}$	$\overline{\text{CS2}}$	$\overline{\text{CS3}}$	TIM
PF[0]																			X				
PF[1]																				X			
PF[2]																					X		
PF[3]																						X	
PF[5:4]																	X						
PF[7:6]									X														
PH[1:0]												O			X								
PH[3:2]													O		X								
PH[5:4]										O						X							
PH[7:6]											O					X							
PJ[0]								O														O	
PJ[1]								O															
PJ[2]																				O			
PJ[3]																							
PJ[4]																		O	O				
PJ[5]																		O			O		
PJ[7:6]	X				O												O						
PL[1:0]										X													
PL[3:2]											X												
PL[5:4]												X											
PL[7:6]													X										
PM[1:0]	O																						
PM[3:2]	X	O												X									
PM[5:4]	X		O		X									X									
PM[7:6]				O	X				O														
PP[3:0]															O								X
PP[7:4]																O							X
PR[7:0]																							O

Table 1-8. Peripheral - Port Routing Options[1] (continued)

	CAN0	CAN1	CAN2	CAN3	CAN4	SCI0	SCI1	SCI2	SCI3	SCI4	SCI5	SCI6	SCI7	SPI0	SPI1	SPI2	IIC0	IIC1	$\overline{CS0}$	$\overline{CS1}$	$\overline{CS2}$	$\overline{CS3}$	TIM
PS[1:0]						O																	
PS[3:2]							O																
PS[7:4]														O									

[1] “O” denotes reset condition, “X” denotes a possible rerouting under software control

Table 1-9. Pin-Out Summary (Sheet 1 of 7)

208 MAPBGA	LQFP 144	LQFP 112	QFP[1] 80	Pin	2nd Func.	3rd Func.	4th Func.	5th Func.
D4	1	1	1	PP3	KWP3	PWM3	$\overline{SS1}$	TIMIOC3
B2	2	2	2	PP2	KWP2	PWM2	SCK1	TIMIOC2
C2	3	3	3	PP1	KWP1	PWM1	MOSI1	TIMIOC1
D3	4	4	4	PP0	KWP0	PWM0	MISO1	TIMIOC0
D2				PJ3	KWJ3			
C1	5			PJ2	KWJ2	$\overline{CS1}$		
E4	6			PK6	ADDR22	ACC2		
E2	7	5		PK3	ADDR19	IQSTAT3		
E3	8	6		PK2	ADDR18	IQSTAT2		
D1	9	7		PK1	ADDR17	IQSTAT1		
E1	10	8		PK0	ADDR16	IQSTAT0		
VDDX				VDDX7				
VSSX				VSSX7				
F3	11	9	5	PT0	IOC0			
F2				PR0	TIMIOC0			
G4	12	10	6	PT1	IOC1			
F1				PR1	TIMIOC1			
G1	13	11	7	PT2	IOC2			
G3				PR2	TIMIOC2			
G2	14	12	8	PT3	IOC3			
H1				PR3	TIMIOC3			
H4	15	13	9	VDDF				

Table 1-9. Pin-Out Summary (Sheet 2 of 7)

208 MAPBGA	LQFP 144	LQFP 112	QFP[1] 80	Pin	2nd Func.	3rd Func.	4th Func.	5th Func.
J4	16	14	10	VSS1				
H3	17	15	11	PT4	IOC4			
H2				PR4	TIMIOC4			
J1	18	16	12	PT5	IOC5	VREGAPI		
J2				PR5	TIMIOC5			
J3	19	17	13	PT6	IOC6			
K1				PR6	TIMIOC6			
K2	20	18	14	PT7	IOC7			
K4				PR7	TIMIOC7			
L1	21	19		PK5	ADDR21	ACC1		
K3	22	20		PK4	ADDR20	ACC0		
L2	23	21		PJ1	KWJ1	TXD2		
M1	24	22		PJ0	KWJ0	RXD2	$\overline{\text{CS3}}$	
L3	25	23	15	BKGD	MODC			
VDDX	26			VDDX4				
VSSX	27			VSSX4				
M2	28			PC0	DATA8			
M4	29			PC1	DATA9			
N1	30			PC2	DATA10			
N2	31			PC3	DATA11			
P1	32	24	16	PB0	ADDR0	IVD0	$\overline{\text{UDS}}$	
M3	33	25	17	PB1	ADDR1	IVD1		
N3	34	26	18	PB2	ADDR2	IVD2		
P2	35	27	19	PB3	ADDR3	IVD3		
P3	36	28	20	PB4	ADDR4	IVD4		
R2	37	29	21	PB5	ADDR5	IVD5		
R3	38	30	22	PB6	ADDR6	IVD6		
R4	39	31	23	PB7	ADDR7	IVD7		
P4	40			PC4	DATA12			

Table 1-9. Pin-Out Summary (Sheet 3 of 7)

208 MAPBGA	LQFP 144	LQFP 112	QFP[1] 80	Pin	2nd Func.	3rd Func.	4th Func.	5th Func.
T3	41			PC5	DATA13			
R5	42			PC6	DATA14			
N4	43			PC7	DATA15			
T4				PL3	TXD5			
T5	44	32		PH7	KWH7	$\overline{\text{SS2}}$	TXD5	
P5				PL2	RXD5			
R6	45	33		PH6	KWH6	SCK2	RXD5	
N5				PL1	TXD4			
T6	46	34		PH5	KWH5	MOSI2	TXD4	
P6				PL0	RXD4			
R7	47	35		PH4	KWH4	MISO2	RXD4	
T7	48	36	24	PE7	$\overline{\text{XCLKS}}$	ECLKX2		
N6	49	37	25	PE6	MODB	$\overline{\text{TAGHI}}$		
R8	50	38	26	PE5	MODA	$\overline{\text{TAGLO}}$	$\overline{\text{RE}}$	
P7	51	39	27	PE4	ECLK			
VSSX	52	40	28	VSSX2				
VDDX	53	41	29	VDDX2				
P8	54	42	30	$\overline{\text{RESET}}$				
N8	55	43	31	VDDR				
N9	56	44	32	VSS3				
R9/T8	57	45	33	VSSPLL				
T9	58	46	34	EXTAL				
T10	59	47	35	XTAL				
R10/T11	60	48	36	VDDPLL				
P9				PL7	TXD7			
N10	61	49		PH3	KWH3	$\overline{\text{SS1}}$	TXD7	
P10				PL6	RXD7			
R11	62	50		PH2	KWH2	SCK1	RXD7	
T12				PL5	TXD6			

Table 1-9. Pin-Out Summary (Sheet 4 of 7)

208 MAPBGA	LQFP 144	LQFP 112	QFP[(1)] 80	Pin	2nd Func.	3rd Func.	4th Func.	5th Func.
N11	63	51		PH1	KWH1	MOSI1	TXD6	
R12				PL4	RXD6			
P11	64	52		PH0	KWH0	MISO1	RXD6	
T13	65			PD0	DATA0			
R13	66			PD1	DATA1			
T14	67			PD2	DATA2			
R14	68			PD3	DATA3			
VDDX				VDDX5				
VSSX				VSSX5				
R15	69	53	37	PE3	$\overline{\text{LSTRB}}$	$\overline{\text{LDS}}$	EROMCTL	
P12	70	54	38	PE2	R$\overline{\text{W}}$	$\overline{\text{WE}}$		
N13	71	55	39	PE1	$\overline{\text{IRQ}}$			
P13	72	56	40	PE0	$\overline{\text{XIRQ}}$			
P14	73	57	41	PA0	ADDR8	IVD8		
N14	74	58	42	PA1	ADDR9	IVD9		
M14	75	59	43	PA2	ADDR10	IVD10		
P15	76	60	44	PA3	ADDR11	IVD11		
P16	77	61	45	PA4	ADDR12	IVD12		
N15	78	62	46	PA5	ADDR13	IVD13		
M13	79	63	47	PA6	ADDR14	IVD14		
N16	80	64	48	PA7	ADDR15	IVD15		
VSSX	81			VSSX3				
VDDX	82			VDDX3				
L14	83			PD4	DATA4			
M15	84			PD5	DATA5			
M16	85			PD6	DATA6			
K14	86			PD7	DATA7			
K13	87	65	49	VDD				
J13	88	66	50	VSS2				

Table 1-9. Pin-Out Summary (Sheet 5 of 7)

208 MAPBGA	LQFP 144	LQFP 112	QFP[1] 80	Pin	2nd Func.	3rd Func.	4th Func.	5th Func.
L15	89	67	51	PAD00	AN00			
L16	90	68		PAD08	AN08			
K15				PAD24	AN24			
K16	91	69	52	PAD01	AN01			
J14	92	70		PAD09	AN09			
J15				PAD25	AN25			
J16	93	71	53	PAD02	AN02			
H16	94	72		PAD10	AN10			
H14				PAD26	AN26			
H13				VSSA2				
G13				VDDA2				
H15	95	73	54	PAD03	AN03			
G16	96	74		PAD11	AN11			
F16				PAD27	AN27			
G15	97	75	55	PAD04	AN04			
G14	98	76		PAD12	AN12			
E16				PAD28	AN28			
F14	99	77	56	PAD05	AN05			
F15	100	78		PAD13	AN13			
D16				PAD29	AN29			
E15	101	79	57	PAD06	AN06			
C16	102	80		PAD14	AN14			
D15				PAD30	AN30			
C15	103	81	58	PAD07	AN07			
E14	104	82		PAD15	AN15			
B15				PAD31	AN31			
C14	105			PAD16	AN16			
D14	106			PAD17	AN17			
F13	107	83	59	VDDA1				

Table 1-9. Pin-Out Summary (Sheet 6 of 7)

208 MAPBGA	LQFP 144	LQFP 112	QFP[1] 80	Pin	2nd Func.	3rd Func.	4th Func.	5th Func.
D13	108	84	60	VRH				
C13	109	85	61	VRL				
E13	110	86	62	VSSA1				
B14	111			PAD18	AN18			
A14	112			PAD19	AN19			
C12	113			PAD20	AN20			
B13	114			PAD21	AN21			
D12	115			PAD22	AN22			
B12	116			PAD23	AN23			
C11	117	87		PM7	TXCAN3	TXCAN4	TXD3	
A13	118	88		PM6	RXCAN3	RXCAN4	RXD3	
D11	119	89	63	PS0	RXD0			
B11	120	90	64	PS1	TXD0			
C10	121	91	65	PS2	RXD1			
A12	122	92	66	PS3	TXD1			
VSSX				VSSX6				
VDDX				VDDX6				
B10	123	93		PS4	MISO0			
A11	124	94		PS5	MOSI0			
A10	125	95		PS6	SCK0			
C9	126	96		PS7	$\overline{\text{SS0}}$			
B9	127	97	67	TEST				
D9	128	98	68	PJ7	KWJ7	TXCAN4	SCL0	TXCAN0
A9	129	99	69	PJ6	KWJ6	RXCAN4	SDA0	RXCAN0
C8	130			PJ5	KWJ5	SCL1	$\overline{\text{CS2}}$	
B8				PF0	$\overline{\text{CS0}}$			
D8	131			PJ4	KWJ4	SDA1	$\overline{\text{CS0}}$	
A8				PF1	$\overline{\text{CS1}}$			
D7	132	100	70	PM5	TXCAN2	TXCAN0	TXCAN4	SCK0

Table 1-9. Pin-Out Summary (Sheet 7 of 7)

208 MAPBGA	LQFP 144	LQFP 112	QFP[1] 80	Pin	2nd Func.	3rd Func.	4th Func.	5th Func.
B7				PF2	$\overline{\text{CS2}}$			
C7	133	101	71	PM4	RXCAN2	RXCAN0	RXCAN4	MOSI0
A7				PF3	$\overline{\text{CS3}}$			
D6	134	102	72	PM3	TXCAN1	TXCAN0	$\overline{\text{SS0}}$	
B6				PF4	SDA0			
C6	135	103	73	PM2	RXCAN1	RXCAN0	MISO0	
A6				PF5	SCL0			
A5	136	104	74	PM1	TXCAN0			
B5				PF6	RXD3			
A4	137	105	75	PM0	RXCAN0			
B4				PF7	TXD3			
VSSX	138	106	76	VSSX1				
VDDX	139	107	77	VDDX1				
C5	140	108		PK7	ROMCTL	$\overline{\text{EWAIT}}$		
A3	141	109	78	PP7	KWP7	PWM7	SCK2	TIMIOC7
B3	142	110		PP6	KWP6	PWM6	$\overline{\text{SS2}}$	TIMIOC6
C4	143	111	79	PP5	KWP5	PWM5	MOSI2	TIMIOC5
C3	144	112	80	PP4	KWP4	PWM4	MISO2	TIMIOC4

[1] Standard 80QFP only. NOTE that XEA256 80QFP is not compatible

Table 1-10. Signal Properties Summary (Sheet 1 of 4)

Pin Name Function 1	Pin Name Function 2	Pin Name Function 3	Pin Name Function 4	Pin Name Function 5	Power Supply	Internal Pull Resistor		Description
						CTRL	Reset State	
EXTAL	—	—	—	—	V_{DDPLL}	NA	NA	Oscillator pins
XTAL	—	—	—	—	V_{DDPLL}	NA	NA	
$\overline{\text{RESET}}$	—	—	—	—	V_{DDX}	PULLUP		External reset
TEST	—	—	—	—	N.A.	$\overline{\text{RESET}}$ pin	DOWN	Test input
BKGD	MODC	—	—	—	V_{DDX}	Always on	Up	Background debug
PAD[31:16]	AN[31:16]	—	—	—	V_{DDA}	PER0AD1 PER1AD1	Disabled	Port AD inputs of ATD1, analog inputs of ATD1
PAD[15:0]	AN[15:0]	—	—	—	V_{DDA}	PER0AD0 PER1AD0	Disabled	Port AD inputs of ATD0, analog inputs of ATD0
PA[7:0]	ADDR[15:8]	IVD[15:8]	—	—	V_{DDX}	PUCR	Disabled	Port A I/O, address bus, internal visibility data
PB[7:1]	ADDR[7:1]	IVD[7:0]	—	—	V_{DDX}	PUCR	Disabled	Port B I/O, address bus, internal visibility data
PB0	ADDR0	$\overline{\text{UDS}}$			V_{DDX}	PUCR	Disabled	Port B I/O, address bus, upper data strobe
PC[7:0]	DATA[15:8]	—	—	—	V_{DDX}	PUCR	Disabled	Port C I/O, data bus
PD[7:0]	DATA[7:0]	—	—	—	V_{DDX}	PUCR	Disabled	Port D I/O, data bus
PE7	ECLKX2	$\overline{\text{XCLKS}}$	—	—	V_{DDX}	PUCR	Up	Port E I/O, system clock output, clock select
PE6	$\overline{\text{TAGHI}}$	MODB	—	—	V_{DDX}	While $\overline{\text{RESET}}$ pin is low: down		Port E I/O, tag high, mode input
PE5	$\overline{\text{RE}}$	MODA	$\overline{\text{TAGLO}}$	—	V_{DDX}	While $\overline{\text{RESET}}$ pin is low: down		Port E I/O, read enable, mode input, tag low input
PE4	ECLK	—	—	—	V_{DDX}	PUCR	Up	Port E I/O, bus clock output
PE3	$\overline{\text{LSTRB}}$	$\overline{\text{LDS}}$	EROMCTL	—	V_{DDX}	PUCR	Up	Port E I/O, low byte data strobe, EROMON control
PE2	R/$\overline{\text{W}}$	$\overline{\text{WE}}$	—	—	V_{DDX}	PUCR	Up	Port E I/O, read/write
PE1	$\overline{\text{IRQ}}$	—	—	—	V_{DDX}	PUCR	Up	Port E Input, maskable interrupt
PE0	$\overline{\text{XIRQ}}$	—	—	—	V_{DDX}	PUCR	Up	Port E input, non-maskable interrupt
PF7	TXD3	—	—	—	V_{DDX}	PERF/ PPSF	Up	Port F I/O, interrupt, TXD of SCI3
PF6	RXD3	—	—	—	V_{DDX}	PERF/ PPSF	Up	Port F I/O, interrupt, RXD of SCI3
PF5	SCL0	—	—	—	V_{DDX}	PERF/ PPSF	Up	Port F I/O, interrupt, SCL of IIC0
PF4	SDA0	—	—	—	V_{DDX}	PERF/ PPSF	Up	Port F I/O, interrupt, SDA of IIC0
PF3	$\overline{\text{CS3}}$	—	—	—	V_{DDX}	PERF/ PPSF	Up	Port F I/O, interrupt, chip select 3

Table 1-10. Signal Properties Summary (Sheet 2 of 4)

Pin Name Function 1	Pin Name Function 2	Pin Name Function 3	Pin Name Function 4	Pin Name Function 5	Power Supply	Internal Pull Resistor		Description
						CTRL	Reset State	
PF2	$\overline{\text{CS2}}$	—	—	—	V_{DDX}	PERF/ PPSF	Up	Port F I/O, interrupt, chip select 2
PF1	$\overline{\text{CS1}}$	—	—	—	V_{DDX}	PERF/ PPSF	Up	Port F I/O, interrupt, chip select 1
PF0	$\overline{\text{CS0}}$	—	—	—	V_{DDX}	PERF/ PPSF	Up	Port F I/O, interrupt, chip select 0
PH7	KWH7	$\overline{\text{SS2}}$	TXD5	—	V_{DDX}	PERH/ PPSH	Disabled	Port H I/O, interrupt, $\overline{\text{SS}}$ of SPI2, TXD of SCI5
PH6	KWH6	SCK2	RXD5	—	V_{DDX}	PERH/ PPSH	Disabled	Port H I/O, interrupt, SCK of SPI2, RXD of SCI5
PH5	KWH5	MOSI2	TXD4	—	V_{DDX}	PERH/ PPSH	Disabled	Port H I/O, interrupt, MOSI of SPI2, TXD of SCI4
PH4	KWH4	MISO2	RXD4	—	V_{DDX}	PERH/PPSH	Disabled	Port H I/O, interrupt, MISO of SPI2, RXD of SCI4
PH3	KWH3	$\overline{\text{SS1}}$	TXD7	—	V_{DDX}	PERH/PPSH	Disabled	Port H I/O, interrupt, $\overline{\text{SS}}$ of SPI1
PH2	KWH2	SCK1	RXD7	—	V_{DDX}	PERH/PPSH	Disabled	Port H I/O, interrupt, SCK of SPI1
PH1	KWH1	MOSI1	TXD6	—	V_{DDX}	PERH/PPSH	Disabled	Port H I/O, interrupt, MOSI of SPI1
PH0	KWH0	MISO1	RXD6	—	V_{DDX}	PERH/PPSH	Disabled	Port H I/O, interrupt, MISO of SPI1
PJ7	KWJ7	TXCAN4	SCL0	TXCAN0	V_{DDX}	PERJ/ PPSJ	Up	Port J I/O, interrupt, TX of CAN4, SCL of IIC0, TX of CAN0
PJ6	KWJ6	RXCAN4	SDA0	RXCAN0	V_{DDX}	PERJ/ PPSJ	Up	Port J I/O, interrupt, RX of CAN4, SDA of IIC0, RX of CAN0
PJ5	KWJ5	SCL1	$\overline{\text{CS2}}$	—	V_{DDX}	PERJ/ PPSJ	Up	Port J I/O, interrupt, SCL of IIC1, chip select 2
PJ4	KWJ4	SDA1	$\overline{\text{CS0}}$	—	V_{DDX}	PERJ/ PPSJ	Up	Port J I/O, interrupt, SDA of IIC1, chip select 0
PJ3	KWJ3	—	—	—	V_{DDX}	PERJ/ PPSJ	Up	Port J I/O, interrupt,
PJ2	KWJ2	$\overline{\text{CS1}}$	—	—	V_{DDX}	PERJ/ PPSJ	Up	Port J I/O, interrupt, chip select 1
PJ1	KWJ1	TXD2	—	—	V_{DDX}	PERJ/ PPSJ	Up	Port J I/O, interrupt, TXD of SCI2
PJ0	KWJ0	RXD2	$\overline{\text{CS3}}$	—	V_{DDX}	PERJ/ PPSJ	Up	Port J I/O, interrupt, RXD of SCI2
PK7	$\overline{\text{EWAIT}}$	ROMCTL	—	—	V_{DDX}	PUCR	Up	Port K I/O, EWAIT input, ROM on control

Table 1-10. Signal Properties Summary (Sheet 3 of 4)

Pin Name Function 1	Pin Name Function 2	Pin Name Function 3	Pin Name Function 4	Pin Name Function 5	Power Supply	Internal Pull Resistor		Description
						CTRL	Reset State	
PK[6:4]	ADDR [22:20]	ACC[2:0]	—	—	V_{DDX}	PUCR	Up	Port K I/O, extended addresses, access source for external access
PK[3:0]	ADDR [19:16]	IQSTAT [3:0]	—	—	V_{DDX}	PUCR	Up	Extended address, PIPE status
PL7	TXD7	—	—	—	V_{DDX}	PERL/ PPSL	Up	Port L I/O, TXD of SCI7
PL6	RXD7	—	—	—	V_{DDX}	PERL/ PPSL	Up	Port LI/O, RXD of SCI7
PL5	TXD6	—	—	—	V_{DDX}	PERL/ PPSL	Up	Port L I/O, TXD of SCI6
PL4	RXD6	—	—	—	V_{DDX}	PERL/ PPSL	Up	Port LI/O, RXD of SCI6
PL3	TXD5	—	—	—	V_{DDX}	PERL/ PPSL	Up	Port L I/O, TXD of SCI5
PL2	RXD5	—	—	—	V_{DDX}	PERL/ PPSL	Up	Port LI/O, RXD of SCI5
PL1	TXD4	—	—	—	V_{DDX}	PERL/ PPSL	Up	Port L I/O, TXD of SCI4
PL0	RXD4	—	—	—	V_{DDX}	PERL/ PPSL	Up	Port LI/O, RXD of SCI4
PM7	TXCAN3	TXD3	TXCAN4	—	V_{DDX}	PERM/ PPSM	Disabled	Port M I/O, TX of CAN3 and CAN4, TXD of SCI3
PM6	RXCAN3	RXD3	RXCAN4	—	V_{DDX}	PERM/PPSM	Disabled	Port M I/O RX of CAN3 and CAN4, RXD of SCI3
PM5	TXCAN2	TXCAN0	TXCAN4	SCK0	V_{DDX}	PERM/PPSM	Disabled	Port M I/OCAN0, CAN2, CAN4, SCK of SPI0
PM4	RXCAN2	RXCAN0	RXCAN4	MOSI0	V_{DDX}	PERM/PPSM	Disabled	Port M I/O, CAN0, CAN2, CAN4, MOSI of SPI0
PM3	TXCAN1	TXCAN0	$\overline{\text{SS0}}$	—	V_{DDX}	PERM/PPSM	Disabled	Port M I/O TX of CAN1, CAN0, $\overline{\text{SS}}$ of SPI0
PM2	RXCAN1	RXCAN0	MISO0	—	V_{DDX}	PERM/PPSM	Disabled	Port M I/O, RX of CAN1, CAN0, MISO of SPI0
PM1	TXCAN0		—	—	V_{DDX}	PERM/PPSM	Disabled	Port M I/O, TX of CAN0
PM0	RXCAN0		—	—	V_{DDX}	PERM/PPSM	Disabled	Port M I/O, RX of CAN0
PP7	KWP7	PWM7	SCK2	TIMIOC7	V_{DDX}	PERP/ PPSP	Disabled	Port P I/O, interrupt, channel 7 of PWM/TIM , SCK of SPI2
PP6	KWP6	PWM6	$\overline{\text{SS2}}$	TIMIOC6	V_{DDX}	PERP/ PPSP	Disabled	Port P I/O, interrupt, channel 6 of PWM/TIM, $\overline{\text{SS}}$ of SPI2
PP5	KWP5	PWM5	MOSI2	TIMIOC5	V_{DDX}	PERP/ PPSP	Disabled	Port P I/O, interrupt, channel 5 of PWM/TIM, MOSI of SPI2

Table 1-10. Signal Properties Summary (Sheet 4 of 4)

Pin Name Function 1	Pin Name Function 2	Pin Name Function 3	Pin Name Function 4	Pin Name Function 5	Power Supply	Internal Pull Resistor		Description
						CTRL	Reset State	
PP4	KWP4	PWM4	MISO2	TIMIOC4	V_{DDX}	PERP/ PPSP	Disabled	Port P I/O, interrupt, channel 4 of PWM/TIM, MISO2 of SPI2
PP3	KWP3	PWM3	$\overline{\text{SS1}}$	TIMIOC3	V_{DDX}	PERP/ PPSP	Disabled	Port P I/O, interrupt, channel 3 of PWM/TIM, $\overline{\text{SS}}$ of SPI1
PP2	KWP2	PWM2	SCK1	TIMIOC2	V_{DDX}	PERP/ PPSP	Disabled	Port P I/O, interrupt, channel 2 of PWM/TIM, SCK of SPI1
PP1	KWP1	PWM1	MOSI1	TIMIOC1	V_{DDX}	PERP/ PPSP	Disabled	Port P I/O, interrupt, channel 1 of PWM/TIM, MOSI of SPI1
PP0	KWP0	PWM0	MISO1	TIMIOC0	V_{DDX}	PERP/ PPSP	Disabled	Port P I/O, interrupt, channel 0 of PWM/TIM, MISO2 of SPI1
PR[7:0]	TIMIOC [7:0]	—	—	—	V_{DDX}	PERR/ PPSR	Disabled	Port RI/O, TIM channels
PS7	$\overline{\text{SS0}}$	—	—	—	V_{DDX}	PERS/ PPSS	Up	Port S I/O, $\overline{\text{SS}}$ of SPI0
PS6	SCK0	—	—	—	V_{DDX}	PERS/ PPSS	Up	Port S I/O, SCK of SPI0
PS5	MOSI0	—	—	—	V_{DDX}	PERS/ PPSS	Up	Port S I/O, MOSI of SPI0
PS4	MISO0	—	—	—	V_{DDX}	PERS/ PPSS	Up	Port S I/O, MISO of SPI0
PS3	TXD1	—	—	—	V_{DDX}	PERS/ PPSS	Up	Port S I/O, TXD of SCI1
PS2	RXD1	—	—	—	V_{DDX}	PERS/ PPSS	Up	Port S I/O, RXD of SCI1
PS1	TXD0	—	—	—	V_{DDX}	PERS/ PPSS	Up	Port S I/O, TXD of SCI0
PS0	RXD0	—	—	—	V_{DDX}	PERS/ PPSS	Up	Port S I/O, RXD of SCI0
PT[7:6]	IOC[7:6]	—	—	—	V_{DDX}	PERT/ PPST	Disabled	Port T I/O, ECT channels
PT[5]	IOC[5]	VREGAPI	—	—	V_{DDX}	PERT/ PPST	Disabled	Port T I/O, ECT channels
PT[4:0]	IOC[4:0]	—	—	—	V_{DDX}	PERT/ PPST	Disabled	Port T I/O, ECT channels

1.2.3 Detailed Signal Descriptions

NOTE

> The pin list of the largest package version of each MC9S12XE-Family derivative gives the complete of interface signals that also exist on smaller package options, although some of them are not bonded out. For devices assembled in smaller packages all non-bonded out pins should be configured as outputs after reset in order to avoid current drawn from floating inputs. Refer to Table 1-10 for affected pins. Particular attention is drawn to Port R, which does not have enabled pull-up/pull-down devices coming out of reset.

1.2.3.1 EXTAL, XTAL — Oscillator Pins

EXTAL and XTAL are the crystal driver and external clock pins. On reset all the device clocks are derived from the EXTAL input frequency. XTAL is the oscillator output.

1.2.3.2 $\overline{\text{RESET}}$ — External Reset Pin

The $\overline{\text{RESET}}$ pin is an active low bidirectional control signal. It acts as an input to initialize the MCU to a known start-up state. As an output it is driven low to indicate when any internal MCU reset source triggers. The $\overline{\text{RESET}}$ pin has an internal pull-up device.

1.2.3.3 TEST — Test Pin

This input only pin is reserved for test. This pin has a pull-down device.

NOTE

> The TEST pin must be tied to V_{SS} in all applications.

1.2.3.4 BKGD / MODC — Background Debug and Mode Pin

The BKGD/MODC pin is used as a pseudo-open-drain pin for the background debug communication. It is used as a MCU operating mode select pin during reset. The state of this pin is latched to the MODC bit at the rising edge of $\overline{\text{RESET}}$. The BKGD pin has a pull-up device.

1.2.3.5 PAD[15:0] / AN[15:0] — Port AD Input Pins of ATD0

PAD[15:0] are general-purpose input or output pins and analog inputs AN[15:0] of the analog-to-digital converter ATD0.

1.2.3.6 PAD[31:16] / AN[31:16] — Port AD Input Pins of ATD1

PAD[31:16] are general-purpose input or output pins and analog inputs AN[31:16] of the analog-to-digital converter ATD1.

1.2.3.7 PA[7:0] / ADDR[15:8] / IVD[15:8] — Port A I/O Pins

PA[7:0] are general-purpose input or output pins. In MCU expanded modes of operation, these pins are used for the external address bus. In MCU emulation modes of operation, these pins are used for external address bus and internal visibility read data.

1.2.3.8 PB[7:1] / ADDR[7:1] / IVD[7:1] — Port B I/O Pins

PB[7:1] are general-purpose input or output pins. In MCU expanded modes of operation, these pins are used for the external address bus. In MCU emulation modes of operation, these pins are used for external address bus and internal visibility read data.

1.2.3.9 PB0 / ADDR0 / $\overline{\text{UDS}}$ / IVD[0] — Port B I/O Pin 0

PB0 is a general-purpose input or output pin. In MCU expanded modes of operation, this pin is used for the external address bus ADDR0 or as upper data strobe signal. In MCU emulation modes of operation, this pin is used for external address bus ADDR0 and internal visibility read data IVD0.

1.2.3.10 PC[7:0] / DATA [15:8] — Port C I/O Pins

PC[7:0] are general-purpose input or output pins. In MCU expanded modes of operation, these pins are used for the external data bus.

The input voltage thresholds for PC[7:0] can be configured to reduced levels, to allow data from an external 3.3-V peripheral to be read by the MCU operating at 5.0 V. The input voltage thresholds for PC[7:0] are configured to reduced levels out of reset in expanded and emulation modes. The input voltage thresholds for PC[7:0] are configured to 5-V levels out of reset in normal modes.

1.2.3.11 PD[7:0] / DATA [7:0] — Port D I/O Pins

PD[7:0] are general-purpose input or output pins. In MCU expanded modes of operation, these pins are used for the external data bus.

The input voltage thresholds for PD[7:0] can be configured to reduced levels, to allow data from an external 3.3-V peripheral to be read by the MCU operating at 5.0 V. The input voltage thresholds for PD[7:0] are configured to reduced levels out of reset in expanded and emulation modes. The input voltage thresholds for PC[7:0] are configured to 5-V levels out of reset in normal modes.

1.2.3.12 PE7 / ECLKX2 / $\overline{\text{XCLKS}}$ — Port E I/O Pin 7

PE7 is a general-purpose input or output pin. ECLKX2 is a free running clock of twice the internal bus frequency, available by default in emulation modes and when enabled in other modes. The $\overline{\text{XCLKS}}$ is an input signal which controls whether a crystal in combination with the internal loop controlled Pierce oscillator is used or whether full swing Pierce oscillator/external clock circuitry is used (refer to Oscillator Configuration). An internal pullup is enabled during reset.

1.2.3.13 PE6 / MODB / $\overline{\text{TAGHI}}$ — Port E I/O Pin 6

PE6 is a general-purpose input or output pin. It is used as a MCU operating mode select pin during reset. The state of this pin is latched to the MODB bit at the rising edge of $\overline{\text{RESET}}$. This pin is an input with a pull-down device which is only active when $\overline{\text{RESET}}$ is low. $\overline{\text{TAGHI}}$ is used to tag the high half of the instruction word being read into the instruction queue.

The input voltage threshold for PE6 can be configured to reduced levels, to allow data from an external 3.3-V peripheral to be read by the MCU operating at 5.0 V. The input voltage threshold for PE6 is configured to reduced levels out of reset in expanded and emulation modes.

1.2.3.14 PE5 / MODA / $\overline{\text{TAGLO}}$ / $\overline{\text{RE}}$ — Port E I/O Pin 5

PE5 is a general-purpose input or output pin. It is used as an MCU operating mode select pin during reset. The state of this pin is latched to the MODA bit at the rising edge of $\overline{\text{RESET}}$. This pin is shared with the read enable $\overline{\text{RE}}$ output. This pin is an input with a pull-down device which is only active when $\overline{\text{RESET}}$ is low. $\overline{\text{TAGLO}}$ is used to tag the low half of the instruction word being read into the instruction queue.

The input voltage threshold for PE5 can be configured to reduced levels, to allow data from an external 3.3-V peripheral to be read by the MCU operating at 5.0 V. The input voltage threshold for PE5 is configured to reduced levels out of reset in expanded and emulation modes.

1.2.3.15 PE4 / ECLK — Port E I/O Pin 4

PE4 is a general-purpose input or output pin. It can be configured to drive the internal bus clock ECLK. ECLK can be used as a timing reference. The ECLK output has a programmable prescaler.

1.2.3.16 PE3 / $\overline{\text{LSTRB}}$ / $\overline{\text{LDS}}$ / EROMCTL— Port E I/O Pin 3

PE3 is a general-purpose input or output pin. In MCU expanded modes of operation, $\overline{\text{LSTRB}}$ or $\overline{\text{LDS}}$ can be used for the low byte strobe function to indicate the type of bus access. At the rising edge of $\overline{\text{RESET}}$ the state of this pin is latched to the EROMON bit.

1.2.3.17 PE2 / R/$\overline{\text{W}}$ / $\overline{\text{WE}}$—Port E I/O Pin 2

PE2 is a general-purpose input or output pin. In MCU expanded modes of operations, this pin drives the read/write output signal or write enable output signal for the external bus. It indicates the direction of data on the external bus.

1.2.3.18 PE1 / $\overline{\text{IRQ}}$ — Port E Input Pin 1

PE1 is a general-purpose input pin and the maskable interrupt request input that provides a means of applying asynchronous interrupt requests. This will wake up the MCU from stop or wait mode.

1.2.3.19 PE0 / $\overline{\text{XIRQ}}$ — Port E Input Pin 0

PE0 is a general-purpose input pin and the non-maskable interrupt request input that provides a means of applying asynchronous interrupt requests. This will wake up the MCU from stop or wait mode. The XIRQ interrupt is level sensitive and active low. As XIRQ is level sensitive, while this pin is low the MCU will not enter STOP mode.

1.2.3.20 PF7 / TXD3 — Port F I/O Pin 7

PF7 is a general-purpose input or output pin. It can be configured as the transmit pin TXD of serial communication interface 3 (SCI3).

1.2.3.21 PF6 / RXD3 — Port F I/O Pin 6

PF6 is a general-purpose input or output pin. It can be configured as the transmit pin RXD of serial communication interface 3 (SCI3).

1.2.3.22 PF5 / SCL0 — Port F I/O Pin 5

PF5 is a general-purpose input or output pin. It can be configured as the serial clock pin SCL of the IIC0 module.

1.2.3.23 PF4 / SDA0 — Port F I/O Pin 4

PF4 is a general-purpose input or output pin. It can be configured as the serial data pin SDA of the IIC0 module.

1.2.3.24 PF[3:0] / $\overline{\text{CS[3:0]}}$ — Port F I/O Pins 3 to 0

PF[3:0] are a general-purpose input or output pins. They can be configured as chip select outputs [3:0].

1.2.3.25 PH7 / KWH7 / $\overline{\text{SS2}}$ / TXD5 — Port H I/O Pin 7

PH7 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as slave select pin $\overline{\text{SS}}$ of the serial peripheral interface 2 (SPI2). It can be configured as the transmit pin TXD of serial communication interface 5 (SCI5).

1.2.3.26 PH6 / KWH6 / SCK2 / RXD5 — Port H I/O Pin 6

PH6 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as serial clock pin SCK of the serial peripheral interface 2 (SPI2). It can be configured as the receive pin (RXD) of serial communication interface 5 (SCI5).

1.2.3.27 PH5 / KWH5 / MOSI2 / TXD4 — Port H I/O Pin 5

PH5 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as master output (during master mode) or slave input pin (during slave mode) MOSI of the serial peripheral interface 2 (SPI2). It can be configured as the transmit pin TXD of serial communication interface 4 (SCI4).

1.2.3.28 PH4 / KWH4 / MISO2 / RXD4 — Port H I/O Pin 4

PH4 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as master input (during master mode) or slave output (during slave mode) pin MISO of the serial peripheral interface 2 (SPI2). It can be configured as the receive pin RXD of serial communication interface 4 (SCI4).

1.2.3.29 PH3 / KWH3 / $\overline{\text{SS1}}$ — Port H I/O Pin 3

PH3 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as slave select pin $\overline{\text{SS}}$ of the serial peripheral interface 1 (SPI1). It can also be configured as the transmit pin TXD of serial communication interface 7 (SCI7).

1.2.3.30 PH2 / KWH2 / SCK1 — Port H I/O Pin 2

PH2 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as serial clock pin SCK of the serial peripheral interface 1 (SPI1). It can be configured as the receive pin RXD of serial communication interface 7 (SCI7).

1.2.3.31 PH1 / KWH1 / MOSI1 — Port H I/O Pin 1

PH1 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as master output (during master mode) or slave input pin (during slave mode) MOSI of the serial peripheral interface 1 (SPI1). It can also be configured as the transmit pin TXD of serial communication interface 6 (SCI6).

1.2.3.32 PH0 / KWH0 / MISO1 — Port H I/O Pin 0

PH0 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as master input (during master mode) or slave output (during slave mode) pin MISO of the serial peripheral interface 1 (SPI1). It can be configured as the receive pin RXD of serial communication interface 6 (SCI6).

1.2.3.33 PJ7 / KWJ7 / TXCAN4 / SCL0 / TXCAN0— PORT J I/O Pin 7

PJ7 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as the transmit pin TXCAN for the scalable controller area network controller 0 or 4 (CAN0 or CAN4) or as the serial clock pin SCL of the IIC0 module.

1.2.3.34 PJ6 / KWJ6 / RXCAN4 / SDA0 / RXCAN0 — PORT J I/O Pin 6

PJ6 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as the receive pin RXCAN for the scalable controller area network controller 0 or 4 (CAN0 or CAN4) or as the serial data pin SDA of the IIC0 module.

1.2.3.35 PJ5 / KWJ5 / SCL1 / $\overline{\text{CS2}}$ — PORT J I/O Pin 5

PJ5 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as the serial clock pin SCL of the IIC1 module. It can be also configured as chip-select output 2.

1.2.3.36 PJ4 / KWJ4 / SDA1 / $\overline{\text{CS0}}$ — PORT J I/O Pin 4

PJ4 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as the serial data pin SDA of the IIC1 module. It can also be configured as chip-select output.

1.2.3.37 PJ3 / KWJ3 — PORT J I/O Pin 3

PJ3 is a general-purpose input or output pins. It can be configured as a keypad wakeup input.

1.2.3.38 PJ2 / KWJ2 / $\overline{\text{CS1}}$ — PORT J I/O Pin 2

PJ2 is a general-purpose input or output pins. It can be configured as a keypad wakeup input. It can also be configured as chip-select output.

1.2.3.39 PJ1 / KWJ1 / TXD2 — PORT J I/O Pin 1

PJ1 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as the transmit pin TXD of the serial communication interface 2 (SCI2).

1.2.3.40 PJ0 / KWJ0 / RXD2 / $\overline{\text{CS3}}$ — PORT J I/O Pin 0

PJ0 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as the receive pin RXD of the serial communication interface 2 (SCI2).It can also be configured as chip-select output 3.

1.2.3.41 PK7 / $\overline{\text{EWAIT}}$ / ROMCTL — Port K I/O Pin 7

PK7 is a general-purpose input or output pin. During MCU emulation modes and normal expanded modes of operation, this pin is used to enable the Flash EEPROM memory in the memory map (ROMCTL). At the rising edge of $\overline{\text{RESET}}$, the state of this pin is latched to the ROMON bit. The $\overline{\text{EWAIT}}$ input signal maintains the external bus access until the external device is ready to capture data (write) or provide data (read).

The input voltage threshold for PK7 can be configured to reduced levels, to allow data from an external 3.3-V peripheral to be read by the MCU operating at 5.0 V.

1.2.3.42 PK[6:4] / ADDR[22:20] / ACC[2:0] — Port K I/O Pin [6:4]

PK[6:4] are general-purpose input or output pins. During MCU expanded modes of operation, the ACC[2:0] signals are used to indicate the access source of the bus cycle. These pins also provide the expanded addresses ADDR[22:20] for the external bus. In Emulation modes ACC[2:0] is available and is time multiplexed with the high addresses

1.2.3.43 PK[3:0] / ADDR[19:16] / IQSTAT[3:0] — Port K I/O Pins [3:0]

PK3-PK0 are general-purpose input or output pins. In MCU expanded modes of operation, these pins provide the expanded address ADDR[19:16] for the external bus and carry instruction pipe information.

1.2.3.44 PL7 / TXD7 — Port L I/O Pin 7

PL7 is a general-purpose input or output pin. It can be configured as the transmit pin TXD of serial communication interface 7 (SCI7).

1.2.3.45 PL6 / RXD7 — Port L I/O Pin 6

PL6 is a general-purpose input or output pin. It can be configured as the receive pin RXD of serial communication interface 7 (SCI7).

1.2.3.46 PL5 / TXD6 — Port L I/O Pin 5

PL5 is a general-purpose input or output pin. It can be configured as the transmit pin TXD of serial communication interface 6 (SCI6).

1.2.3.47 PL4 / RXD6 — Port L I/O Pin 4

PL4 is a general-purpose input or output pin. It can be configured as the receive pin RXD of serial communication interface 6 (SCI6).

1.2.3.48 PL3 / TXD5 — Port L I/O Pin 3

PL3 is a general-purpose input or output pin. It can be configured as the transmit pin TXD of serial communication interface 5 (SCI5).

1.2.3.49 PL2 / RXD5 — Port L I/O Pin 2

PL2 is a general-purpose input or output pin. It can be configured as the receive pin RXD of serial communication interface 5 (SCI5).

1.2.3.50 PL1 / TXD4 — Port L I/O Pin 1

PL1 is a general-purpose input or output pin. It can be configured as the transmit pin TXD of serial communication interface 4 (SCI4).

1.2.3.51 PL0 / RXD4 — Port L I/O Pin 0

PL0 is a general-purpose input or output pin. It can be configured as the receive pin RXD of serial communication interface 4 (SCI4).

1.2.3.52 PM7 / TXCAN3 / TXCAN4 / TXD3 — Port M I/O Pin 7

PM7 is a general-purpose input or output pin. It can be configured as the transmit pin TXCAN of the scalable controller area network controller 3 or 4 (CAN3 or CAN4). PM7 can be configured as the transmit pin TXD3 of the serial communication interface 3 (SCI3).

1.2.3.53 PM6 / RXCAN3 / RXCAN4 / RXD3 — Port M I/O Pin 6

PM6 is a general-purpose input or output pin. It can be configured as the receive pin RXCAN of the scalable controller area network controller 3 or 4 (CAN3 or CAN4). PM6 can be configured as the receive pin RXD3 of the serial communication interface 3 (SCI3).

1.2.3.54 PM5 / TXCAN0 / TXCAN2 / TXCAN4 / SCK0 — Port M I/O Pin 5

PM5 is a general-purpose input or output pin. It can be configured as the transmit pin TXCAN of the scalable controller area network controllers 0, 2 or 4 (CAN0, CAN2, or CAN4). It can be configured as the serial clock pin SCK of the serial peripheral interface 0 (SPI0).

1.2.3.55 PM4 / RXCAN0 / RXCAN2 / RXCAN4 / MOSI0 — Port M I/O Pin 4

PM4 is a general-purpose input or output pin. It can be configured as the receive pin RXCAN of the scalable controller area network controllers 0, 2, or 4 (CAN0, CAN2, or CAN4). It can be configured as the master output (during master mode) or slave input pin (during slave mode) MOSI for the serial peripheral interface 0 (SPI0).

1.2.3.56 PM3 / TXCAN1 / TXCAN0 / $\overline{\text{SS0}}$ — Port M I/O Pin 3

PM3 is a general-purpose input or output pin. It can be configured as the transmit pin TXCAN of the scalable controller area network controllers 1 or 0 (CAN1 or CAN0). It can be configured as the slave select pin $\overline{\text{SS}}$ of the serial peripheral interface 0 (SPI0).

1.2.3.57 PM2 / RXCAN1 / RXCAN0 / MISO0 — Port M I/O Pin 2

PM2 is a general-purpose input or output pin. It can be configured as the receive pin RXCAN of the scalable controller area network controllers 1 or 0 (CAN1 or CAN0). It can be configured as the master input (during master mode) or slave output pin (during slave mode) MISO for the serial peripheral interface 0 (SPI0).

1.2.3.58 PM1 / TXCAN0 — Port M I/O Pin 1

PM1 is a general-purpose input or output pin. It can be configured as the transmit pin TXCAN of the scalable controller area network controller 0 (CAN0).

1.2.3.59 PM0 / RXCAN0 — Port M I/O Pin 0

PM0 is a general-purpose input or output pin. It can be configured as the receive pin RXCAN of the scalable controller area network controller 0 (CAN0).

1.2.3.60 PP7 / KWP7 / PWM7 / SCK2 / TIMIOC7— Port P I/O Pin 7

PP7 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as pulse width modulator (PWM) channel 7 output, TIM channel 7, or as serial clock pin SCK of the serial peripheral interface 2 (SPI2).

1.2.3.61 PP6 / KWP6 / PWM6 / $\overline{\text{SS2}}$ / TIMIOC6— Port P I/O Pin 6

PP6 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as pulse width modulator (PWM) channel 6 output, TIM channel 6 or as the slave select pin $\overline{\text{SS}}$ of the serial peripheral interface 2 (SPI2).

1.2.3.62 PP5 / KWP5 / PWM5 / MOSI2 / TIMIOC5— Port P I/O Pin 5

PP5 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as pulse width modulator (PWM) channel 5 output, TIM channel 5 or as the master output (during master mode) or slave input pin (during slave mode) MOSI of the serial peripheral interface 2 (SPI2).

1.2.3.63 PP4 / KWP4 / PWM4 / MISO2 / TIMIOC4— Port P I/O Pin 4

PP4 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as pulse width modulator (PWM) channel 4 output, TIM channel 4 or as the master input (during master mode) or slave output (during slave mode) pin MISO of the serial peripheral interface 2 (SPI2).

1.2.3.64 PP3 / KWP3 / PWM3 / $\overline{\text{SS1}}$ / TIMIOC3— Port P I/O Pin 3

PP3 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as pulse width modulator (PWM) channel 3 output, TIM channel 3, or as the slave select pin $\overline{\text{SS}}$ of the serial peripheral interface 1 (SPI1).

1.2.3.65 PP2 / KWP2 / PWM2 / SCK1 / TIMIOC2— Port P I/O Pin 2

PP2 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as pulse width modulator (PWM) channel 2 output, TIM channel 2, or as the serial clock pin SCK of the serial peripheral interface 1 (SPI1).

1.2.3.66 PP1 / KWP1 / PWM1 / MOSI1 / TIMIOC1— Port P I/O Pin 1

PP1 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as pulse width modulator (PWM) channel 1 output, TIM channel 1, or master output (during master mode) or slave input pin (during slave mode) MOSI of the serial peripheral interface 1 (SPI1).

1.2.3.67 PP0 / KWP0 / PWM0 / MISO1 / TIMIOC0— Port P I/O Pin 0

PP0 is a general-purpose input or output pin. It can be configured as a keypad wakeup input. It can be configured as pulse width modulator (PWM) channel 0 output, TIM channel 0 or as the master input (during master mode) or slave output (during slave mode) pin MISO of the serial peripheral interface 1 (SPI1).

1.2.3.68 PR[7:0] / TIMIOC[7:0] — Port R I/O Pins [7:0]

PR[7:0] are general-purpose input or output pins. They can be configured as input capture or output compare pins IOC[7:0] of the standard timer (TIM).

1.2.3.69 PS7 / $\overline{\text{SS0}}$ — Port S I/O Pin 7

PS7 is a general-purpose input or output pin. It can be configured as the slave select pin $\overline{\text{SS}}$ of the serial peripheral interface 0 (SPI0).

1.2.3.70 PS6 / SCK0 — Port S I/O Pin 6

PS6 is a general-purpose input or output pin. It can be configured as the serial clock pin SCK of the serial peripheral interface 0 (SPI0).

1.2.3.71 PS5 / MOSI0 — Port S I/O Pin 5

PS5 is a general-purpose input or output pin. It can be configured as master output (during master mode) or slave input pin (during slave mode) MOSI of the serial peripheral interface 0 (SPI0).

1.2.3.72 PS4 / MISO0 — Port S I/O Pin 4

PS4 is a general-purpose input or output pin. It can be configured as master input (during master mode) or slave output pin (during slave mode) MOSI of the serial peripheral interface 0 (SPI0).

1.2.3.73 PS3 / TXD1 — Port S I/O Pin 3

PS3 is a general-purpose input or output pin. It can be configured as the transmit pin TXD of serial communication interface 1 (SCI1).

1.2.3.74 PS2 / RXD1 — Port S I/O Pin 2

PS2 is a general-purpose input or output pin. It can be configured as the receive pin RXD of serial communication interface 1 (SCI1).

1.2.3.75 PS1 / TXD0 — Port S I/O Pin 1

PS1 is a general-purpose input or output pin. It can be configured as the transmit pin TXD of serial communication interface 0 (SCI0).

1.2.3.76 PS0 / RXD0 — Port S I/O Pin 0

PS0 is a general-purpose input or output pin. It can be configured as the receive pin RXD of serial communication interface 0 (SCI0).

1.2.3.77 PT[7:6] / IOC[7:6] — Port T I/O Pins [7:6]

PT[7:6] are general-purpose input or output pins. They can be configured as input capture or output compare pins IOC[7:6] of the enhanced capture timer (ECT).

1.2.3.78 PT[5] / IOC[5] / VREG_API— Port T I/O Pins [5]

PT[5] is a general-purpose input or output pin. It can be configured as input capture or output compare pin IOC[5] of the enhanced capture timer (ECT) or can be configured to output the VREG_API signal.

1.2.3.79 PT[4:0] / IOC[4:0] — Port T I/O Pins [4:0]

PT[4:0] are general-purpose input or output pins. They can be configured as input capture or output compare pins IOC[4:0] of the enhanced capture timer (ECT).

1.2.4 Power Supply Pins

MC9S12XE-Family power and ground pins are described below.

Because fast signal transitions place high, short-duration current demands on the power supply, use bypass capacitors with high-frequency characteristics and place them as close to the MCU as possible.

NOTE

All V_{SS} pins must be connected together in the application.

1.2.4.1 VDDX[7:1], VSSX[7:1] — Power and Ground Pins for I/O Drivers

External power and ground for I/O drivers. Bypass requirements depend on how heavily the MCU pins are loaded. All V_{DDX} pins are connected together internally. All V_{SSX} pins are connected together internally.

1.2.4.2 VDDR — Power Pin for Internal Voltage Regulator

Input to the internal voltage regulator. The internal voltage regulator is turned off, if V_{DDR} is tied to ground

1.2.4.3 VDD, VSS1,VSS2,VSS3 — Core Power Pins

Power is supplied to the MCU core from the internal voltage regulator, whose load capacitor must be connected to VDD. The voltage supply of nominally 1.8V is derived from the internal voltage regulator. The return current path is through the VSS1,VSS2 and VSS3 pins. No static external loading of these pins is permitted.

1.2.4.4 VDDF — NVM Power Pin

Power is supplied to the MCU NVM through VDDF . The voltage supply of nominally 2.8V is derived from the internal voltage regulator. No static external loading of these pins is permitted.

1.2.4.5 VDDA2, VDDA1, VSSA2, VSSA1 — Power Supply Pins for ATD and Voltage Regulator

These are the power supply and ground input pins for the analog-to-digital converters and the voltage regulator. Internally the V_{DDA} pins are connected together. Internally the V_{SSA} pins are connected together.

1.2.4.6 VRH, VRL — ATD Reference Voltage Input Pins

V_{RH} and V_{RL} are the reference voltage input pins for the analog-to-digital converter.

1.2.4.7 VDDPLL, VSSPL — Power Supply Pins for PLL

These pins provide operating voltage and ground for the oscillator and the phased-locked loop. The voltage supply of nominally 1.8V is derived from the internal voltage regulator. This allows the supply voltage to the oscillator and PLL to be bypassed independently. This voltage is generated by the internal voltage regulator. No static external loading of these pins is permitted.

Table 1-11. Power and Ground Connection Summary

Mnemonic	Nominal Voltage	Description
VDDR	5.0 V	External power supply to internal voltage regulator
VDDX[7:1]	5.0 V	External power and ground, supply to pin drivers
VSSX[7:1]	0 V	
VDDA2, VDDA1	5.0 V	Operating voltage and ground for the analog-to-digital converters and the reference for the internal voltage regulator, allows the supply voltage to the A/D to be bypassed independently.
VSSA2, VSSA1	0 V	
VRL	0 V	Reference voltages for the analog-to-digital converter.
VRH	5.0 V	
VDD	1.8 V	Internal power and ground generated by internal regulator for the internal core.
VSS1, VSS2, VSS3	0V	
VDDF	2.8 V	Internal power and ground generated by internal regulator for the internal NVM.

Table 1-11. Power and Ground Connection Summary (continued)

Mnemonic	Nominal Voltage	Description
VDDPLL	1.8 V	Provides operating voltage and ground for the phased-locked loop. This allows the supply voltage to the PLL to be bypassed independently. Internal power and ground generated by internal regulator.
VSSPLL	0 V	

1.3 System Clock Description

The clock and reset generator module (CRG) provides the internal clock signals for the core and all peripheral modules. Figure 1-9 shows the clock connections from the CRG to all modules.

Consult the CRG specification for details on clock generation.

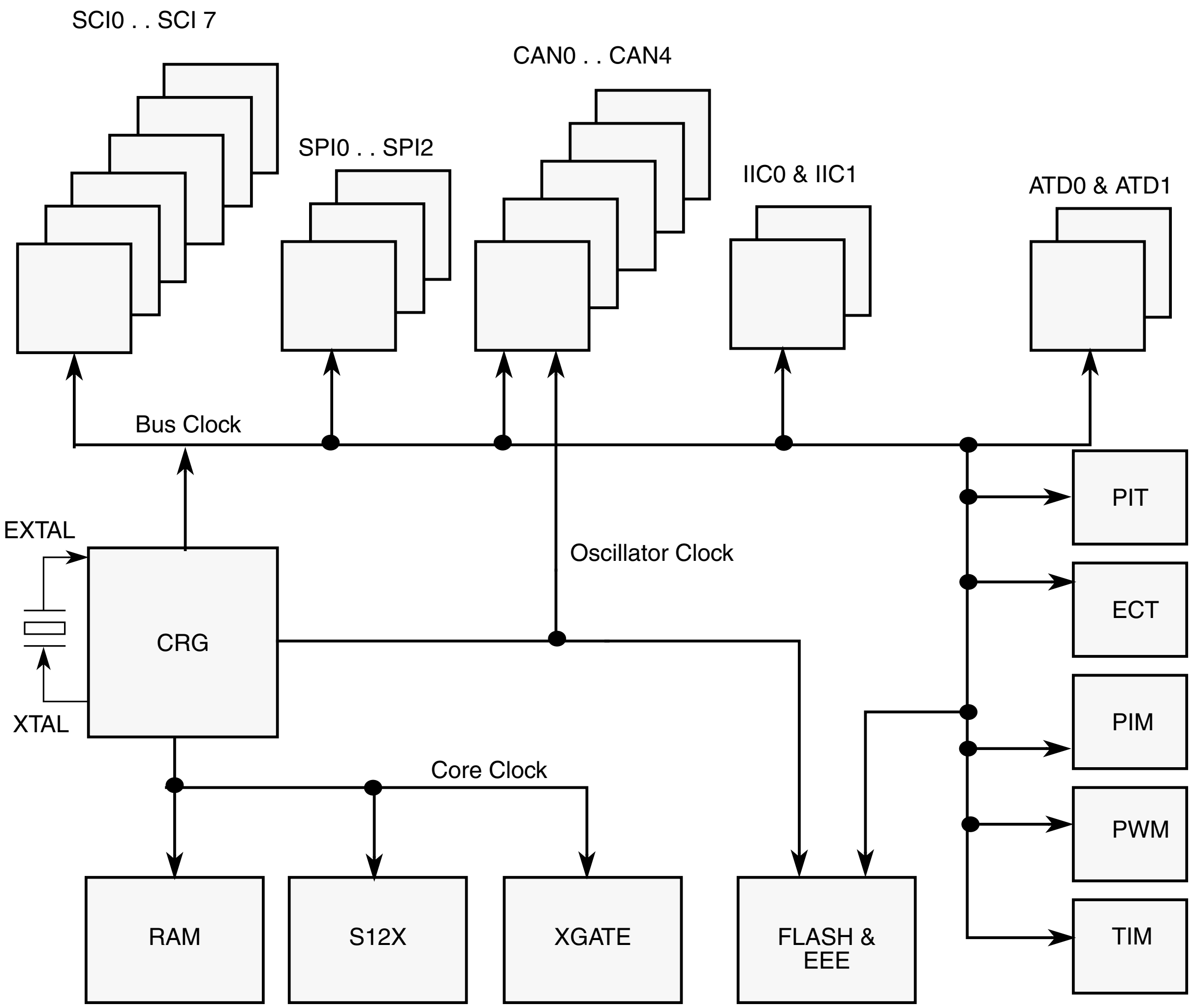


Figure 1-9. Clock Connections

The system clock can be supplied in several ways enabling a range of system operating frequencies to be supported:

- The on-chip phase locked loop (PLL)
- the PLL self clocking
- the oscillator

The clock generated by the PLL or oscillator provides the main system clock frequencies core clock and bus clock. As shown in Figure 1-9, these system clocks are used throughout the MCU to drive the core, the memories, and the peripherals.

The program Flash memory and the EEPROM are 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 NVM's.

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 associated with MCU resource mapping and bus interface configuration. 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.

For system integrity support separate system states are featured as explained in 1.4.4 System States.

1.4.1 Chip Configuration Summary

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

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

The operating mode out of reset is determined by the states of the MODC, MODB, and MODA signals during reset (see Table 1-12). The MODC, MODB, and MODA bits in the MODE register show the current operating mode and provide limited mode switching during operation. The states of the MODC, MODB, and MODA signals are latched into these bits on the rising edge of $\overline{\text{RESET}}$.

In normal expanded mode and in emulation modes the ROMON bit and the EROMON bit in the MMCCTL1 register defines if the on chip flash memory is the memory map, or not. (See Table 1-12.) For a detailed explanation of the ROMON and EROMON bits refer to the MMC description.

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

Table 1-12. Chip Modes and Data Sources

Chip Modes	MODC	MODB	MODA	ROMCTL	EROMCTL	Data Source[1]
Normal single chip	1	0	0	X	X	Internal
Special single chip	0	0	0			
Emulation single chip	0	0	1	X	0	Emulation memory
				X	1	Internal Flash
Normal expanded	1	0	1	0	X	External application
				1	X	Internal Flash
Emulation expanded	0	1	1	0	X	External application
				1	0	Emulation memory
				1	1	Internal Flash
Special test	0	1	0	0	X	External application
				1	X	Internal Flash

[1] Internal means resources inside the MCU are read/written.
Internal Flash means Flash resources inside the MCU are read/written.
Emulation memory means resources inside the emulator are read/written (PRU registers, Flash replacement, RAM, EEPROM, and register space are always considered internal).
External application means resources residing outside the MCU are read/written.

1.4.1.1 Normal Expanded Mode

Ports K, A, and B are configured as a 23-bit address bus, ports C and D are configured as a 16-bit data bus, and port E provides bus control and status signals. This mode allows 16-bit external memory and peripheral devices to be interfaced to the system. The fastest external bus rate is divide by 2 from the internal bus rate.

1.4.1.2 Normal Single-Chip Mode

There is no external bus in this mode. The processor program is executed from internal memory. Ports A, B,C,D, K, and most pins of port E are available as general-purpose I/O.

1.4.1.3 Special Single-Chip Mode

This mode is used for debugging single-chip operation, boot-strapping, or security related operations. The background debug module BDM is active in this mode. The CPU executes a monitor program located in an on-chip ROM. BDM firmware waits for additional serial commands through the BKGD pin. There is no external bus after reset in this mode.

1.4.1.4 Emulation of Expanded Mode

Developers use this mode for emulation systems in which the users target application is normal expanded mode. Code is executed from external memory or from internal memory depending on the state of ROMON and EROMON bit. In this mode the internal operation is visible on external bus interface.

1.4.1.5 Emulation of Single-Chip Mode

Developers use this mode for emulation systems in which the user's target application is normal single-chip mode. Code is executed from external memory or from internal memory depending on the state of ROMON and EROMON bit. In this mode the internal operation is visible on external bus interface.

1.4.1.6 Special Test Mode

This is for Freescale internal use only.

1.4.2 Power Modes

The MCU features two main low-power modes. Consult the respective module description for module specific behavior in system stop, system pseudo stop, and system wait mode. An important source of information about the clock system is the Clock and Reset Generator description (CRG).

1.4.2.1 System Stop Modes

The system stop modes are entered if the CPU executes the STOP instruction unless either the XGATE is active or an NVM command is active. The XGATE is active if it executes a thread or the XGFACT bit in the XGMCTL register is set. Depending on the state of the PSTP bit in the CLKSEL register the MCU goes into pseudo stop mode or full stop mode. Please refer to CRG description. Asserting $\overline{\text{RESET}}$, $\overline{\text{XIRQ}}$, $\overline{\text{IRQ}}$ or any other interrupt that is not masked exits system stop modes. System stop modes can be exited by XGATE or CPU activity independently, depending on the configuration of the interrupt request. If System-Stop is exited on an XGATE request then, as long as the XGATE does not set an interrupt flag on the CPU and the XGATE fake activity bit (FACT) remains cleared, once XGATE activity is completed System Stop mode will automatically be re-entered.

If the CPU executes the STOP instruction whilst XGATE is active or an NVM command is being processed, then the system clocks continue running until XGATE/NVM activity is completed. If a non-masked interrupt occurs within this time then the system does not effectively enter stop mode although the STOP instruction has been executed.

1.4.2.2 Full Stop Mode

The oscillator is stopped in this mode. By default all clocks are switched off and all counters and dividers remain frozen. The Autonomous Periodic Interrupt (API) and ATD modules may be enabled to self wake the device. A Fast wake up mode is available to allow the device to wake from Full Stop mode immediately on the PLL internal clock without starting the oscillator clock.

1.4.2.3 Pseudo Stop Mode

In this mode the system clocks are stopped but the oscillator is still running and the real time interrupt (RTI) and watchdog (COP), API and ATD modules may be enabled. Other peripherals are turned off. This mode consumes more current than system stop mode but, as the oscillator continues to run, the full speed wake up time from this mode is significantly shorter.

1.4.2.4 XGATE Fake Activity Mode

This mode is entered if the CPU executes the STOP instruction when the XGATE is not executing a thread and the XGFACT bit in the XGMCTL register is set. The oscillator remains active and any enabled peripherals continue to function.

1.4.2.5 Wait Mode

This mode is entered when the CPU executes the WAI instruction. In this mode the CPU will not execute instructions. The internal CPU clock is switched off. All peripherals and the XGATE can be active in system wait mode. For further power consumption the peripherals can individually turn off their local clocks. Asserting $\overline{\text{RESET}}$, $\overline{\text{XIRQ}}$, $\overline{\text{IRQ}}$ or any other interrupt that is not masked and is not routed to XGATE ends system wait mode.

1.4.2.6 Run Mode

Although this is not a low-power mode, unused peripheral modules should not be enabled in order to save power.

1.4.3 Freeze Mode

The enhanced capture timer, pulse width modulator, analog-to-digital converters, and the periodic interrupt timer provide a software programmable option to freeze the module status when the background debug module is active. This is useful when debugging application software. For detailed description of the behavior of the ATD0, ATD1, ECT, PWM, and PIT when the background debug module is active consult the corresponding Block Guides.

1.4.4 System States

To facilitate system integrity the MCU can run in Supervisor state or User state. The System States strategy is implemented by additional features on the S12X CPU and a Memory Protection Unit. This is designed to support restricted access for code modules executed by kernels or operating systems supporting access control to system resources.

The current system state is indicated by the U bit in the CPU condition code register. In User state certain CPU instructions are restricted. See the CPU reference guide for details of the U bit and of those instructions affected by User state.

In the case that software task accesses resources outside those defined for it in the MPU a non-maskable interrupt is generated.

1.4.4.1 Supervisor State

This state is intended for configuring the MPU for different tasks that are then executed in User state, returning to Supervisor state on completion of each task. This is the default ’state’ following reset and can be re-entered from User state by an exception (interrupt). If the SVSEN bit in the MPUSEL register of the

MPU is set, access to system resources is only allowed if enabled by a memory range descriptor as defined in the Memory Protection Unit (MPU) description.

1.4.4.2 User State

This state is intended for carrying out system tasks and is entered by setting the U bit of the condition codes register while in Supervisor state. Restrictions apply for the execution of several CPU instructions in User state and access to system resources is only allowed in if enabled by a memory range descriptor as defined in the Memory Protection Unit (MPU) description.

1.5 Security

The MCU security feature allows the protection of the on chip Flash and emulated EEPROM memory. For a detailed description of the security features refer to the S12X9SEC description.

1.6 Resets and Interrupts

Consult the S12XCPU manual and the S12XINT description for information on exception processing.

1.6.1 Resets

Resets are explained in detail in the Clock Reset Generator (CRG) description.

Table 1-13. Reset Sources and Vector Locations

Vector Address	Reset Source	CCR Mask	Local Enable
$FFFE	Power-On Reset (POR)	None	None
$FFFE	Low Voltage Reset (LVR)	None	None
$FFFE	External pin $\overline{\text{RESET}}$	None	None
$FFFE	Illegal Address Reset	None	None
$FFFC	Clock monitor reset	None	PLLCTL (CME, SCME)
$FFFA	COP watchdog reset	None	COP rate select

1.6.2 Vectors

Table 1-14 lists all interrupt sources and vectors in the default order of priority. The interrupt module (S12XINT) provides an interrupt vector base register (IVBR) to relocate the vectors. Associated with each I-bit maskable service request is a configuration register. It selects if the service request is enabled, the service request priority level and whether the service request is handled either by the S12X CPU or by the XGATE module.

Table 1-14. Interrupt Vector Locations (Sheet 1 of 4)

Vector Address[1]	XGATE Channel ID[2]	Interrupt Source	CCR Mask	Local Enable	STOP Wake up	WAIT Wake up
Vector base + $F8	—	Unimplemented instruction trap	None	None	—	—
Vector base+ $F6	—	SWI	None	None	—	—
Vector base+ $F4	—	$\overline{\text{XIRQ}}$	X Bit	None	Yes	Yes
Vector base+ $F2	—	$\overline{\text{IRQ}}$	I bit	IRQCR (IRQEN)	Yes	Yes
Vector base+ $F0	$78	Real time interrupt	I bit	CRGINT (RTIE)	Refer to CRG interrupt section	
Vector base+ $EE	$77	Enhanced capture timer channel 0	I bit	TIE (C0I)	No	Yes
Vector base + $EC	$76	Enhanced capture timer channel 1	I bit	TIE (C1I)	No	Yes
Vector base+ $EA	$75	Enhanced capture timer channel 2	I bit	TIE (C2I)	No	Yes
Vector base+ $E8	$74	Enhanced capture timer channel 3	I bit	TIE (C3I)	No	Yes
Vector base+ $E6	$73	Enhanced capture timer channel 4	I bit	TIE (C4I)	No	Yes
Vector base+ $E4	$72	Enhanced capture timer channel 5	I bit	TIE (C5I)	No	Yes
Vector base + $E2	$71	Enhanced capture timer channel 6	I bit	TIE (C6I)	No	Yes
Vector base+ $E0	$70	Enhanced capture timer channel 7	I bit	TIE (C7I)	No	Yes
Vector base+ $DE	$6F	Enhanced capture timer overflow	I bit	TSRC2 (TOF)	No	Yes
Vector base+ $DC	$6E	Pulse accumulator A overflow	I bit	PACTL (PAOVI)	No	Yes
Vector base + $DA	$6D	Pulse accumulator input edge	I bit	PACTL (PAI)	No	Yes
Vector base + $D8	$6C	SPI0	I bit	SPI0CR1 (SPIE, SPTIE)	No	Yes
Vector base+ $D6	$6B	SCI0	I bit	SCI0CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $D4	$6A	SCI1	I bit	SCI1CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $D2	$69	ATD0	I bit	ATD0CTL2 (ASCIE)	Yes	Yes
Vector base + $D0	$68	ATD1	I bit	ATD1CTL2 (ASCIE)	Yes	Yes
Vector base + $CE	$67	Port J	I bit	PIEJ (PIEJ7-PIEJ0)	Yes	Yes
Vector base + $CC	$66	Port H	I bit	PIEH (PIEH7-PIEH0)	Yes	Yes
Vector base + $CA	$65	Modulus down counter underflow	I bit	MCCTL(MCZI)	No	Yes
Vector base + $C8	$64	Pulse accumulator B overflow	I bit	PBCTL(PBOVI)	No	Yes
Vector base + $C6	$63	CRG PLL lock	I bit	CRGINT(LOCKIE)	Refer to CRG interrupt section	
Vector base + $C4	$62	CRG self-clock mode	I bit	CRGINT (SCMIE)	Refer to CRG interrupt section	
Vector base + $C2	$61	SCI6	I bit	SCI6CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $C0	$60	IIC0 bus	I bit	IBCR0 (IBIE)	No	Yes

Table 1-14. Interrupt Vector Locations (Sheet 2 of 4)

Vector Address[1]	XGATE Channel ID[2]	Interrupt Source	CCR Mask	Local Enable	STOP Wake up	WAIT Wake up
Vector base + $BE	$5F	SPI1	I bit	SPI1CR1 (SPIE, SPTIE)	No	Yes
Vector base + $BC	$5E	SPI2	I bit	SPI2CR1 (SPIE, SPTIE)	No	Yes
Vector base + $BA	$5D	FLASH Fault Detect	I bit	FCNFG2 (FDIE)	No	No
Vector base + $B8	$5C	FLASH	I bit	FCNFG (CCIE, CBEIE)	No	Yes
Vector base + $B6	$5B	CAN0 wake-up	I bit	CAN0RIER (WUPIE)	Yes	Yes
Vector base + $B4	$5A	CAN0 errors	I bit	CAN0RIER (CSCIE, OVRIE)	No	Yes
Vector base + $B2	$59	CAN0 receive	I bit	CAN0RIER (RXFIE)	No	Yes
Vector base + $B0	$58	CAN0 transmit	I bit	CAN0TIER (TXEIE[2:0])	No	Yes
Vector base + $AE	$57	CAN1 wake-up	I bit	CAN1RIER (WUPIE)	Yes	Yes
Vector base + $AC	$56	CAN1 errors	I bit	CAN1RIER (CSCIE, OVRIE)	No	Yes
Vector base + $AA	$55	CAN1 receive	I bit	CAN1RIER (RXFIE)	No	Yes
Vector base + $A8	$54	CAN1 transmit	I bit	CAN1TIER (TXEIE[2:0])	No	Yes
Vector base + $A6	$53	CAN2 wake-up	I bit	CAN2RIER (WUPIE)	Yes	Yes
Vector base + $A4	$52	CAN2 errors	I bit	CAN2RIER (CSCIE, OVRIE)	No	Yes
Vector base + $A2	$51	CAN2 receive	I bit	CAN2RIER (RXFIE)	No	Yes
Vector base + $A0	$50	CAN2 transmit	I bit	CAN2TIER (TXEIE[2:0])	No	Yes
Vector base + $9E	$4F	CAN3 wake-up	I bit	CAN3RIER (WUPIE)	Yes	Yes
Vector base+ $9C	$4E	CAN3 errors	I bit	CAN3RIER (CSCIE, OVRIE)	No	Yes
Vector base+ $9A	$4D	CAN3 receive	I bit	CAN3RIER (RXFIE)	No	Yes
Vector base + $98	$4C	CAN3 transmit	I bit	CAN3TIER (TXEIE[2:0])	No	Yes
Vector base + $96	$4B	CAN4 wake-up	I bit	CAN4RIER (WUPIE)	Yes	Yes
Vector base + $94	$4A	CAN4 errors	I bit	CAN4RIER (CSCIE, OVRIE)	No	Yes
Vector base + $92	$49	CAN4 receive	I bit	CAN4RIER (RXFIE)	No	Yes
Vector base + $90	$48	CAN4 transmit	I bit	CAN4TIER (TXEIE[2:0])	No	Yes
Vector base + $8E	$47	Port P Interrupt	I bit	PIEP (PIEP7-PIEP0)	Yes	Yes
Vector base+ $8C	$46	PWM emergency shutdown	I bit	PWMSDN (PWMIE)	No	Yes

Table 1-14. Interrupt Vector Locations (Sheet 3 of 4)

Vector Address[1]	XGATE Channel ID[2]	Interrupt Source	CCR Mask	Local Enable	STOP Wake up	WAIT Wake up
Vector base + $8A	$45	SCI2	I bit	SCI2CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $88	$44	SCI3	I bit	SCI3CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $86	$43	SCI4	I bit	SCI4CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $84	$42	SCI5	I bit	SCI5CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $82	$41	IIC1 Bus	I bit	IBCR (IBIE)	No	Yes
Vector base + $80	$40	Low-voltage interrupt (LVI)	I bit	VREGCTRL (LVIE)	No	Yes
Vector base + $7E	$3F	Autonomous periodical interrupt (API)	I bit	VREGAPICTRL (APIE)	Yes	Yes
Vector base + $7C	—	High Temperature Interrupt	I bit	VREGHTCL (HTIE)	No	Yes
Vector base + $7A	$3D	Periodic interrupt timer channel 0	I bit	PITINTE (PINTE0)	No	Yes
Vector base + $78	$3C	Periodic interrupt timer channel 1	I bit	PITINTE (PINTE1)	No	Yes
Vector base + $76	$3B	Periodic interrupt timer channel 2	I bit	PITINTE (PINTE2)	No	Yes
Vector base + $74	$3A	Periodic interrupt timer channel 3	I bit	PITINTE (PINTE3)	No	Yes
Vector base + $72	$39	XGATE software trigger 0	I bit	XGMCTL (XGIE)	No	Yes
Vector base + $70	$38	XGATE software trigger 1	I bit	XGMCTL (XGIE)	No	Yes
Vector base + $6E	$37	XGATE software trigger 2	I bit	XGMCTL (XGIE)	No	Yes
Vector base + $6C	$36	XGATE software trigger 3	I bit	XGMCTL (XGIE)	No	Yes
Vector base + $6A	$35	XGATE software trigger 4	I bit	XGMCTL (XGIE)	No	Yes
Vector base + $68	$34	XGATE software trigger 5	I bit	XGMCTL (XGIE)	No	Yes
Vector base + $66	$33	XGATE software trigger 6	I bit	XGMCTL (XGIE)	No	Yes
Vector base + $64	$32	XGATE software trigger 7	I bit	XGMCTL (XGIE)	No	Yes
Vector base + $62	Reserved					
Vector base + $60	Reserved					
Vector base + $5E	$2F	Periodic interrupt timer channel 4	I bit	PITINTE (PINTE4)	No	Yes
Vector base + $5C	$2E	Periodic interrupt timer channel 5	I bit	PITINTE (PINTE5)	No	Yes
Vector base + $5A	$2D	Periodic interrupt timer channel 6	I bit	PITINTE (PINTE6)	No	Yes
Vector base + $58	$2C	Periodic interrupt timer channel 7	I bit	PITINTE (PINTE7)	No	Yes
Vector base + $56	$2B	SCI7	I bit	SCI7CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $54	$2A	TIM timer channel 0	I bit	TIE (C0I)	No	Yes
Vector base + $52	$29	TIM timer channel 1	I bit	TIE (C1I)	No	Yes
Vector base + $50	$28	TIM timer channel 2	I bit	TIE (C2I)	No	Yes

Table 1-14. Interrupt Vector Locations (Sheet 4 of 4)

Vector Address[1]	XGATE Channel ID[2]	Interrupt Source	CCR Mask	Local Enable	STOP Wake up	WAIT Wake up
Vector base+ $4E	$27	TIM timer channel 3	I bit	TIE (C3I)	No	Yes
Vector base + $4C	$26	TIM timer channel 4	I bit	TIE (C4I)	No	Yes
Vector base+ $4A	$25	TIM timer channel 5	I bit	TIE (C5I)	No	Yes
Vector base+ $48	$24	TIM timer channel 6	I bit	TIE (C6I)	No	Yes
Vector base+ $46	$23	TIM timer channel 7	I bit	TIE (C7I)	No	Yes
Vector base+ $44	$22	TIM timer overflow	I bit	TSRC2 (TOF)	No	Yes
Vector base + $42	$21	TIM Pulse accumulator A overflow	I bit	PACTL (PAOVI)	No	Yes
Vector base+ $40	$20	TIM Pulse accumulator input edge	I bit	PACTL (PAI)	No	Yes
Vector base + $3E	$1F	ATD0 Compare Interrupt	I bit	ATD0CTL2 (ACMPIE)	Yes	Yes
Vector base + $3C	$1E	ATD1 Compare Interrupt	I bit	ATD1CTL2 (ACMPIE)	Yes	Yes
Vector base+ $18 to Vector base + $3A	Reserved					
Vector base + $16	—	XGATE software error interrupt	None	None	No	Yes
Vector base + $14	—	MPU Access Error	None	None	No	No
Vector base + $12	—	System Call Interrupt (SYS)	—	None	—	—
Vector base + $10	—	Spurious interrupt	—	None	—	—

[1] 16 bits vector address based

[2] For detailed description of XGATE channel ID refer to XGATE Block Guide

1.6.3 Effects of Reset

When a reset occurs, MCU registers and control bits are initialized. Refer to the respective block descriptions for register reset states.

On each reset, the Flash module executes a reset sequence to load Flash configuration registers and initialize the buffer RAM EEE partition, if required.

1.6.3.1 Flash Configuration Reset Sequence (Core Hold Phase)

On each reset, the Flash module will hold CPU activity while loading Flash module registers and configuration from the Flash memory. The duration of this phase is given as t_{RST} in the device electrical parameter specification. If double faults are detected in the reset phase, Flash module protection and security may be active on leaving reset. This is explained in more detail in the Flash module section.

1.6.3.2 EEE Reset Sequence Phase (Core Active Phase)

During this phase of the reset sequence (following on from the core hold phase) the CPU can execute instructions while the FTM initialization completes and, if configured for EEE operation, the EEE RAM is loaded with valid data from the D-Flash EEE partition. Completion of this phase is indicated by the CCIF flag in the FTM FSTAT register becoming set. If the CPU accesses any EEE RAM location before the CCIF flag is set, the CPU is stalled until the FTM reset sequence is complete and the EEE RAM data is valid. Once the CCIF flag is set, indicating the end of this phase, the EEE RAM can be accessed without impacting the CPU and FTM commands can be executed.

1.6.3.3 Reset While Flash Command Active

If a reset occurs while any Flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.

1.6.3.4 I/O Pins

Refer to the PIM block description for reset configurations of all peripheral module ports.

1.6.3.5 Memory

The RAM arrays are not initialized out of reset.

1.6.3.6 COP Configuration

The COP timeout rate bits CR[2:0] and the WCOP bit in the COPCTL register are loaded on rising edge of $\overline{\text{RESET}}$ from the Flash register FOPT. See Table 1-15 and Table 1-16 for coding. The FOPT register is loaded from the Flash configuration field byte at global address $7FFF0E during the reset sequence.

If the MCU is secured the COP timeout rate is always set to the longest period (CR[2:0] = 111) after COP reset.

Table 1-15. Initial COP Rate Configuration

NV[2:0] in FOPT Register	CR[2:0] in COPCTL Register
000	111
001	110
010	101
011	100
100	011
101	010
110	001
111	000

Table 1-16. Initial WCOP Configuration

NV[3] in FOPT Register	WCOP in COPCTL Register
1	0
0	1

1.7 ADC0 Configuration

1.7.1 External Trigger Input Connection

The ADC module includes four external trigger inputs ETRIG0, ETRIG1, ETRIG2, and ETRIG3. The external trigger allows the user to synchronize ADC conversion to external trigger events. Table 1-17 shows the connection of the external trigger inputs.

Table 1-17. ATD0 External Trigger Sources

External Trigger Input	Connectivity
ETRIG0	Pulse width modulator channel 1
ETRIG1	Pulse width modulator channel 3
ETRIG2	Periodic interrupt timer hardware trigger 0
ETRIG3	Periodic interrupt timer hardware trigger 1

Consult the ATD block description for information about the analog-to-digital converter module. ATD block description refererences to freeze mode are equivalent to active BDM mode.

1.7.2 ADC0 Channel[17] Connection

Further to the 16 externally available channels, ADC0 features an extra channel[17] that is connected to the internal temperature sensor at device level. To access this channel ADC0 must use the channel encoding SC:CD:CC:CB:CA = 1:0:0:0:1 in ATDCTL5. For more temperature sensor information, please refer to 1.10.1 Temperature Sensor Configuration

1.8 ADC1 External Trigger Input Connection

The ADC module includes four external trigger inputs ETRIG0, ETRIG1, ETRIG2, and ETRIG3. The external trigger feature allows the user to synchronize ADC conversion to external trigger events. Table 1-18 shows the connection of the external trigger inputs.

Table 1-18. ATD1 External Trigger Sources

External Trigger Input	Connectivity
ETRIG0	Pulse width modulator channel 1
ETRIG1	Pulse width modulator channel 3
ETRIG2	Periodic interrupt timer hardware trigger 0
ETRIG3	Periodic interrupt timer hardware trigger 1

Consult the ADC block description for information about the analog-to-digital converter module. ADC block description refererences to freeze mode are equivalent to active BDM mode.

1.9 MPU Configuration

The MPU has the option of a third bus master (CPU + XGATE + other) which is not present on this device family but may be on other parts.

1.10 VREG Configuration

The VREGEN connection of the voltage regulator is tied internally to VDDR such that the voltage regulator is always enabled with VDDR connected to a positive supply voltage. The device must be configured with the internal voltage regulator enabled. Operation in conjunction with an external voltage regulator is not supported.

The autonomous periodic interrupt clock output is mapped to PortT[5].

The API trimming register APITR is loaded on rising edge of $\overline{\text{RESET}}$ from the Flash IFR option field at global address 0x40_00F0 bits[5:0] during the reset sequence. Currently factory programming of this IFR range is not supported.

1.10.1 Temperature Sensor Configuration

The VREG high temperature trimming register bits VREGHTTR[3:0] are loaded from the internal Flash during the reset sequence. To use the high temperature interrupt within the specified limits (T_{HTIA} and T_{HTID}) these bits must be loaded with 0x8. Currently factory programming is not supported.

The device temperature can be monitored on ADC0 channel[17].
The internal bandgap reference voltage can also be mapped to ADC0 analog input channel[17]. The voltage regulator VSEL bit when set, maps the bandgap and, when clear, maps the temperature sensor to ADC0 channel[17].

Read access to reserved VREG register space returns “0”. Write accesses have no effect. This device does not support access abort of reserved VREG register space.

1.11 BDM Clock Configuration

The BDM alternate clock source is the oscillator clock.

1.12 S12XEPIM Configuration

On smaller derivatives the S12XEPIM module is a subset of the S12XEP100. The registers of the unavailable ports are unimplemented.

1.13 Oscillator Configuration

The $\overline{\text{XCLKS}}$ is an input signal which controls whether a crystal in combination with the internal loop controlled (low power) Pierce oscillator is used or whether full swing Pierce oscillator/external clock circuitry is used. For this device $\overline{\text{XCLKS}}$ is mapped to PE7.

The $\overline{\text{XCLKS}}$ signal selects the oscillator configuration during reset low phase while a clock quality check is ongoing. This is the case for:

- Power on reset or low-voltage reset
- Clock monitor reset
- Any reset while in self-clock mode or full stop mode

The selected oscillator configuration is frozen with the rising edge of the RESET pin in any of these above described reset cases.

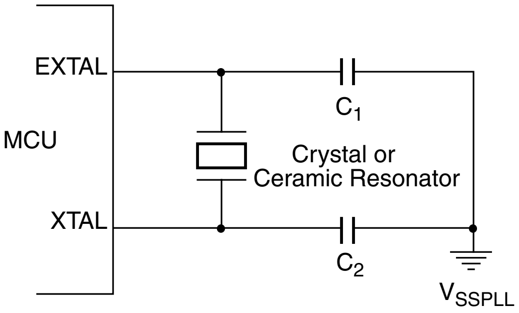


Figure 1-10. Loop Controlled Pierce Oscillator Connections ($\overline{\text{XCLKS}}$ = 1)

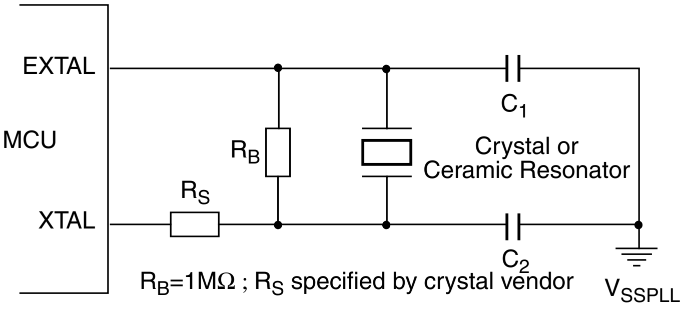


Figure 1-11. Full Swing Pierce Oscillator Connections ($\overline{\text{XCLKS}}$ = 0)

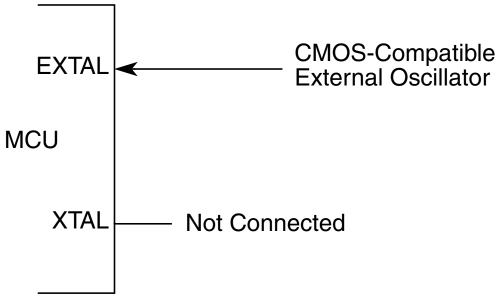


Figure 1-12. External Clock Connections ($\overline{\text{XCLKS}}$ = 0)

Chapter 2
Port Integration Module (S12XEPIMV1)

Table 2-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V01.17	02 Apr 2008		• Corrected reduced drive strength to 1/5 • Separated PE1,0 bit descriptions from other PE GPIO
V01.18	25 Nov 2008	2.3.19/120 2.4.3.4/181	• Corrected alternative functions on Port K (ACC[2:0]) • Corrected functions on PE[5] (MODB) and PE[2] (WE)
V01.19	18 Dec 2009		• Added function independency to reduced drive and wired-or bit descriptions • Minor corrections

2.1 Introduction

2.1.1 Overview

The S12XE Family Port Integration Module establishes the interface between the peripheral modules including the non-multiplexed External Bus Interface module (S12X_EBI) and the I/O pins for all ports. It controls the electrical pin properties as well as the signal prioritization and multiplexing on shared pins.

This document covers:

- Port A and B used as address output of the S12X_EBI
- Port C and D used as data I/O of the S12X_EBI
- Port E associated with the S12X_EBI control signals and the $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$ interrupt inputs
- Port K associated with address output and control signals of the S12X_EBI
- Port T associated with 1 ECT module
- Port S associated with 2 SCI and 1 SPI modules
- Port M associated with 4 MSCAN and 1 SCI module
- Port P connected to the PWM and 2 SPI modules - inputs can be used as an external interrupt source
- Port H associated with 4 SCI modules - inputs can be used as an external interrupt source
- Port J associated with 1 MSCAN, 1 SCI, 2 IIC modules and chip select outputs - inputs can be used as an external interrupt source
- Port AD0 and AD1 associated with two 16-channel ATD modules
- Port R associated with 1 standard timer (TIM) module
- Port L associated with 4 SCI modules

- Port F associated with IIC, SCI and chip select outputs

Most I/O pins can be configured by register bits to select data direction and drive strength, to enable and select pull-up or pull-down devices.

NOTE

This document assumes the availability of all features (208-pin package option). Some functions are not available on lower pin count package options. Refer to the pin-out summary in the SOC Guide.

2.1.2 Features

The Port Integration Module includes these distinctive registers:

- Data and data direction registers for Ports A, B, C, D, E, K, T, S, M, P, H, J, AD0, AD1, R, L, and F when used as general-purpose I/O
- Control registers to enable/disable pull-device and select pull-ups/pull-downs on Ports T, S, M, P, H, J, R, L, and F on per-pin basis
- Control registers to enable/disable pull-up devices on Ports AD0 and AD1 on per-pin basis
- Single control register to enable/disable pull-ups on Ports A, B, C, D, E, and K on per-port basis and on BKGD pin
- Control registers to enable/disable reduced output drive on Ports T, S, M, P, H, J, AD0, AD1, R, L, and F on per-pin basis
- Single control register to enable/disable reduced output drive on Ports A, B, C, D, E, and K on per-port basis
- Control registers to enable/disable open-drain (wired-or) mode on Ports S, M, and L
- Interrupt flag register for pin interrupts on Ports P, H, and J
- Control register to configure $\overline{\text{IRQ}}$ pin operation
- Free-running clock outputs

A standard port pin has the following minimum features:

- Input/output selection
- 5V output drive with two selectable drive strengths
- 5V digital and analog input
- Input with selectable pull-up or pull-down device

Optional features supported on dedicated pins:

- Open drain for wired-or connections
- Interrupt inputs with glitch filtering
- Reduced input threshold to support low voltage applications

2.2 External Signal Description

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

Table 2-2 shows all the pins and their functions that are controlled by the Port Integration Module. *Refer to the SOC Guide 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 2-2. Pin Functions and Priorities

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Pin Function after Reset
-	BKGD	MODC [2]	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] [3]	O	High-order external bus address output (multiplexed with IVIS data)	Mode dependent [4]
		GPIO	I/O	General-purpose I/O	
B	PB[7:1]	ADDR[7:1] mux IVD[7:1] [3]	O	Low-order external bus address output (multiplexed with IVIS data)	Mode dependent [4]
		GPIO	I/O	General-purpose I/O	
	PB[0]	ADDR[0] mux IVD0 [3]	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 [4]
		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 [4]
		GPIO	I/O	General-purpose I/O	

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Pin Function after Reset
E	PE[7]	$\overline{\text{XCLKS}}$ [2]	I	External clock selection input during $\overline{\text{RESET}}$	Mode dependent [4]
		ECLKX2	I	Free-running clock output at Core Clock rate (ECLK x 2)	
		GPIO	I/O	General-purpose I/O	
	PE[6]	MODB [2]	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 [2]	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 [2]	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	
		GPI	I	General-purpose input	
	PE[0]	$\overline{\text{XIRQ}}$	I	Non-maskable level-sensitive interrupt input	
		GPI	I	General-purpose input	
K	PK[7]	ROMCTL [2]	I	ROMON bit control input during $\overline{\text{RESET}}$	Mode dependent [3]
		$\overline{\text{EWAIT}}$	I	External Wait signal Configurable for reduced input threshold	
		GPIO	I/O	General-purpose I/O	
	PK[6:4]	ADDR[22:20] mux ACC[2:0] [3]	O	Extended external bus address output (multiplexed with access master output)	
		GPIO	I/O	General-purpose I/O	
	PK[3:0]	ADDR[19:16] mux IQSTAT[3:0] [3]	O	Extended external bus address output (multiplexed with instruction pipe status bits)	
		GPIO	I/O	General-purpose I/O	

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Pin Function after Reset
T	PT[7]	IOC[7]	I/O	Enhanced Capture Timer Channels 7 input/output	GPIO
		GPIO	I/O	General-purpose I/O	
	PT[5]	IOC[5]	I/O	Enhanced Capture Timer Channel 5 input/output	
		VREG_API	O	VREG Autonomous Periodical Interrupt output	
		GPIO	I/O	General-purpose I/O	
	PT[4:0]	IOC[4:0]	I/O	Enhanced Capture Timer Channels 4 - 0 input/output	
		GPIO	I/O	General-purpose I/O	
S	PS7	$\overline{\text{SS0}}$	I/O	Serial Peripheral Interface 0 slave select output in master mode, input in slave mode or master mode.	GPIO
		GPIO	I/O	General-purpose I/O	
	PS6	SCK0	I/O	Serial Peripheral Interface 0 serial clock pin	
		GPIO	I/O	General-purpose I/O	
	PS5	MOSI0	I/O	Serial Peripheral Interface 0 master out/slave in pin	
		GPIO	I/O	General-purpose I/O	
	PS4	MISO0	I/O	Serial Peripheral Interface 0 master in/slave out pin	
		GPIO	I/O	General-purpose I/O	
	PS3	TXD1	O	Serial Communication Interface 1 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PS2	RXD1	I	Serial Communication Interface 1 receive pin	
		GPIO	I/O	General-purpose I/O	
	PS1	TXD0	O	Serial Communication Interface 0 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PS0	RXD0	I	Serial Communication Interface 0 receive pin	
		GPIO	I/O	General-purpose I/O	

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Pin Function after Reset
M	PM7	TXCAN3	O	MSCAN3 transmit pin	GPIO
		(TXCAN4)	O	MSCAN4 transmit pin	
		TXD3	O	Serial Communication Interface 3 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PM6	RXCAN3	I	MSCAN3 receive pin	
		(RXCAN4)	I	MSCAN4 receive pin	
		RXD3	I	Serial Communication Interface 3 receive pin	
		GPIO	I/O	General-purpose I/O	
	PM5	TXCAN2	O	MSCAN2 transmit pin	
		(TXCAN0)	O	MSCAN0 transmit pin	
		(TXCAN4)	O	MSCAN4 transmit pin	
		(SCK0)	I/O	Serial Peripheral Interface 0 serial clock pin If CAN0 is routed to PM[3:2] the SPI0 can still be used in bidirectional master mode.	
		GPIO	I/O	General-purpose I/O	
	PM4	RXCAN2	I	MSCAN2 receive pin	
		(RXCAN0)	I	MSCAN0 receive pin	
		(RXCAN4)	I	MSCAN4 receive pin	
		(MOSI0)	I/O	Serial Peripheral Interface 0 master out/slave in pin If CAN0 is routed to PM[3:2] the SPI0 can still be used in bidirectional master mode.	
		GPIO	I/O	General-purpose I/O	
	PM3	TXCAN1	O	MSCAN1 transmit pin	
		(TXCAN0)	O	MSCAN0 transmit pin	
		($\overline{\text{SS0}}$)	I/O	Serial Peripheral Interface 0 slave select output in master mode, input for slave mode or master mode.	
		GPIO	I/O	General-purpose I/O	
	PM2	RXCAN1	I	MSCAN1 receive pin	
		(RXCAN0)	I	MSCAN0 receive pin	
		(MISO0)	I/O	Serial Peripheral Interface 0 master in/slave out pin	
		GPIO	I/O	General-purpose I/O	
	PM1	TXCAN0	O	MSCAN0 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PM0	RXCAN0	I	MSCAN0 receive pin	
		GPIO	I/O	General-purpose I/O	

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Pin Function after Reset
P	PP7	PWM7	I/O	Pulse Width Modulator input/output channel 7	GPIO
		SCK2	I/O	Serial Peripheral Interface 2 serial clock pin	
		(TIMIOC7)	I/O	Timer Channel 7 input/output	
		GPIO/KWP7	I/O	General-purpose I/O with interrupt	
	PP6	PWM6	O	Pulse Width Modulator output channel 6	
		$\overline{\text{SS2}}$	I/O	Serial Peripheral Interface 2 slave select output in master mode, input for slave mode or master mode.	
		(TIMIOC6)	I/O	Timer Channel 6 input/output	
		GPIO/KWP6	I/O	General-purpose I/O with interrupt	
	PP5	PWM5	O	Pulse Width Modulator output channel 5	
		MOSI2	I/O	Serial Peripheral Interface 2 master out/slave in pin	
		(TIMIOC5)	I/O	Timer Channel 5 input/output	
		GPIO/KWP5	I/O	General-purpose I/O with interrupt	
	PP4	PWM4	O	Pulse Width Modulator output channel 4	
		MISO2	I/O	Serial Peripheral Interface 2 master in/slave out pin	
		(TIMIOC4)	I/O	Timer Channel 4 input/output	
		GPIO/KWP4	I/O	General-purpose I/O with interrupt	
	PP3	PWM3	O	Pulse Width Modulator output channel 3	
		$\overline{\text{SS1}}$	I/O	Serial Peripheral Interface 1 slave select output in master mode, input for slave mode or master mode.	
		(TIMIOC3)	I/O	Timer Channel 3 input/output	
		GPIO/KWP3	I/O	General-purpose I/O with interrupt	
	PP2	PWM2	O	Pulse Width Modulator output channel 2	
		SCK1	I/O	Serial Peripheral Interface 1 serial clock pin	
		(TIMIOC2)	I/O	Timer Channel 2 input/output	
		GPIO/KWP2	I/O	General-purpose I/O with interrupt	
	PP1	PWM1	O	Pulse Width Modulator output channel 1	
		MOSI1	I/O	Serial Peripheral Interface 1 master out/slave in pin	
		(TIMIOC1)	I/O	Timer Channel 1 input/output	
		GPIO/KWP1	I/O	General-purpose I/O with interrupt	
	PP0	PWM0	O	Pulse Width Modulator output channel 0	
		MISO1	I/O	Serial Peripheral Interface 1 master in/slave out pin	
		(TIMIOC0)	I/O	Timer Channel 0 input/output	
		GPIO/KWP0	I/O	General-purpose I/O with interrupt	

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Pin Function after Reset
H	PH7	($\overline{\text{SS2}}$)	I/O	Serial Peripheral Interface 2 slave select output in master mode, input for slave mode or master mode	GPIO
		TXD5	O	Serial Communication Interface 5 transmit pin	
		GPIO/KWH7	I/O	General-purpose I/O with interrupt	
	PH6	(SCK2)	I/O	Serial Peripheral Interface 2 serial clock pin	
		RXD5	I	Serial Communication Interface 5 receive pin	
		GPIO/KWH6	I/O	General-purpose I/O with interrupt	
	PH5	(MOSI2)	I/O	Serial Peripheral Interface 2 master out/slave in pin	
		TXD4	O	Serial Communication Interface 4 transmit pin	
		GPIO/KWH5	I/O	General-purpose I/O with interrupt	
	PH4	(MISO2)	I/O	Serial Peripheral Interface 2 master in/slave out pin	
		RXD4	I	Serial Communication Interface 4 receive pin	
		GPIO/KWH4	I/O	General-purpose I/O with interrupt	
	PH3	($\overline{\text{SS1}}$)	I/O	Serial Peripheral Interface 1 slave select output in master mode, input for slave mode or master mode.	
		TXD7	O	Serial Communication Interface 7 transmit pin	
		GPIO/KWH3	I/O	General-purpose I/O with interrupt	
	PH2	(SCK1)	I/O	Serial Peripheral Interface 1 serial clock pin	
		RXD7	I	Serial Communication Interface 7 receive pin	
		GPIO/KWH2	I/O	General-purpose I/O with interrupt	
	PH1	(MOSI1)	I/O	Serial Peripheral Interface 1 master out/slave in pin	
		TXD6	O	Serial Communication Interface 6 transmit pin	
		GPIO/KWH1	I/O	General-purpose I/O with interrupt	
	PH0	(MISO1)	I/O	Serial Peripheral Interface 1 master in/slave out pin	
		TXD6	O	Serial Communication Interface 6 transmit pin	
		GPIO/KWH0	I/O	General-purpose I/O with interrupt	

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Pin Function after Reset
J	PJ7	TXCAN4	O	MSCAN4 transmit pin	GPIO
		SCL0	O	Inter Integrated Circuit 0 serial clock line	
		(TXCAN0)	O	MSCAN0 transmit pin	
		GPIO/KWJ7	I/O	General-purpose I/O with interrupt	
	PJ6	RXCAN4	I	MSCAN4 receive pin	
		SDA0	I/O	Inter Integrated Circuit 0 serial data line	
		(RXCAN0)	I	MSCAN0 receive pin	
		GPIO/KWJ6	I/O	General-purpose I/O with interrupt	
	PJ5	SCL1	O	Inter Integrated Circuit 1 serial clock line	
		$\overline{\text{CS2}}$	O	Chip select 2	
		GPIO/KWJ5	I/O	General-purpose I/O with interrupt	
	PJ4	SDA1	I/O	Inter Integrated Circuit 1 serial data line	
		$\overline{\text{CS0}}$	O	Chip select 0	
		GPIO/KWJ4	I/O	General-purpose I/O with interrupt	
	PJ3	GPIO/KWJ3	I/O	General-purpose I/O with interrupt	
	PJ2	$\overline{\text{CS1}}$	O	Chip select 1	
		GPIO/KWJ2	I/O	General-purpose I/O with interrupt	
	PJ1	TXD2	O	Serial Communication Interface 2 transmit pin	
		GPIO/KWJ1	I/O	General-purpose I/O with interrupt	
	PJ0	RXD2	I	Serial Communication Interface 2 receive pin	
		$\overline{\text{CS3}}$	O	Chip select 3	
		GPIO/KWJ0	I/O	General-purpose I/O with interrupt	
AD0	PAD[15:0]	GPIO	I/O	General-purpose I/O	GPIO
		AN[15:0]	I	ATD0 analog inputs	
AD1	PAD[31:16]	GPIO	I/O	General-purpose I/O	GPIO
		AN[15:0]	I	ATD1 analog inputs	
R	PR[7:0]	TIMIOC[7:0]	I/O	Timer Channels 7- 0 input/output	GPIO
		GPIO	I/O	General-purpose I/O	

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Pin Function after Reset
L	PL7	(TXD7)	O	Serial Communication Interface 7 transmit pin	GPIO
		GPIO	I/O	General-purpose I/O	
	PL6	(RXD7)	I	Serial Communication Interface 7 receive pin	
		GPIO	I/O	General-purpose I/O	
	PL5	(TXD6)	O	Serial Communication Interface 6 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PL4	(RXD6)	I	Serial Communication Interface 6 receive pin	
		GPIO	I/O	General-purpose I/O	
	PL3	(TXD5)	O	Serial Communication Interface 5 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PL2	(RXD5)	I	Serial Communication Interface 5 receive pin	
		GPIO	I/O	General-purpose I/O	
	PL1	(TXD4)	O	Serial Communication Interface 4 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PL0	(RXD4)	I	Serial Communication Interface 4 receive pin	
		GPIO	I/O	General-purpose I/O	
F	PF7	(TXD3)	O	Serial Communication Interface 3 transmit pin	GPIO
		GPIO	I/O	General-purpose I/O	
	PF6	(RXD3)	I	Serial Communication Interface 3 receive pin	
		GPIO	I/O	General-purpose I/O	
	PF5	(SCL0)	O	Inter Integrated Circuit 0 serial clock line	
		GPIO	I/O	General-purpose I/O	
	PF4	(SDA0)	I/O	Inter Integrated Circuit 0 serial data line	
		GPIO	I/O	General-purpose I/O	
	PF3	($\overline{\text{CS3}}$)	O	Chip select 3	
		GPIO	I/O	General-purpose I/O	
	PF2	($\overline{\text{CS2}}$)	O	Chip select 2	
		GPIO	I/O	General-purpose I/O	
	PF1	($\overline{\text{CS1}}$)	O	Chip select 1	
		GPIO	I/O	General-purpose I/O	
	PF0	($\overline{\text{CS0}}$)	O	Chip select 0	
		GPIO	I/O	General-purpose I/O	

1. Signals in brackets denote alternative module routing pins.
2. Function active when $\overline{\text{RESET}}$ asserted.
3. Only available in emulation modes or in Special Test Mode with IVIS on.
4. Refer to S12X_EBI section.

2.3 Memory Map and Register Definition

This section provides a detailed description of all Port Integration Module registers.

2.3.1 Memory Map

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	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 PORTC	R	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
	W								
0x0005 PORTD	R	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	W								
0x0006 DDRC	R	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
	W								
0x0007 DDRD	R	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
	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 Non-PIM Address Range	R	Non-PIM Address Range							
	W								
0x000C PUCR	R	PUPKE	BKPUE	0	PUPEE	PUPDE	PUPCE	PUPBE	PUPAE
	W			▨					
0x000D RDRIV	R	RDPK	0	0	RDPE	RDPD	RDPC	RDPB	RDPA
	W		▨	▨					

▨ = Unimplemented or Reserved

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000E– 0x001B Non-PIM Address Range	R	Non-PIM Address Range							
	W								
0x001C ECLKCTL	R	NECLK	NCLKX2	DIV16	EDIV4	EDIV3	EDIV2	EDIV1	EDIV0
	W								
0x001D Reserved	R	0	0	0	0	0	0	0	0
	W								
0x001E IRQCR	R	IRQE	IRQEN	0	0	0	0	0	0
	W								
0x001F Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0020– 0x0031 Non-PIM Address Range	R	Non-PIM Address Range							
	W								
0x0032 PORTK	R	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
	W								
0x0033 DDRK	R	DDRK7	DDRK6	DDRK5	DDRK4	DDRK3	DDRK2	DDRK1	DDRK0
	W								
0x0034– 0x023F Non-PIM Address Range	R	Non-PIM Address Range							
	W								
0x0240 PTT	R	PTT7	PTT6	PTT5	PTT4	PTT3	PTT2	PTT1	PTT0
	W								
0x0241 PTIT	R	PTIT7	PTIT6	PTIT5	PTIT4	PTIT3	PTIT2	PTIT1	PTIT0
	W								
0x0242 DDRT	R	DDRT7	DDRT6	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0
	W								
0x0243 RDRT	R	RDRT7	RDRT6	RDRT5	RDRT4	RDRT3	RDRT2	RDRT1	RDRT0
	W								

▨ = Unimplemented or Reserved

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0244 PERT	R	PERT7	PERT6	PERT5	PERT4	PERT3	PERT2	PERT1	PERT0
	W								
0x0245 PPST	R	PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0
	W								
0x0246 Reserved	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0247 Reserved	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0248 PTS	R	PTS7	PTS6	PTS5	PTS4	PTS3	PTS2	PTS1	PTS0
	W								
0x0249 PTIS	R	PTIS7	PTIS6	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x024A DDRS	R	DDRS7	DDRS6	DDRS5	DDRS4	DDRS3	DDRS2	DDRS1	DDRS0
	W								
0x024B RDRS	R	RDRS7	RDRS6	RDRS5	RDRS4	RDRS3	RDRS2	RDRS1	RDRS0
	W								
0x024C PERS	R	PERS7	PERS6	PERS5	PERS4	PERS3	PERS2	PERS1	PERS0
	W								
0x024D PPSS	R	PPSS7	PPSS6	PPSS5	PPSS4	PPSS3	PPSS2	PPSS1	PPSS0
	W								
0x024E WOMS	R	WOMS7	WOMS6	WOMS5	WOMS4	WOMS3	WOMS2	WOMS1	WOMS0
	W								
0x024F Reserved	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0250 PTM	R	PTM7	PTM6	PTM5	PTM4	PTM3	PTM2	PTM1	PTM0
	W								
0x0251 PTIM	R	PTIM7	PTIM6	PTIM5	PTIM4	PTIM3	PTIM2	PTIM1	PTIM0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0252 DDRM	R	DDRM7	DDRM6	DDRM5	DDRM4	DDRM3	DDRM2	DDRM1	DDRM0
	W								

▨ = Unimplemented or Reserved

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0253 RDRM	R	RDRM7	RDRM6	RDRM5	RDRM4	RDRM3	RDRM2	RDRM1	RDRM0
	W								
0x0254 PERM	R	PERM7	PERM6	PERM5	PERM4	PERM3	PERM2	PERM1	PERM0
	W								
0x0255 PPSM	R	PPSM7	PPSM6	PPSM5	PPSM4	PPSM3	PPSM2	PPSM1	PPSM0
	W								
0x0256 WOMM	R	WOMM7	WOMM6	WOMM5	WOMM4	WOMM3	WOMM2	WOMM1	WOMM0
	W								
0x0257 MODRR	R	0	MODRR6	MODRR5	MODRR4	MODRR3	MODRR2	MODRR1	MODRR0
	W	▨							
0x0258 PTP	R	PTP7	PTP6	PTP5	PTP4	PTP3	PTP2	PTP1	PTP0
	W								
0x0259 PTIP	R	PTIP7	PTIP6	PTIP5	PTIP4	PTIP3	PTIP2	PTIP1	PTIP0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x025A DDRP	R	DDRP7	DDRP6	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0
	W								
0x025B RDRP	R	RDRP7	RDRP6	RDRP5	RDRP4	RDRP3	RDRP2	RDRP1	RDRP0
	W								
0x025C PERP	R	PERP7	PERP6	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0
	W								
0x025D PPSP	R	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
	W								
0x025E PIEP	R	PIEP7	PIEP6	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
	W								
0x025F PIFP	R	PIFP7	PIFP6	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
	W								
0x0260 PTH	R	PTH7	PTH6	PTH5	PTH4	PTH3	PTH2	PTH1	PTH0
	W								
0x0261 PTIH	R	PTIH7	PTIH6	PTIH5	PTIH4	PTIH3	PTIH2	PTIH1	PTIH0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0262 DDRH	R	DDRH7	DDRH6	DDRH5	DDRH4	DDRH3	DDRH2	DDRH1	DDRH0
	W								

▨ = Unimplemented or Reserved

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0263 RDRH	R	RDRH7	RDRH6	RDRH5	RDRH4	RDRH3	RDRH2	RDRH1	RDRH0
	W								
0x0264 PERH	R	PERH7	PERH6	PERH5	PERH4	PERH3	PERH2	PERH1	PERH0
	W								
0x0265 PPSH	R	PPSH7	PPSH6	PPSH5	PPSH4	PPSH3	PPSH2	PPSH1	PPSH0
	W								
0x0266 PIEH	R	PIEH7	PIEH6	PIEH5	PIEH4	PIEH3	PIEH2	PIEH1	PIEH0
	W								
0x0267 PIFH	R	PIFH7	PIFH6	PIFH5	PIFH4	PIFH3	PIFH2	PIFH1	PIFH0
	W								
0x0268 PTJ	R	PTJ7	PTJ6	PTJ5	PTJ4	PTJ3	PTJ2	PTJ1	PTJ0
	W								
0x0269 PTIJ	R	PTIJ7	PTIJ6	PTIJ5	PTIJ4	PTIJ3	PTIJ2	PTIJ1	PTIJ0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x026A DDRJ	R	DDRJ7	DDRJ6	DDRJ5	DDRJ4	DDRJ3	DDRJ2	DDRJ1	DDRJ0
	W								
0x026B RDRJ	R	RDRJ7	RDRJ6	RDRJ5	RDRJ4	RDRJ3	RDRJ2	RDRJ1	RDRJ0
	W								
0x026C PERJ	R	PERJ7	PERJ6	PERJ5	PERJ4	PERJ3	PERJ2	PERJ1	PERJ0
	W								
0x026D PPSJ	R	PPSJ7	PPSJ6	PPSJ5	PPSJ4	PPSJ3	PPSJ2	PPSJ1	PPSJ0
	W								
0x026E PIEJ	R	PIEJ7	PIEJ6	PIEJ5	PIEJ4	PIEJ3	PIEJ2	PIEJ1	PIEJ0
	W								
0x026F PIFJ	R	PIFJ7	PIFJ6	PIFJ5	PIFJ4	PIFJ3	PIFJ2	PIFJ1	PIFJ0
	W								
0x0270 PT0AD0	R	PT0AD07	PT0AD06	PT0AD05	PT0AD04	PT0AD03	PT0AD02	PT0AD01	PT0AD00
	W								
0x0271 PT1AD0	R	PT1AD07	PT1AD06	PT1AD05	PT1AD04	PT1AD03	PT1AD02	PT1AD01	PT1AD00
	W								

▨ = Unimplemented or Reserved

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0272 DDR0AD0	R W	DDR0AD07	DDR0AD06	DDR0AD05	DDR0AD04	DDR0AD03	DDR0AD02	DDR0AD01	DDR0AD00
0x0273 DDR1AD0	R W	DDR1AD07	DDR1AD06	DDR1AD05	DDR1AD04	DDR1AD03	DDR1AD02	DDR1AD01	DDR1AD00
0x0274 RDR0AD0	R W	RDR0AD07	RDR0AD06	RDR0AD05	RDR0AD04	RDR0AD03	RDR0AD02	RDR0AD01	RDR0AD00
0x0275 RDR1AD0	R W	RDR1AD07	RDR1AD06	RDR1AD05	RDR1AD04	RDR1AD03	RDR1AD02	RDR1AD01	RDR1AD00
0x0276 PER0AD0	R W	PER0AD07	PER0AD06	PER0AD05	PER0AD04	PER0AD03	PER0AD02	PER0AD01	PER0AD00
0x0277 PER1AD0	R W	PER1AD07	PER1AD06	PER1AD05	PER1AD04	PER1AD03	PER1AD02	PER1AD01	PER1AD00
0x0278 PT0AD1	R W	PT0AD17	PT0AD16	PT0AD15	PT0AD14	PT0AD13	PT0AD12	PT0AD11	PT0AD10
0x0279 PT1AD1	R W	PT1AD17	PT1AD16	PT1AD15	PT1AD14	PT1AD13	PT1AD12	PT1AD11	PT1AD10
0x027A DDR0AD1	R W	DDR0AD17	DDR0AD16	DDR0AD15	DDR0AD14	DDR0AD13	DDR0AD12	DDR0AD11	DDR0AD10
0x027B DDR1AD1	R W	DDR1AD17	DDR1AD16	DDR1AD15	DDR1AD14	DDR1AD13	DDR1AD12	DDR1AD11	DDR1AD10
0x027C RDR0AD1	R W	RDR0AD17	RDR0AD16	RDR0AD15	RDR0AD14	RDR0AD13	RDR0AD12	RDR0AD11	RDR0AD10
0x027D RDR1AD1	R W	RDR1AD17	RDR1AD16	RDR1AD15	RDR1AD14	RDR1AD13	RDR1AD12	RDR1AD11	RDR1AD10
0x027E PER0AD1	R W	PER0AD17	PER0AD16	PER0AD15	PER0AD14	PER0AD13	PER0AD12	PER0AD1‘	PER0AD10
0x027F PER1AD1	R W	PER1AD17	PER1AD16	PER1AD15	PER1AD14	PER1AD13	PER1AD12	PER1AD11	PER1AD10
0x0280– 0x0267 Non-PIM Address Range	R W	Non-PIM Address Range							

▨ = Unimplemented or Reserved

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0368 PTR	R	PTR7	PTR6	PTR5	PTR4	PTR3	PTR2	PTR1	PTR0
	W								
0x0369 PTIR	R	PTIR7	PTIR6	PTIR5	PTIR4	PTIR3	PTIR2	PTIR1	PTIR0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x036A DDRR	R	DDRR7	DDRR6	DDRR5	DDRR4	DDRR3	DDRR2	DDRR1	DDRR0
	W								
0x036B RDRR	R	RDRR7	RDRR6	RDRR5	RDRR4	RDRR3	RDRR2	RDRR1	RDRR0
	W								
0x036C PERR	R	PERR7	PERR6	PERR5	PERR4	PERR3	PERR2	PERR1	PERR0
	W								
0x036D PPSR	R	PPSR7	PPSR6	PPSR5	PPSR4	PPSR3	PPSR2	PPSR1	PPSR0
	W								
0x036E Reserved	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x036F PTRRR	R	PTRRR7	PTRRR6	PTRRR5	PTRRR4	PTRRR3	PTRRR2	PTRRR1	PTRRR0
	W								
0x0370 PTL	R	PTL7	PTL6	PTL5	PTL4	PTL3	PTL2	PTL1	PTL0
	W								
0x0371 PTIL	R	PTIL7	PTIL6	PTIL5	PTIL4	PTIL3	PTIL2	PTIL1	PTIL0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0372 DDRL	R	DDRL7	DDRL6	DDRL5	DDRL4	DDRL3	DDRL2	DDRL1	DDRL0
	W								
0x0373 RDRL	R	RDRL7	RDRL6	RDRL5	RDRL4	RDRL3	RDRL2	RDRL1	RDRL0
	W								
0x0374 PERL	R	PERL7	PERL6	PERL5	PERL4	PERL3	PERL2	PERL1	PERL0
	W								
0x0375 PPSL	R	PPSL7	PPSL6	PPSL5	PPSL4	PPSL3	PPSL2	PPSL1	PPSL0
	W								
0x0376 WOML	R	WOML7	WOML6	WOML5	WOML4	WOML3	WOML2	WOML1	WOML0
	W								
0x0377 PTLRR	R	PTLRR7	PTLRR6	PTLRR5	PTLRR4	0	0	0	0
	W					▨	▨	▨	▨

▨ = Unimplemented or Reserved

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0378 PTF	R	PTF7	PTF6	PTF5	PTF4	PTF3	PTF2	PTF1	PTF0
	W								
0x0379 PTIF	R	PTIF7	PTIF6	PTIF5	PTIF4	PTIF3	PTIF2	PTIF1	PTIF0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x037A DDRF	R	DDRF7	DDRF6	DDRF5	DDRF4	DDRF3	DDRF2	DDRF1	DDRF0
	W								
0x037B RDRF	R	RDRF7	RDRF6	RDRF5	RDRF4	RDRF3	RDRF2	RDRF1	RDRF0
	W								
0x037C PERF	R	PERF7	PERF6	PERF5	PERF4	PERF3	PERF2	PERF1	PERF0
	W								
0x037D PPSF	R	PPSF7	PPSF6	PPSF5	PPSF4	PPSF3	PPSF2	PPSF1	PPSF0
	W								
0x037E Reserved	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x037F PTFRR	R	0	0	PTFRR5	PTFRR4	PTFRR3	PTFRR2	PTFRR1	PTFRR0
	W	▨	▨						

▨ = Unimplemented or Reserved

2.3.2 Register Descriptions

The following table summarizes the effect of the various configuration bits, i.e. data direction (DDR), output level (IO), reduced drive (RDR), pull enable (PE), pull select (PS) on the pin function and pull device activity.

The configuration bit PS is used for two purposes:

1. Configure the sensitive interrupt edge (rising or falling), if interrupt is enabled.
2. Select either a pull-up or pull-down device if PE is active.

Table 2-3. Pin Configuration Summary

DDR	IO	RDR	PE	PS[1]	IE[2]	Function	Pull Device	Interrupt
0	x	x	0	x	0	Input	Disabled	Disabled
0	x	x	1	0	0	Input	Pull Up	Disabled
0	x	x	1	1	0	Input	Pull Down	Disabled
0	x	x	0	0	1	Input	Disabled	Falling edge
0	x	x	0	1	1	Input	Disabled	Rising edge
0	x	x	1	0	1	Input	Pull Up	Falling edge
0	x	x	1	1	1	Input	Pull Down	Rising edge
1	0	0	x	x	0	Output, full drive to 0	Disabled	Disabled
1	1	0	x	x	0	Output, full drive to 1	Disabled	Disabled
1	0	1	x	x	0	Output, reduced drive to 0	Disabled	Disabled
1	1	1	x	x	0	Output, reduced drive to 1	Disabled	Disabled
1	0	0	x	0	1	Output, full drive to 0	Disabled	Falling edge
1	1	0	x	1	1	Output, full drive to 1	Disabled	Rising edge
1	0	1	x	0	1	Output, reduced drive to 0	Disabled	Falling edge
1	1	1	x	1	1	Output, reduced drive to 1	Disabled	Rising edge

[1] Always “0” on Port A, B, C, D, E, K, AD0, and AD1.

[2] Applicable only on Port P, H, and J.

NOTE

All register bits in this module are completely synchronous to internal clocks during a register read.

2.3.3 Port A Data Register (PORTA)

Address 0x0000 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Altern. Function	ADDR15 mux IVD15	ADDR14 mux IVD14	ADDR13 mux IVD13	ADDR12 mux IVD12	ADDR11 mux IVD11	ADDR10 mux IVD10	ADDR9 mux IVD9	ADDR8 mux IVD8
Reset	0	0	0	0	0	0	0	0

Figure 2-1. Port A Data Register (PORTA)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

Table 2-4. PORTA Register Field Descriptions

Field	Description
7-0 PA	**Port A general purpose input/output data**—Data Register Port A pins 7 through 0 are associated with address outputs ADDR[15:8] respectively in expanded modes. In emulation modes the address is multiplexed with IVD[15:8]. When not used with the alternative function, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.4 Port B Data Register (PORTB)

Address 0x0001 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Altern. Function	ADDR7 mux IVD7	ADDR6 mux IVD6	ADDR5 mux IVD5	ADDR4 mux IVD4	ADDR3 mux IVD3	ADDR2 mux IVD2	ADDR1 mux IVD1	ADDR0 mux IVD0 or $\overline{\text{UDS}}$
Reset	0	0	0	0	0	0	0	0

Figure 2-2. Port B Data Register (PORTB)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

Table 2-5. PORTB Register Field Descriptions

Field	Description
7-0 PB	**Port B general purpose input/output data**—Data Register Port B pins 7 through 0 are associated with address outputs ADDR[7:0] respectively in expanded modes. In emulation modes the address is multiplexed with IVD[7:0]. In normal expanded mode pin 0 is related to the $\overline{\text{UDS}}$ input. When not used with the alternative function, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.5 Port A Data Direction Register (DDRA)

Address 0x0002 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset	0	0	0	0	0	0	0	0

Figure 2-3. Port A Data Direction Register (DDRA)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
   Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

Table 2-6. DDRA Register Field Descriptions

Field	Description
7-0 DDRA	**Port A Data Direction**— This register controls the data direction of pins 7 through 0. The external bus function forces the I/O state to be outputs for all associated pins. In this case the data direction bits will not change. When operating a pin as a general purpose I/O, the associated data direction bit determines whether it is an input or output. 1 Associated pin is configured as output. 0 Associated pin is configured as high-impedance input.

2.3.6 Port B Data Direction Register (DDRB)

Address 0x0003 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Reset	0	0	0	0	0	0	0	0

Figure 2-4. Port B Data Direction Register (DDRB)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
   Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

Table 2-7. DDRB Register Field Descriptions

Field	Description
7-0 DDRB	**Port B Data Direction**— This register controls the data direction of pins 7 through 0. The external bus function forces the I/O state to be outputs for all associated pins. In this case the data direction bits will not change. When operating a pin as a general purpose I/O, the associated data direction bit determines whether it is an input or output. 1 Associated pin is configured as output. 0 Associated pin is configured as high-impedance input.

2.3.7 Port C Data Register (PORTC)

Address 0x0004 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Altern. Function	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
Reset	0	0	0	0	0	0	0	0

Figure 2-5. Port C Data Register (PORTC)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
   Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

Table 2-8. PORTC Register Field Descriptions

Field	Description
7-0 PC	**Port C general purpose input/output data**—Data Register Port C pins 7 through 0 are associated with data I/O lines DATA[15:8] respectively in expanded modes. When not used with the alternative function, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.8 Port D Data Register (PORTD)

Address 0x0005 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Altern. Function	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
Reset	0	0	0	0	0	0	0	0

Figure 2-6. Port D Data Register (PORTD)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

Table 2-9. PORTD Register Field Descriptions

Field	Description
7-0 PD	**Port D general purpose input/output data**—Data Register Port D pins 7 through 0 are associated with data I/O lines DATA[7:0] respectively in expanded modes. When not used with the alternative function, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.9 Port C Data Direction Register (DDRC)

Address 0x0006 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
Reset	0	0	0	0	0	0	0	0

Figure 2-7. Port C Data Direction Register (DDRC)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

Table 2-10. DDRC Register Field Descriptions

Field	Description
7-0 DDRC	Port C Data Direction— This register controls the data direction of pins 7 through 0. The external bus function controls the data direction for the associated pins. In this case the data direction bits will not change. When operating a pin as a general purpose I/O, the associated data direction bit determines whether it is an input or output. 1 Associated pin is configured as output. 0 Associated pin is configured as high-impedance input.

2.3.10 Port D Data Direction Register (DDRD)

Address 0x0007 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Reset	0	0	0	0	0	0	0	0

Figure 2-8. Port D Data Direction Register (DDRD)

[1] Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

Table 2-11. DDRD Register Field Descriptions

Field	Description
7-0 DDRD	**Port D Data Direction—** This register controls the data direction of pins 7 through 0. When used with the external bus this function controls the data direction for the associated pins. In this case the data direction bits will not change. When operating a pin as a general purpose I/O, the associated data direction bit determines whether it is an input or output. 1 Associated pin is configured as output. 0 Associated pin is configured as high-impedance input.

2.3.11 Port E Data Register (PORTE)

Address 0x0008 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
W							▨	▨
Altern. Function	$\overline{XCLKS}$ or ECLKX2	MODB or $\overline{TAGHI}$	MODA or $\overline{RE}$ or $\overline{TAGLO}$	ECLK	EROMCTL or $\overline{LSTRB}$ or $\overline{LDS}$	R$\overline{W}$ or $\overline{WE}$	$\overline{IRQ}$	$\overline{XIRQ}$
Reset	0	0	0	0	0	0	—[2]	—[2]

▨ = Unimplemented or Reserved

Figure 2-9. Port E Data Register (PORTE)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
   Write: Anytime. In emulation modes, write operations will also be directed to the external bus.
2. These registers are reset to zero. Two bus clock cycles after reset release the register values are updated with the associated pin values.

Table 2-12. PORTE Register Field Descriptions

Field	Description
7-2 PE	**Port E general purpose input/output data**—Data Register Port E bits 7 through 0 are associated with external bus control signals and interrupt inputs. These include mode select (MODB, MODA), E clock, double frequency E clock, Instruction Tagging High and Low ($\overline{TAGHI}$, $\overline{TAGLO}$), Read/Write (R$\overline{W}$), Read Enable and Write Enable ($\overline{RE}$, $\overline{WE}$), Lower Data Select ($\overline{LDS}$). When not used with the alternative functions, Port E pins 7-2 can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. Pins 6 and 5 are inputs with enabled pull-down devices while RESET pin is low. Pins 7 and 3 are inputs with enabled pull-up devices while RESET pin is low.
1 PE	**Port E general purpose input data and interrupt**—Data Register, $\overline{IRQ}$ input. This pin can be used as general purpose and $\overline{IRQ}$ input.
0 PE	**Port E general purpose input data and interrupt**—Data Register, $\overline{XIRQ}$ input. This pin can be used as general purpose and $\overline{XIRQ}$ input.

2.3.12 Port E Data Direction Register (DDRE)

Address 0x0009 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	0	0
W							▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 2-10. Port E Data Direction Register (DDRE)

[1] Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

Table 2-13. DDRE Register Field Descriptions

Field	Description
7-2 DDRE	**Port E Data Direction—** This register controls the data direction of pins 7 through 2. The external bus function controls the data direction for the associated pins. In this case the data direction bits will not change. When operating a pin as a general purpose I/O, the associated data direction bit determines whether it is an input or output. 1 Associated pin is configured as output. 0 Associated pin is configured as high-impedance input.
1-0	**Reserved—** Port E bit 1 (associated with $\overline{\text{IRQ}}$) and bit 0 (associated with $\overline{\text{XIRQ}}$) cannot be configured as outputs. Port E, bits 1 and 0, can be read regardless of whether the alternate interrupt function is enabled.

2.3.13 S12X_EBI ports, BKGD pin Pull-up Control Register (PUCR)

Address 0x000C (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	PUPKE	BKPUE	0	PUPEE	PUPDE	PUPCE	PUPBE	PUPAE
W			▨					
Reset	1	1	0	1	0	0	0	0

▨ = Unimplemented or Reserved

Figure 2-11. S12X_EBI ports, BKGD pin Pull-up Control Register (PUCR)

[1] Read:Anytime in single-chip modes.
Write:Anytime, except BKPUE which is writable in Special Test Mode only.

Table 2-14. PUCR Register Field Descriptions

Field	Description
7 PUPKE	**Pull-up Port K Enable**—Enable pull-up devices on all Port K input pins This bit configures whether pull-up devices are activated, if the pins are used as inputs. This bit has no effect if the pins are used as outputs. Out of reset the pull-up devices are enabled. 1 Pull-up devices enabled. 0 Pull-up devices disabled.
6 BKPUE	**BKGD pin pull-up Enable**—Enable pull-up devices on BKGD pin This bit configures whether a pull-up device is activated, if the pin is used as input. This bit has no effect if the pin is used as outputs. Out of reset the pull-up device is enabled. 1 Pull-up device enabled. 0 Pull-up device disabled.
5	**Reserved**—
4 PUPEE	**Pull-up Port E Enable**—Enable pull-up devices on all Port E input pins except on pins 5 and 6 which have pull-down devices only enabled during reset. This bit has no effect on these pins. This bit configures whether pull-up devices are activated, if the pins are used as inputs. This bit has no effect if the pins are used as outputs. Out of reset the pull-up devices are enabled. 1 Pull-up devices enabled. 0 Pull-up devices disabled.
3 PUPDE	**Pull-up Port D Enable**—Enable pull-up devices on all Port D input pins This bit configures whether pull-up devices are activated, if the pins are used as inputs. This bit has no effect if the pins are used as outputs. Out of reset the pull-up devices are disabled. 1 Pull-up devices enabled. 0 Pull-up devices disabled.
2 PUPCE	**Pull-up Port C Enable**—Enable pull-up devices on all Port C input pins This bit configures whether pull-up devices are activated, if the pins are used as inputs. This bit has no effect if the pins are used as outputs. Out of reset the pull-up devices are disabled. 1 Pull-up devices enabled. 0 Pull-up devices disabled.
1 PUPBE	**Pull-up Port B Enable**—Enable pull-up devices on all Port B input pins This bit configures whether pull-up devices are activated, if the pins are used as inputs. This bit has no effect if the pins are used as outputs. Out of reset the pull-up devices are disabled. 1 Pull-up devices enabled. 0 Pull-up devices disabled.
0 PUPAE	**Pull-up Port A Enable**—Enable pull-up devices on all Port A input pins This bit configures whether pull-up devices are activated, if the pins are used as inputs. This bit has no effect if the pins are used as outputs. Out of reset the pull-up devices are disabled. 1 Pull-up devices enabled. 0 Pull-up devices disabled.

2.3.14 S12X_EBI ports Reduced Drive Register (RDRIV)

Address 0x000D (PRR)

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	RDPK	0	0	RDPE	RDPD	RDPC	RDPB	RDPA
W		▨	▨					
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 2-12. S12X_EBI ports Reduced Drive Register (RDRIV)

[1] Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
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 approx. 1/5 of the full drive strength.

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

Table 2-15. RDRIV Register Field Descriptions

Field	Description
7 RDPK	**Port K reduced drive**—Select reduced drive for outputs This bit configures the drive strength of all Port K output pins as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.
6-5	**Reserved**—
4 RDPE	**Port E reduced drive**—Select reduced drive for outputs This bit configures the drive strength of all Port E output pins as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.
3 RDPD	**Port D reduced drive**—Select reduced drive for outputs This bit configures the drive strength of all output pins as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.
2 RDPC	**Port C reduced drive**—Select reduced drive for outputs This bit configures the drive strength of all output pins as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

Table 2-15. RDRIV Register Field Descriptions (continued)

Field	Description
1 RDPB	**Port B reduced drive**—Select reduced drive for outputs This bit configures the drive strength of all output pins as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.
0 RDPA	**Port A reduced drive**—Select reduced drive for outputs This bit configures the drive strength of all output pins as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.15 ECLK Control Register (ECLKCTL)

Address 0x001C (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	NECLK	NCLKX2	DIV16	EDIV4	EDIV3	EDIV2	EDIV1	EDIV0
Reset[2]:	Mode Depen-dent	1	0	0	0	0	0	0
SS	0	1	0	0	0	0	0	0
ES	1	1	0	0	0	0	0	0
ST	0	1	0	0	0	0	0	0
EX	0	1	0	0	0	0	0	0
NS	1	1	0	0	0	0	0	0
NX	0	1	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 2-13. ECLK Control Register (ECLKCTL)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
   Write: Anytime. In emulation modes, write operations will also be directed to the external bus.
2. Reset values in emulation modes are identical to those of the target mode.

The ECLKCTL register is used to control the availability of the free-running clocks and the free-running clock divider.

Table 2-16. ECLKCTL Register Field Descriptions

Field	Description
7 NECLK	**No ECLK**—Disable ECLK output This bit controls the availability of a free-running clock on the ECLK pin. Clock output is always active in emulation modes and if enabled in all other operating modes. 1 ECLK disabled 0 ECLK enabled
6 NCLKX2	**No ECLKX2**—Disable ECLKX2 output This bit controls the availability of a free-running clock on the ECLKX2 pin. This clock has a fixed rate of twice the internal Bus Clock. Clock output is always active in emulation modes and if enabled in all other operating modes. 1 ECLKX2 disabled 0 ECLKX2 enabled
5 DIV16	**Free-running ECLK predivider**—Divide by 16 This bit enables a divide-by-16 stage on the selected EDIV rate. 1 Divider enabled: ECLK rate = EDIV rate divided by 16 0 Divider disabled: ECLK rate = EDIV rate
4-0 EDIV	**Free-running ECLK Divider**—Configure ECLK rate These bits determine the rate of the free-running clock on the ECLK pin. Divider is always disabled in emulation modes and active as programmed in all other operating modes. 00000 ECLK rate = Bus Clock rate 00001 ECLK rate = Bus Clock rate divided by 2 00010 ECLK rate = Bus Clock rate divided by 3, ... 11111 ECLK rate = Bus Clock rate divided by 32

2.3.16 PIM Reserved Register

Address 0x001D (PRR) Access: User read[1]

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

▨ = Unimplemented or Reserved

Figure 2-14. PIM Reserved Register

[1] Read: Always reads 0x00
Write: Unimplemented

2.3.17 IRQ Control Register (IRQCR)

Address 0x001E Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	IRQE	IRQEN	0	0	0	0	0	0
W			▨	▨	▨	▨	▨	▨
Reset	0	1	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 2-15. IRQ Control Register (IRQCR)

1. Read: See individual bit descriptions below.
   Write: See individual bit descriptions below.

Table 2-17. IRQCR Register Field Descriptions

Field	Description
7 IRQE	**IRQ select edge sensitive only—** Special modes: Read or write anytime. Normal & emulation modes: Read anytime, write once. 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. 0 $\overline{\text{IRQ}}$ configured for low level recognition.
6 IRQEN	**External IRQ enable—** Read or write anytime. 1 External IRQ pin is connected to interrupt logic. 0 External IRQ pin is disconnected from interrupt logic.
5-0	**Reserved—**

2.3.18 PIM Reserved Register

This register is reserved for factory testing of the PIM module and is not available in normal operation.

Address 0x001F Access: User read[1]

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

▨ = Unimplemented or Reserved

Figure 2-16. PIM Reserved Register

1. Read: Always reads 0x00
   Write: Unimplemented

NOTE

Writing to this register when in special modes can alter the pin functionality.

2.3.19 Port K Data Register (PORTK)

Address 0x0032 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0
Altern. Function	ROMCTL or $\overline{\text{EWAIT}}$	ADDR22 mux ACC2	ADDR21 mux ACC1	ADDR20 mux ACC0	ADDR19 mux IQSTAT3	ADDR18 mux IQSTAT2	ADDR17 mux IQSTAT1	ADDR16 mux IQSTAT0
Reset	0	0	0	0	0	0	0	0

Figure 2-17. Port K Data Register (PORTK)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

Table 2-18. PORTK Register Field Descriptions

Field	Description
7-0 PK	**Port K general purpose input/output data**—Data Register Port K pins 7 through 0 are associated with external bus control signals and internal memory expansion emulation pins. These include ADDR[22:16], Access Source (ACC[2:0]), External Wait ($\overline{\text{EWAIT}}$) and instruction pipe signals IQSTAT[3:0]. Bits 6-0 carry the external addresses in all expanded modes. In emulation modes the address is multiplexed with the alternate functions ACC and IQSTAT on the respective pins. When not used with the alternative function, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.20 Port K Data Direction Register (DDRK)

Address 0x0033 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRK7	DDRK6	DDRK5	DDRK4	DDRK3	DDRK2	DDRK1	DDRK0
Reset	0	0	0	0	0	0	0	0

Figure 2-18. Port K Data Direction Register (DDRK)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

Table 2-19. DDRK Register Field Descriptions

Field	Description
7-0 DDRK	**Port K Data Direction**— This register controls the data direction of pins 7 through 0. The external bus function controls the data direction for the associated pins. In this case the data direction bits will not change. When operating a pin as a general purpose I/O, the associated data direction bit determines whether it is an input or output. 1 Associated pin is configured as output. 0 Associated pin is configured as high-impedance input.

2.3.21 Port T Data Register (PTT)

Address 0x0240 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PTT7	PTT6	PTT5	PTT4	PTT3	PTT2	PTT1	PTT0
Altern. Function	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0
	—	—	VREG_API	—	—	—	—	—
Reset	0	0	0	0	0	0	0	0

Figure 2-19. Port T Data Register (PTT)

[1] Read: Anytime.
Write: Anytime.

Table 2-20. PTT Register Field Descriptions

Field	Description
7-6 PTT	**Port T general purpose input/output data**—Data Register Port T pins 7 through 0 are associated with ECT channels IOC7 and IOC6. When not used with the alternative function, these pins can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
5 PTT	**Port T general purpose input/output data**—Data Register Port T pins 5 is associated with ECT channel IOC5 and the VREG_API output. The ECT function takes precedence over the VREG_API and the general purpose I/O function if the related channel is enabled. When not used with the alternative function, these pins can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
4-0 PTT	**Port T general purpose input/output data**—Data Register Port T pins 4 through 0 are associated with ECT channels IOC4 through IOC0. When not used with the alternative function, these pins can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.22 Port T Input Register (PTIT)

Address 0x0241 Access: User read[1]

	7	6	5	4	3	2	1	0
R	PTIT7	PTIT6	PTIT5	PTIT4	PTIT3	PTIT2	PTIT1	PTIT0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	u	u	u	u	u	u	u	u

▨ = Unimplemented or Reserved u = Unaffected by reset

Figure 2-20. Port T Input Register (PTIT)

[1] Read: Anytime.
Write:Never, writes to this register have no effect.

Table 2-21. PTIT Register Field Descriptions

Field	Description
7-0 PTIT	**Port T input data**— This register always reads back the buffered state of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

2.3.23 Port T Data Direction Register (DDRT)

Address 0x0242 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRT7	DDRT6	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0
Reset	0	0	0	0	0	0	0	0

Figure 2-21. Port T Data Direction Register (DDRT)

[1] Read: Anytime.
Write: Anytime.

Table 2-22. DDRT Register Field Descriptions

Field	Description
7-0 DDRT	**Port T data direction**— This register controls the data direction of pins 7 through 0. The ECT forces the I/O state to be an output for each timer port associated with an enabled output compare. In this case the data direction bits will not change. The data direction bits revert to controlling the I/O direction of a pin when the associated timer output compare is disabled. The timer Input Capture always monitors the state of the pin. 1 Associated pin is configured as output. 0 Associated pin is configured as high-impedance input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTT or PTIT registers, when changing the DDRT register.

2.3.24 Port T Reduced Drive Register (RDRT)

Address 0x0243 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDRT7	RDRT6	RDRT5	RDRT4	RDRT3	RDRT2	RDRT1	RDRT0
Reset	0	0	0	0	0	0	0	0

Figure 2-22. Port T Reduced Drive Register (RDRT)

[1] Read: Anytime.
Write: Anytime.

Table 2-23. RDRT Register Field Descriptions

Field	Description
7-0 RDRT	**Port T reduced drive**—Select reduced drive for outputs This register configures the drive strength of output pins 7 through 0 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.25 Port T Pull Device Enable Register (PERT)

Address 0x0244 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PERT7	PERT6	PERT5	PERT4	PERT3	PERT2	PERT1	PERT0
Reset	0	0	0	0	0	0	0	0

Figure 2-23. Port T Pull Device Enable Register (PERT)

[1] Read: Anytime.
Write: Anytime.

Table 2-24. PERT Register Field Descriptions

Field	Description
7-0 PERT	**Port T pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset no pull device is enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.26 Port T Polarity Select Register (PPST)

Address 0x0245 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	PPST7	PPST6	PPST5	PPST4	PPST3	PPST2	PPST1	PPST0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-24. Port T Polarity Select Register (PPST)

[1] Read: Anytime.
Write: Anytime.

Table 2-25. PPST Register Field Descriptions

Field	Description
7-0 PPST	**Port T pull device select**—Determine pull device polarity on input pins This register selects whether a pull-down or a pull-up device is connected to the pin. 1 A pull-down device is connected to the associated pin, if enabled and if the pin is used as input. 0 A pull-up device is connected to the associated pin, if enabled and if the pin is used as input.

2.3.27 PIM Reserved Register

Address 0x0246 Access: User read[1]

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

▨ = Unimplemented or Reserved

Figure 2-25. PIM Reserved Register

[1] Read: Always reads 0x00
Write: Unimplemented

2.3.28 PIM Reserved Register

Address 0x0247 Access: User read[1]

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

▨ = Unimplemented or Reserved

Figure 2-26. PIM Reserved Register

[1] Read: Always reads 0x00
Write: Unimplemented

2.3.29 Port S Data Register (PTS)

Address 0x0248 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PTS7	PTST6	PTS5	PTS4	PTS3	PTS2	PTS1	PTS0
Altern. Function	$\overline{\text{SS0}}$	SCK0	MOSI0	MISO0	TXD1	RXD1	TXD0	RXD0
Reset	0	0	0	0	0	0	0	0

Figure 2-27. Port S Data Register (PTS)

[1] Read: Anytime.
Write: Anytime.

Table 2-26. PTS Register Field Descriptions

Field	Description
7 PTS	**Port S general purpose input/output data**—Data Register Port S pin 7 is associated with the $\overline{\text{SS}}$ signal of the SPI0 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
6 PTS	**Port S general purpose input/output data**—Data Register Port S pin 6 is associated with the SCK signal of the SPI0 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
5 PTS	**Port S general purpose input/output data**—Data Register Port S pin 5 is associated with the MOSI signal of the SPI0 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
4 PTS	**Port S general purpose input/output data**—Data Register Port S pin 4 is associated with the MISO signal of the SPI0 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
3 PTS	**Port S general purpose input/output data**—Data Register Port S pin 3 is associated with the TXD signal of the SCI1 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
2 PTS	**Port S general purpose input/output data**—Data Register Port S bits 2 is associated with the RXD signal of the SCI1 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

Table 2-26. PTS Register Field Descriptions (continued)

Field	Description
1 PTS	**Port S general purpose input/output data**—Data Register Port S pin 3 is associated with the TXD signal of the SCI0 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
0 PTS	**Port S general purpose input/output data**—Data Register Port S bits 2 is associated with the RXD signal of the SCI0 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.30 Port S Input Register (PTIS)

Address 0x0249 Access: User read[1]

	7	6	5	4	3	2	1	0
R	PTIS7	PTIS6	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	u	u	u	u	u	u	u	u

▨ = Unimplemented or Reserved u = Unaffected by reset

Figure 2-28. Port S Input Register (PTIS)

1. Read: Anytime.
   Write:Never, writes to this register have no effect.

Table 2-27. PTIS Register Field Descriptions

Field	Description
7-0 PTIS	**Port S input data**— This register always reads back the buffered state of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

2.3.31 Port S Data Direction Register (DDRS)

Address 0x024A Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRS7	DDRS6	DDRS5	DDRS4	DDRS3	DDRS2	DDRS1	DDRS0
Reset	0	0	0	0	0	0	0	0

Figure 2-29. Port S Data Direction Register (DDRS)

1. Read: Anytime.
   Write: Anytime.

Table 2-28. DDRS Register Field Descriptions

Field	Description
7-0 DDRS	**Port S data direction**— This register controls the data direction of pins 7 through 0.This register configures each Port S pin as either input or output. If SPI0 is enabled, the SPI0 determines the pin direction. *Refer to SPI section for details*. If the associated SCI transmit or receive channel is enabled this register has no effect on the pins. The pin is forced to be an output if a SCI transmit channel is enabled, it is forced to be an input if the SCI receive channel is enabled. The data direction bits revert to controlling the I/O direction of a pin when the associated channel is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTS or PTIS registers, when changing the DDRS register.

2.3.32 Port S Reduced Drive Register (RDRS)

Address 0x024B

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDRS7	RDRS6	RDRS5	RDRS4	RDRS3	RDRS2	RDRS1	RDRS0
Reset	0	0	0	0	0	0	0	0

Figure 2-30. Port S Reduced Drive Register (RDRS)

[1] Read: Anytime.
Write: Anytime.

Table 2-29. RDRS Register Field Descriptions

Field	Description
7-0 RDRS	**Port S reduced drive**—Select reduced drive for outputs This register configures the drive strength of output pins 7 through 0 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.33 Port S Pull Device Enable Register (PERS)

Address 0x024C — Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PERS7	PERS6	PERS5	PERS4	PERS3	PERS2	PERS1	PERS0
Reset	1	1	1	1	1	1	1	1

Figure 2-31. Port S Pull Device Enable Register (PERS)

[1] Read: Anytime.
Write: Anytime.

Table 2-30. PERS Register Field Descriptions

Field	Description
7-0 PERS	**Port S pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset all pull devices are enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.34 Port S Polarity Select Register (PPSS)

Address 0x024D — Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PPSS7	PPSS6	PPSS5	PPSS4	PPSS3	PPSS2	PPSS1	PPSS0
Reset	0	0	0	0	0	0	0	0

Figure 2-32. Port S Polarity Select Register (PPSS)

[1] Read: Anytime.
Write: Anytime.

Table 2-31. PPSS Register Field Descriptions

Field	Description
7-0 PPSS	**Port S pull device select**—Determine pull device polarity on input pins This register selects whether a pull-down or a pull-up device is connected to the pin. 1 A pull-down device is connected to the associated pin, if enabled and if the pin is used as input. 0 A pull-up device is connected to the associated pin, if enabled and if the pin is used as input.

2.3.35 Port S Wired-Or Mode Register (WOMS)

Address 0x024E | Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	WOMS7	WOMS6	WOMS5	WOMS4	WOMS3	WOMS2	WOMS1	WOMS0
Reset	0	0	0	0	0	0	0	0

Figure 2-33. Port S Wired-Or Mode Register (WOMS)

[1] Read: Anytime.
Write: Anytime.

Table 2-32. WOMS Register Field Descriptions

Field	Description
7-0 WOMS	**Port S wired-or mode**—Enable wired-or functionality This register configures the output pins as wired-or independent of the function used on the pins. If enabled the output is driven active low only (open-drain). A logic level of “1” is not driven.This allows a multipoint connection of several serial modules. These bits have no influence on pins used as inputs. 1 Output buffers operate as open-drain outputs. 0 Output buffers operate as push-pull outputs.

2.3.36 PIM Reserved Register

Address 0x024F | Access: User read[1]

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

▨ = Unimplemented or Reserved | u = Unaffected by reset

Figure 2-34. PIM Reserved Register

[1] Read: Always reads 0x00
Write: Unimplemented

2.3.37 Port M Data Register (PTM)

Address 0x0250 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PTM7	PTM6	PTM5	PTM4	PTM3	PTM2	PTM1	PTM0
Altern. Function	TXCAN3	RXCAN3	TXCAN2	RXCAN2	TXCAN1	RXCAN1	TXCAN0	RXCAN0
	—	—	(TXCAN0)	(RXCAN0)	(TXCAN0)	(RXCAN0)	—	—
	(TXCAN4)	(RXCAN4)	(TXCAN4)	(RXCAN4)	—	—	—	—
	—	—	(SCK0)	(MOSI0)	($\overline{\text{SS0}}$)	(MISO0)	—	—
	TXD3	RXD3	—	—	—	—	—	—
Reset	0	0	0	0	0	0	0	0

Figure 2-35. Port M Data Register (PTM)

1. Read: Anytime.
   Write: Anytime.

Table 2-33. PTM Register Field Descriptions

Field	Description
7-6 PTM	**Port M general purpose input/output data**—Data Register Port M pins 7 and 6 are associated with TXCAN and RXCAN signals of CAN3 and the routed CAN4, as well as with TXD and RXD signals of SCI3, respectively. The CAN3 function takes precedence over the CAN4, SCI3 and the general purpose I/O function if the CAN3 module is enabled. The CAN4 function takes precedence over the SCI3 and the general purpose I/O function if the CAN4 module is enabled. The SCI3 function takes precedence over the general purpose I/O function if the SCI3 module is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
5 PTM	**Port M general purpose input/output data**—Data Register Port M pin 5 is associated with the TXCAN signal of CAN2 and the routed CAN4 and CAN0, as well as with SCK signals of SPI0. The CAN2 function takes precedence over the routed CAN0, routed CAN4, the routed SPI0 and the general purpose I/O function if the CAN2 module is enabled. The routed CAN0 function takes precedence over the routed CAN4, the routed SPI0 and the general purpose I/O function if the routed CAN0 module is enabled. The routed CAN4 function takes precedence over the routed SPI0 and general purpose I/O function if the routed CAN4 module is enabled. The routed SPI0 function takes precedence of the general purpose I/O function if the routed SPI0 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

Table 2-33. PTM Register Field Descriptions (continued)

Field	Description
4 PTM	**Port M general purpose input/output data**—Data Register Port M pin 4 is associated with the RXCAN signal of CAN2 and the routed CAN4 and CAN0, as well as with MOSI signals of SPI0. The CAN2 function takes precedence over the routed CAN0, routed CAN4, the routed SPI0 and the general purpose I/O function if the CAN2 module is enabled. The routed CAN0 function takes precedence over the routed CAN4, the routed SPI0 and the general purpose I/O function if the routed CAN0 module is enabled. The routed CAN4 function takes precedence over the routed SPI0 and general purpose I/O function if the routed CAN4 module is enabled. The routed SPI0 function takes precedence of the general purpose I/O function if the routed SPI0 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
3 PTM	**Port M general purpose input/output data**—Data Register Port M pin 5 is associated with the TXCAN signal of CAN1 and the routed CAN0, as well as with $\overline{\text{SS0}}$ signals of SPI0. The CAN1 function takes precedence over the routed CAN0, the routed SPI0 and the general purpose I/O function if the CAN1 module is enabled. The routed CAN0 function takes precedence over the routed SPI0 and the general purpose I/O function if the routed CAN0 module is enabled. The routed SPI0 function takes precedence of the general purpose I/O function if the routed SPI0 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
2 PTM	**Port M general purpose input/output data**—Data Register Port M pin 4 is associated with the RXCAN signal of CAN1 and the routed CAN0, as well as with MISO signals of SPI0. The CAN1 function takes precedence over the routed CAN0, the routed SPI0 and the general purpose I/O function if the CAN1 module is enabled. The routed CAN0 function takes precedence over the routed SPI0 and the general purpose I/O function if the routed CAN0 module is enabled. The routed SPI0 function takes precedence of the general purpose I/O function if the routed SPI0 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
1-0 PTM	**Port M general purpose input/output data**—Data Register Port M pins 1 and 0 are associated with TXCAN and RXCAN signals of CAN0, respectively. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.38 Port M Input Register (PTIM)

Address 0x0251 Access: User read[1]

	7	6	5	4	3	2	1	0
R	PTIM7	PTIM6	PTIM5	PTIM4	PTIM3	PTIM2	PTIM1	PTIM0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	u	u	u	u	u	u	u	u

▨ = Unimplemented or Reserved u = Unaffected by reset

Figure 2-36. Port M Input Register (PTIM)

1. Read: Anytime.
   Write:Never, writes to this register have no effect.

Table 2-34. PTIM Register Field Descriptions

Field	Description
7-0 PTIM	**Port M input data—** This register always reads back the buffered state of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

2.3.39 Port M Data Direction Register (DDRM)

Address 0x0252 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRM7	DDRM6	DDRM5	DDRM4	DDRM3	DDRM2	DDRM1	DDRM0
Reset	0	0	0	0	0	0	0	0

Figure 2-37. Port M Data Direction Register (DDRM)

1. Read: Anytime.
   Write: Anytime.

Table 2-35. DDRM Register Field Descriptions

Field	Description
7 DDRM	**Port M data direction—** This register controls the data direction of pin 7. The enabled CAN3, routed CAN4, or routed SCI3 forces the I/O state to be an output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
6 DDRM	**Port M data direction—** This register controls the data direction of pin 6. The enabled CAN3, routed CAN4, or routed SCI3 forces the I/O state to be an input. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
5 DDRM	**Port M data direction—** This register controls the data direction of pin 5. The enabled CAN2, routed CAN0, or routed CAN4 forces the I/O state to be an output. Depending on the configuration of the enabled routed SPI0 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

Table 2-35. DDRM Register Field Descriptions (continued)

Field	Description
4 DDRM	**Port M data direction—** This register controls the data direction of pin 4. The enabled CAN2, routed CAN0, or routed CAN4 forces the I/O state to be an input. Depending on the configuration of the enabled routed SPI0 this pin will be forced to be input or output.In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
3 DDRM	**Port M data direction—** This register controls the data direction of pin 3. The enabled CAN1 or routed CAN0 forces the I/O state to be an output. Depending on the configuration of the enabled routed SPI0 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
2 DDRM	**Port M data direction—** This register controls the data direction of pin 2. The enabled CAN1 or routed CAN0 forces the I/O state to be an input. Depending on the configuration of the enabled routed SPI0 this pin will be forced to be input or output.In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
1 DDRM	**Port M data direction—** This register controls the data direction of pin 1. The enabled CAN0 forces the I/O state to be an output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
0 DDRM	**Port M data direction—** This register controls the data direction of pin 0. The enabled CAN0 forces the I/O state to be an input. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTM or PTIM registers, when changing the DDRM register.

2.3.40 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 2-38. Port M Reduced Drive Register (RDRM)

[1] Read: Anytime.
Write: Anytime.

Table 2-36. RDRM Register Field Descriptions

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

2.3.41 Port M Pull Device Enable Register (PERM)

Address 0x0254 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PERM7	PERM6	PERM5	PERM4	PERM3	PERM2	PERM1	PERM0
Reset	0	0	0	0	0	0	0	0

Figure 2-39. Port M Pull Device Enable Register (PERM)

[1] Read: Anytime.
Write: Anytime.

Table 2-37. PERM Register Field Descriptions

Field	Description
7-0 PERM	**Port M pull device enable**—Enable pull-up devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input or wired-or output. This bit has no effect if the pin is used as push-pull output. Out of reset no pull device is enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.42 Port M Polarity Select Register (PPSM)

Address 0x0255 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PPSM7	PPSM6	PPSM5	PPSM4	PPSM3	PPSM2	PPSM1	PPSM0
Reset	0	0	0	0	0	0	0	0

Figure 2-40. Port M Polarity Select Register (PPSM)

[1] Read: Anytime.
Write: Anytime.

Table 2-38. PPSM Register Field Descriptions

Field	Description
7-0 PPSM	**Port M pull device select**—Determine pull device polarity on input pins This register selects whether a pull-down or a pull-up device is connected to the pin. If CAN is active a pull-up device can be activated on the RXCAN[3:0] inputs, but not a pull-down. 1 A pull-down device is connected to the associated Port M pin, if enabled by the associated bit in register PERM and if the port is used as a general purpose but not as RXCAN. 0 A pull-up device is connected to the associated Port M pin, if enabled by the associated bit in register PERM and if the port is used as general purpose or RXCAN input.

2.3.43 Port M Wired-Or Mode Register (WOMM)

Address 0x0256 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	WOMM7	WOMM6	WOMM5	WOMM4	WOMM3	WOMM2	WOMM1	WOMM0
Reset	0	0	0	0	0	0	0	0

Figure 2-41. Port M Wired-Or Mode Register (WOMM)

[1] Read: Anytime.
Write: Anytime.

Table 2-39. WOMM Register Field Descriptions

Field	Description
7-0 WOMM	**Port M wired-or mode**—Enable wired-or functionality This register configures the output pins as wired-or independent of the function used on the pins. If enabled the output is driven active low only (open-drain). A logic level of “1” is not driven.This allows a multipoint connection of several serial modules. These bits have no influence on pins used as inputs. 1 Output buffers operate as open-drain outputs. 0 Output buffers operate as push-pull outputs.

2.3.44 Module Routing Register (MODRR)

Address 0x0257 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	MODRR6	MODRR5	MODRR4	MODRR3	MODRR2	MODRR1	MODRR0
W	▨							
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 2-42. Module Routing Register (MODRR)

[1] Read: Anytime.
Write: Anytime.

This register configures the re-routing of CAN0, CAN4, SPI0, SPI1, and SPI2 on alternative ports.

Table 2-40. Module Routing Summary

<table>
<tr><th>Module</th><th colspan="7">MODRR</th><th colspan="4">Related Pins</th></tr>
<tr><td></td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td colspan="4"></td></tr>
<tr><td colspan="8"></td><td colspan="2"><b>RXCAN</b></td><td colspan="2"><b>TXCAN</b></td></tr>
<tr><td>CAN0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td><td>0</td><td colspan="2">PM0</td><td colspan="2">PM1</td></tr>
<tr><td></td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td><td>1</td><td colspan="2">PM2</td><td colspan="2">PM3</td></tr>
<tr><td></td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>1</td><td>0</td><td colspan="2">PM4</td><td colspan="2">PM5</td></tr>
<tr><td></td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>1</td><td>1</td><td colspan="2">PJ6</td><td colspan="2">PJ7</td></tr>
<tr><td>CAN4</td><td>x</td><td>x</td><td>x</td><td>0</td><td>0</td><td>x</td><td>x</td><td colspan="2">PJ6</td><td colspan="2">PJ7</td></tr>
<tr><td></td><td>x</td><td>x</td><td>x</td><td>0</td><td>1</td><td>x</td><td>x</td><td colspan="2">PM4</td><td colspan="2">PM5</td></tr>
<tr><td></td><td>x</td><td>x</td><td>x</td><td>1</td><td>0</td><td>x</td><td>x</td><td colspan="2">PM6</td><td colspan="2">PM7</td></tr>
<tr><td></td><td>x</td><td>x</td><td>x</td><td>1</td><td>1</td><td>x</td><td>x</td><td colspan="4">Reserved</td></tr>
<tr><td colspan="8"></td><td><b>MISO</b></td><td><b>MOSI</b></td><td><b>SCK</b></td><td><b>$\overline{\text{SS}}$</b></td></tr>
<tr><td>SPI0</td><td>x</td><td>x</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>PS4</td><td>PS5</td><td>PS6</td><td>PS7</td></tr>
<tr><td></td><td>x</td><td>x</td><td>1</td><td>x</td><td>x</td><td>x</td><td>x</td><td>PM2</td><td>PM4</td><td>PM5</td><td>PM3</td></tr>
<tr><td>SPI1</td><td>x</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>PP0</td><td>PP1</td><td>PP2</td><td>PP3</td></tr>
<tr><td></td><td>x</td><td>1</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>PH0</td><td>PH1</td><td>PH2</td><td>PH3</td></tr>
<tr><td>SPI2</td><td>0</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>PP4</td><td>PP5</td><td>PP7</td><td>PP6</td></tr>
<tr><td></td><td>1</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>PH4</td><td>PH5</td><td>PH6</td><td>PH7</td></tr>
</table>

2.3.45 Port P Data Register (PTP)

Address 0x0258 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PTP7	PTP6	PTP5	PTP4	PTP3	PTP2	PTP1	PTP0
Altern. Function	PWM7	PWM6	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0
	SCK2	$\overline{\text{SS2}}$	MOSI2	MISO2	$\overline{\text{SS1}}$	SCK1	MOSI1	MISO1
Reset	0	0	0	0	0	0	0	0

Figure 2-43. Port P Data Register (PTP)

[1] Read: Anytime.
Write: Anytime.

Table 2-41. PTP Register Field Descriptions

Field	Description
7 PTP	**Port P general purpose input/output data**—Data Register Port P pin 6 is associated with the PWM output channel 7 and the SCK signal of SPI2. The PWM function takes precedence over the SPI2 and the general purpose I/O function if the PWM channel 7 is enabled. The SPI2 function takes precedence of the general purpose I/O function if the routed SPI2 is enabled. When not used with the alternative functions, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
6 PTP	**Port P general purpose input/output data**—Data Register Port P pin 6 is associated with the PWM output channel 6 and the $\overline{\text{SS}}$ signal of SPI2. The PWM function takes precedence over the SPI2 and the general purpose I/O function if the PWM channel 6 is enabled. The SPI2 function takes precedence of the general purpose I/O function if the routed SPI2 is enabled. When not used with the alternative functions, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
5 PTP	**Port P general purpose input/output data**—Data Register Port P pin 5 is associated with the PWM output channel 5 and the MOSI signal of SPI2. The PWM function takes precedence over the SPI2 and the general purpose I/O function if the PWM channel 5 is enabled. The SPI2 function takes precedence of the general purpose I/O function if the routed SPI2 is enabled. When not used with the alternative functions, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
4 PTP	**Port P general purpose input/output data**—Data Register Port P pin 4 is associated with the PWM output channel 4 and the MISO signal of SPI2. The PWM function takes precedence over the SPI2 and the general purpose I/O function if the PWM channel 4 is enabled. The SPI2 function takes precedence of the general purpose I/O function if the routed SPI2 is enabled. When not used with the alternative functions, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

Table 2-41. PTP Register Field Descriptions (continued)

Field	Description
3 PTP	**Port P general purpose input/output data**—Data Register Port P pin 3 is associated with the PWM output channel 3 and the $\overline{SS}$ signal of SPI1. The PWM function takes precedence over the SPI1 and the general purpose I/O function if the PWM channel 3 is enabled. The SPI1 function takes precedence of the general purpose I/O function if the routed SPI1 is enabled. When not used with the alternative functions, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
2 PTP	**Port P general purpose input/output data**—Data Register Port P pin 2 is associated with the PWM output channel 2 and the SCK signal of SPI1. The PWM function takes precedence over the SPI1 and the general purpose I/O function if the PWM channel 2 is enabled. The SPI1 function takes precedence of the general purpose I/O function if the routed SPI1 is enabled. When not used with the alternative functions, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
1 PTP	**Port P general purpose input/output data**—Data Register Port P pin 1 is associated with the PWM output channel 1 and the MOSI signal of SPI1. The PWM function takes precedence over the SPI1 and the general purpose I/O function if the PWM channel 1 is enabled. The SPI1 function takes precedence of the general purpose I/O function if the routed SPI1 is enabled. When not used with the alternative functions, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
0 PTP	**Port P general purpose input/output data**—Data Register Port P pin 0 is associated with the PWM output channel 0 and the MISO signal of SPI1. The PWM function takes precedence over the SPI1 and the general purpose I/O function if the PWM channel 0 is enabled. The SPI1 function takes precedence of the general purpose I/O function if the routed SPI1 is enabled. When not used with the alternative functions, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.46 Port P Input Register (PTIP)

Address 0x0259 Access: User read[1]

	7	6	5	4	3	2	1	0
R	PTIP7	PTIP6	PTIP5	PTIP4	PTIP3	PTIP2	PTIP1	PTIP0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	u	u	u	u	u	u	u	u

▨ = Unimplemented or Reserved u = Unaffected by reset

Figure 2-44. Port P Input Register (PTIP)

[1] Read: Anytime.
Write:Never, writes to this register have no effect.

Table 2-42. PTIP Register Field Descriptions

Field	Description
7-0 PTIP	**Port P input data**— This register always reads back the buffered state of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

2.3.47 Port P Data Direction Register (DDRP)

Address 0x025A Access: User read/write[1]

	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 2-45. Port P Data Direction Register (DDRP)

[1] Read: Anytime.
Write: Anytime.

Table 2-43. DDRP Register Field Descriptions

Field	Description
7 DDRP	**Port P data direction**— This register controls the data direction of pin 7. The enabled PWM channel 7 forces the I/O state to be an output. If the PWM shutdown feature is enabled this pin is forced to be an input. In these cases the data direction bit will not change. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
6-0 DDRP	**Port P data direction**— The PWM forces the I/O state to be an output for each port line associated with an enabled PWM6-0 channel. In this case the data direction bit will not change. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

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.

2.3.48 Port P Reduced Drive Register (RDRP)

Address 0x025B Access: User read/write[1]

	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 2-46. Port P Reduced Drive Register (RDRP)

[1] Read: Anytime.
Write: Anytime.

Table 2-44. RDRP Register Field Descriptions

Field	Description
7-0 RDRP	**Port P reduced drive**—Select reduced drive for outputs This register configures the drive strength of output pins 7 through 0 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.49 Port P Pull Device Enable Register (PERP)

Address 0x025C Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
Reset	0	0	0	0	0	0	0	0

Figure 2-47. Port P Pull Device Enable Register (PERP)

[1] Read: Anytime.
Write: Anytime.

Table 2-45. PERP Register Field Descriptions

Field	Description
7-0 PERP	**Port P pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset no pull device is enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.50 Port P Polarity Select Register (PPSP)

Address 0x025D Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
Reset	0	0	0	0	0	0	0	0

Figure 2-48. Port P Polarity Select Register (PPSP)

[1] Read: Anytime.
Write: Anytime.

Table 2-46. PPSP Register Field Descriptions

Field	Description
7-0 PPSP	**Port P pull device select**—Determine pull device polarity on input pins This register serves a dual purpose by selecting the polarity of the active interrupt edge as well as selecting a pull-up or pull-down device if enabled. 1 A rising edge on the associated Port P pin sets the associated flag bit in the PIFP register. A pull-down device is connected to the associated Port P pin, if enabled by the associated bit in register PERP and if the port is used as input. 0 A falling edge on the associated Port P pin sets the associated flag bit in the PIFP register.A pull-up device is connected to the associated Port P pin, if enabled by the associated bit in register PERP and if the port is used as input.

2.3.51 Port P Interrupt Enable Register (PIEP)

Read: Anytime.

Address 0x025E Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIEP7	PIEP6	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
Reset	0	0	0	0	0	0	0	0

Figure 2-49. Port P Interrupt Enable Register (PIEP)

[1] Read: Anytime.
Write: Anytime.

Table 2-47. PPSP Register Field Descriptions

Field	Description
7-0 PIEP	**Port P interrupt enable**— This register disables or enables on a per-pin basis the edge sensitive external interrupt associated with Port P. 1 Interrupt is enabled. 0 Interrupt is disabled (interrupt flag masked).

2.3.52 Port P Interrupt Flag Register (PIFP)

Address 0x025F | Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIFP7	PIFP6	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
Reset	0	0	0	0	0	0	0	0

Figure 2-50. Port P Interrupt Flag Register (PIFP)

[1] Read: Anytime.
Write: Anytime.

Table 2-48. PPSP Register Field Descriptions

Field	Description
7-0 PIFP	**Port P interrupt flag**— Each flag is set by an active edge on the associated input pin. This could be a rising or a falling edge based on the state of the PPSP register. To clear this flag, write logic level 1 to the corresponding bit in the PIFP register. Writing a 0 has no effect. 1 Active edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set). 0 No active edge pending.

2.3.53 Port H Data Register (PTH)

Address 0x0260 | Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PTH7	PTH6	PTH5	PTH4	PTH3	PTH2	PTH1	PTH0
Altern. Function	$\overline{\text{SS2}}$	SCK2	MOSI2	MISO2	$\overline{\text{SS1}}$	SCK1	MOSI1	MISO1
	TXD5	RXD5	TXD4	RXD4	TXD7	RXD7	TXD6	RXD6
Reset	0	0	0	0	0	0	0	0

Figure 2-51. Port H Data Register (PTH)

[1] Read: Anytime.
Write: Anytime.

Table 2-49. PTH Register Field Descriptions

Field	Description
7 PTH	**Port H general purpose input/output data**—Data Register Port H pin 7 is associated with the TXD signal of the SCI5 module and the $\overline{\text{SS}}$ signal of the routed SPI2. The routed SPI2 function takes precedence over the SCI5 and the general purpose I/O function if the routed SPI2 module is enabled. The SCI5 function takes precedence over the general purpose I/O function if the SCI5 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
6 PTH	**Port H general purpose input/output data**—Data Register Port H pin 6 is associated with the RXD signal of the SCI5 module and the SCK signal of the routed SPI2. The routed SPI2 function takes precedence over the SCI5 and the general purpose I/O function if the routed SPI2 module is enabled. The SCI5 function takes precedence over the general purpose I/O function if the SCI5 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
5 PTH	**Port H general purpose input/output data**—Data Register Port H pin 5 is associated with the TXD signal of the SCI4 module and the MOSI signal of the routed SPI2. The routed SPI2 function takes precedence over the SCI4 and the general purpose I/O function if the routed SPI2 module is enabled. The SCI4 function takes precedence over the general purpose I/O function if the SCI4 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
4 PTH	**Port H general purpose input/output data**—Data Register Port H pin 4 is associated with the RXD signal of the SCI4 module and the MISO signal of the routed SPI2. The routed SPI2 function takes precedence over the SCI4 and the general purpose I/O function if the routed SPI2 module is enabled. The SCI4 function takes precedence over the general purpose I/O function if the SCI4 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
3 PTH	**Port H general purpose input/output data**—Data Register Port H pin 3 is associated with the TXD signal of the SCI7 module and the $\overline{\text{SS}}$ signal of the routed SPI1. The routed SPI1 function takes precedence over the SCI7 and the general purpose I/O function if the routed SPI1 module is enabled. The SCI7 function takes precedence over the general purpose I/O function if the SCI7 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
2 PTH	**Port H general purpose input/output data**—Data Register Port H pin 2 is associated with the RXD signal of the SCI7 module and the SCK signal of the routed SPI1. The routed SPI1 function takes precedence over the SCI7 and the general purpose I/O function if the routed SPI1 module is enabled. The SCI7 function takes precedence over the general purpose I/O function if the SCI7 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

Table 2-49. PTH Register Field Descriptions (continued)

Field	Description
1 PTH	**Port H general purpose input/output data**—Data Register Port H pin 1 is associated with the TXD signal of the SCI6 module and the MOSI signal of the routed SPI1. The routed SPI1 function takes precedence over the SCI6 and the general purpose I/O function if the routed SPI1 module is enabled. The SCI6 function takes precedence over the general purpose I/O function if the SCI6 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
0 PTH	**Port H general purpose input/output data**—Data Register Port H pin 0 is associated with the RXD signal of the SCI6 module and the MISO signal of the routed SPI1. The routed SPI1 function takes precedence over the SCI6 and the general purpose I/O function if the routed SPI1 module is enabled. The SCI6 function takes precedence over the general purpose I/O function if the SCI6 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.54 Port H Input Register (PTIH)

Address 0x0261 Access: User read[1]

	7	6	5	4	3	2	1	0
R	PTIH7	PTIH6	PTIH5	PTIH4	PTIH3	PTIH2	PTIH1	PTIH0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	u	u	u	u	u	u	u	u

▨ = Unimplemented or Reserved u = Unaffected by reset

Figure 2-52. Port H Input Register (PTIH)

[1] Read: Anytime.
Write:Never, writes to this register have no effect.

Table 2-50. PTIH Register Field Descriptions

Field	Description
7-0 PTIH	**Port H input data**— This register always reads back the buffered state of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

2.3.55 Port H Data Direction Register (DDRH)

Address 0x0262 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRH7	DDRH6	DDRH5	DDRH4	DDRH3	DDRH2	DDRH1	DDRH0
Reset	0	0	0	0	0	0	0	0

Figure 2-53. Port H Data Direction Register (DDRH)

1. Read: Anytime.
   Write: Anytime.

Table 2-51. DDRH Register Field Descriptions

Field	Description
7 DDRH	**Port H data direction—** This register controls the data direction of pin 7. The enabled SCI5 forces the I/O state to be an output. Depending on the configuration of the enabled routed SPI2 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
6 DDRH	**Port H data direction—** This register controls the data direction of pin 6. The enabled SCI5 forces the I/O state to be an input. Depending on the configuration of the enabled routed SPI2 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
5 DDRH	**Port H data direction—** This register controls the data direction of pin 5. The enabled SCI4 forces the I/O state to be an output. Depending on the configuration of the enabled routed SPI2 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
4 DDRH	**Port H data direction—** This register controls the data direction of pin 4. The enabled SCI4 forces the I/O state to be an input. Depending on the configuration of the enabled routed SPI2 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
3 DDRH	**Port H data direction—** This register controls the data direction of pin 3. The enabled SCI7 forces the I/O state to be an output. Depending on the configuration of the enabled routed SPI1 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
2 DDRH	**Port H data direction—** This register controls the data direction of pin 2. The enabled SCI7 forces the I/O state to be an input. Depending on the configuration of the enabled routed SPI1 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

Table 2-51. DDRH Register Field Descriptions (continued)

Field	Description
1 DDRH	**Port H data direction—** This register controls the data direction of pin 1. The enabled SCI6 forces the I/O state to be an output. Depending on the configuration of the enabled routed SPI1 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
0 DDRH	**Port H data direction—** This register controls the data direction of pin 0. The enabled SCI6 forces the I/O state to be an input. Depending on the configuration of the enabled routed SPI1 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTH or PTIH registers, when changing the DDRH register.

2.3.56 Port H Reduced Drive Register (RDRH)

Address 0x0263 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDRH7	RDRH6	RDRH5	RDRH4	RDRH3	RDRH2	RDRH1	RDRH0
Reset	0	0	0	0	0	0	0	0

Figure 2-54. Port H Reduced Drive Register (RDRH)

[1] Read: Anytime.
Write: Anytime.

Table 2-52. RDRH Register Field Descriptions

Field	Description
7-0 RDRH	**Port H reduced drive**—Select reduced drive for outputs This register configures the drive strength of output pins 7 through 0 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.57 Port H Pull Device Enable Register (PERH)

Address 0x0264 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PERH7	PERH6	PERH5	PERH4	PERH3	PERH2	PERH1	PERH0
Reset	0	0	0	0	0	0	0	0

Figure 2-55. Port H Pull Device Enable Register (PERH)

[1] Read: Anytime.
Write: Anytime.

Table 2-53. PERH Register Field Descriptions

Field	Description
7-0 PERH	**Port H pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset no pull device is enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.58 Port H Polarity Select Register (PPSH)

Address 0x0265 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PPSH7	PPSH6	PPSH5	PPSH4	PPSH3	PPSH2	PPSH1	PPSH0
Reset	0	0	0	0	0	0	0	0

Figure 2-56. Port H Polarity Select Register (PPSH)

[1] Read: Anytime.
Write: Anytime.

Table 2-54. PPSH Register Field Descriptions

Field	Description
7-0 PPSH	**Port H pull device select**—Determine pull device polarity on input pins This register serves a dual purpose by selecting the polarity of the active interrupt edge as well as selecting a pull-up or pull-down device if enabled. 1 A rising edge on the associated Port H pin sets the associated flag bit in the PIFH register. A pull-down device is connected to the associated Port H pin, if enabled by the associated bit in register PERH and if the port is used as input. 0 A falling edge on the associated Port H pin sets the associated flag bit in the PIFH register.A pull-up device is connected to the associated Port H pin, if enabled by the associated bit in register PERH and if the port is used as input.

2.3.59 Port H Interrupt Enable Register (PIEH)

Read: Anytime.

Address 0x0266 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIEH7	PIEH6	PIEH5	PIEH4	PIEH3	PIEH2	PIEH1	PIEH0
Reset	0	0	0	0	0	0	0	0

Figure 2-57. Port H Interrupt Enable Register (PIEH)

[1] Read: Anytime.
Write: Anytime.

Table 2-55. PPSP Register Field Descriptions

Field	Description
7-0 PIEH	**Port H interrupt enable**— This register disables or enables on a per-pin basis the edge sensitive external interrupt associated with Port H. 1 Interrupt is enabled. 0 Interrupt is disabled (interrupt flag masked).

2.3.60 Port H Interrupt Flag Register (PIFH)

Address 0x0267 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIFH7	PIFH6	PIFH5	PIFH4	PIFH3	PIFH2	PIFH1	PIFH0
Reset	0	0	0	0	0	0	0	0

Figure 2-58. Port H Interrupt Flag Register (PIFH)

[1] Read: Anytime.
Write: Anytime.

Table 2-56. PPSP Register Field Descriptions

Field	Description
7-0 PIFH	**Port H interrupt flag**— Each flag is set by an active edge on the associated input pin. This could be a rising or a falling edge based on the state of the PPSH register. To clear this flag, write logic level 1 to the corresponding bit in the PIFH register. Writing a 0 has no effect. 1 Active edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set). 0 No active edge pending.

2.3.61 Port J Data Register (PTJ)

Address 0x0268 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PTJ7	PTJ6	PTJ5	PTJ4	PTJ3	PTJ2	PTJ1	PTJ0
Altern. Function	TXCAN4	RXCAN4	—	—	—	—	TXD2	RXD2
	SCL0	SDA0	SCL1	SDA1	—	—	—	—
	(TXCAN0)	(RXCAN0)	$\overline{\text{CS2}}$	$\overline{\text{CS0}}$	—	$\overline{\text{CS1}}$	—	$\overline{\text{CS3}}$
Reset	0	0	0	0	0	0	0	0

Figure 2-59. Port J Data Register (PTJ)

1. Read: Anytime.
   Write: Anytime.

Table 2-57. PTJ Register Field Descriptions

Field	Description
7-6 PTJ	**Port J general purpose input/output data**—Data Register Port J pins 7 and 6 are associated with TXCAN and RXCAN signals of CAN4 and the routed CAN0, as well as with SCL and SDA signals of IIC0, respectively. The CAN4 function takes precedence over the IIC0, the routed CAN0 and the general purpose I/O function if the CAN4 module is enabled. The IIC0 function takes precedence over the routed CAN0 and the general purpose I/O function if the IIC0 is enabled. If the IIC0 module takes precedence the SDA0 and SCL0 outputs are configured as open drain outputs. The routed CAN0 function takes precedence over the general purpose I/O function if the routed CAN0 module is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
5-4 PTJ	**Port J general purpose input/output data**—Data Register This pin is associated with the SCL and SDA signals of IIC1, and with chip select outputs $\overline{\text{CS2}}$ and $\overline{\text{CS0}}$, respectively. The IIC1 function takes precedence over the chip select and general purpose I/O function if the IIC1 is enabled. The chip selects take precedence over the general purpose I/O. If the IIC1 module takes precedence the SDA1 and SCL1 outputs are configured as open drain outputs. Refer to IIC section for details. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
3 PTJ	**Port J general purpose input/output data**—Data Register This pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
2 PTJ	**Port J general purpose input/output data**—Data Register This pin is associated with the chip select output signal $\overline{\text{CS2}}$. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

Table 2-57. PTJ Register Field Descriptions (continued)

Field	Description
1 PTJ	**Port J general purpose input/output data**—Data Register This pin is associated with the TXD signal of SCI2. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
0 PTJ	**Port J general purpose input/output data**—Data Register This pin is associated with the TXD signal of SCI2 and chip select output $\overline{\text{CS3}}$. The SCI function takes precedence over the chip select and general purpose I/O function if the SCI2 is enabled. The chip select takes precedence over the general purpose I/O. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.62 Port J Input Register (PTIJ)

Address 0x0269 Access: User read[1]

	7	6	5	4	3	2	1	0
R	PTIJ7	PTIJ6	PTIJ5	PTIJ4	PTIJ3	PTIJ2	PTIJ1	PTIJ0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	u	u	u	u	u	u	u	u

▨ = Unimplemented or Reserved u = Unaffected by reset

Figure 2-60. Port J Input Register (PTIJ)

[1] Read: Anytime.
Write:Never, writes to this register have no effect.

Table 2-58. PTIJ Register Field Descriptions

Field	Description
7-0 PTIJ	**Port J input data**— This register always reads back the buffered state of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

2.3.63 Port J Data Direction Register (DDRJ)

Address 0x026A Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRJ7	DDRJ6	DDRJ5	DDRJ4	DDRJ3	DDRJ2	DDRJ1	DDRJ0
Reset	0	0	0	0	0	0	0	0

Figure 2-61. Port J Data Direction Register (DDRJ)

1. Read: Anytime.
   Write: Anytime.

Table 2-59. DDRJ Register Field Descriptions

Field	Description
7 DDRJ	**Port J data direction**— This register controls the data direction of pin 7. The enabled CAN4 or routed CAN0 forces the I/O state to be an output. The enabled IIC0 module forces this pin to be a open drain output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
6 DDRJ	**Port J data direction**— This register controls the data direction of pin 6. The enabled CAN4 or routed CAN0 forces the I/O state to be an input. The enabled IIC0 module forces this pin to be a open drain output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
5 DDRJ	**Port J data direction**— This register controls the data direction of pin 5. The enabled $\overline{\text{CS2}}$ signal forces the I/O state to be an output. The enabled IIC1 module forces this pin to be a open drain output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
4 DDRJ	**Port J data direction**— This register controls the data direction of pin 4. The enabled $\overline{\text{CS0}}$ signal forces the I/O state to be an output. The enabled IIC1 module forces this pin to be a open drain output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
3 DDRJ	**Port J data direction**— This register controls the data direction of pin 3. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
2 DDRJ	**Port J data direction**— This register controls the data direction of pin 2. The enabled $\overline{\text{CS1}}$ signal forces the I/O state to be an output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

Table 2-59. DDRJ Register Field Descriptions (continued)

Field	Description
1 DDRJ	**Port J data direction—** This register controls the data direction of pin 1. The enabled SCI2 forces the I/O state to be an output. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
0 DDRJ	**Port J data direction—** This register controls the data direction of pin 0. The enabled SCI3 or $\overline{\text{CS3}}$ signal forces the I/O state to be an output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTH or PTIH registers, when changing the DDRH register.

2.3.64 Port J Reduced Drive Register (RDRJ)

Address 0x026B

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDRJ7	RDRJ6	RDRJ5	RDRJ4	RDRJ3	RDRJ2	RDRJ1	RDRJ0
Reset	0	0	0	0	0	0	0	0

Figure 2-62. Port J Reduced Drive Register (RDRJ)

[1] Read: Anytime.
Write: Anytime.

Table 2-60. RDRJ Register Field Descriptions

Field	Description
7-0 RDRJ	**Port J reduced drive**—Select reduced drive for outputs This register configures the drive strength of output pins 7 through 0 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.65 Port J Pull Device Enable Register (PERJ)

Address 0x026C

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PERJ7	PERJ6	PERJ5	PERJ4	PERJ3	PERJ2	PERJ1	PERJ0
Reset	1	1	1	1	1	1	1	1

Figure 2-63. Port J Pull Device Enable Register (PERJ)

[1] Read: Anytime.
Write: Anytime.

Table 2-61. PERJ Register Field Descriptions

Field	Description
7-0 PERJ	**Port J pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset all pull device are enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.66 Port J Polarity Select Register (PPSJ)

Address 0x026D

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PPSJ7	PPSJ6	PPSJ5	PPSJ4	PPSJ3	PPSJ2	PPSJ1	PPSJ0
Reset	0	0	0	0	0	0	0	0

Figure 2-64. Port J Polarity Select Register (PPSJ)

[1] Read: Anytime.
Write: Anytime.

Table 2-62. PPSJ Register Field Descriptions

Field	Description
7-0 PPSJ	**Port J pull device select**—Determine pull device polarity on input pins This register serves a dual purpose by selecting the polarity of the active interrupt edge as well as selecting a pull-up or pull-down device if enabled. 1 A rising edge on the associated Port J pin sets the associated flag bit in the PIFJ register. A pull-down device is connected to the associated Port J pin, if enabled by the associated bit in register PERJ and if the port is used as input. 0 A falling edge on the associated Port J pin sets the associated flag bit in the PIFJ register.A pull-up device is connected to the associated Port J pin, if enabled by the associated bit in register PERJ and if the port is used as input.

2.3.67 Port J Interrupt Enable Register (PIEJ)

Read: Anytime.

Address 0x026E Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIEJ7	PIEJ6	PIEJ5	PIEJ4	PIEJ3	PIEJ2	PIEJ1	PIEJ0
Reset	0	0	0	0	0	0	0	0

Figure 2-65. Port J Interrupt Enable Register (PIEJ)

[1] Read: Anytime.
Write: Anytime.

Table 2-63. PPSP Register Field Descriptions

Field	Description
7-0 PIEJ	**Port J interrupt enable**— This register disables or enables on a per-pin basis the edge sensitive external interrupt associated with Port J. 1 Interrupt is enabled. 0 Interrupt is disabled (interrupt flag masked).

2.3.68 Port J Interrupt Flag Register (PIFJ)

Address 0x026F Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIFJ7	PIFJ6	PIFJ5	PIFJ4	PIFJ3	PIFJ2	PIFJ1	PIFJ0
Reset	0	0	0	0	0	0	0	0

Figure 2-66. Port J Interrupt Flag Register (PIFJ)

[1] Read: Anytime.
Write: Anytime.

Table 2-64. PPSP Register Field Descriptions

Field	Description
7-0 PIFJ	**Port J interrupt flag**— Each flag is set by an active edge on the associated input pin. This could be a rising or a falling edge based on the state of the PPSJ register. To clear this flag, write logic level 1 to the corresponding bit in the PIFJ register. Writing a 0 has no effect. 1 Active edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set). 0 No active edge pending.

2.3.69 Port AD0 Data Register 0 (PT0AD0)

Address 0x0270 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PT0AD07	PT0AD06	PT0AD05	PT0AD04	PT0AD03	PT0AD02	PT0AD01	PT0AD00
Altern. Function	AN15	AN14	AN13	AN12	AN11	AN10	AN9	AN8
Reset	0	0	0	0	0	0	0	0

Figure 2-67. Port AD0 Data Register 0 (PT0AD0)

1. Read: Anytime.
Write: Anytime.

Table 2-65. PT0AD0 Register Field Descriptions

Field	Description
7-0 PT0AD0	**Port AD0 general purpose input/output data**—Data Register This register is associated with ATD0 analog inputs AN[15:8] on PAD[15:8], respectively. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.70 Port AD0 Data Register 1 (PT1AD0)

Address 0x0271 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PT1AD07	PT1AD06	PT1AD05	PT1AD04	PT1AD03	PT1AD02	PT1AD01	PT1AD00
Altern. Function	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
Reset	0	0	0	0	0	0	0	0

Figure 2-68. Port AD0 Data Register 1 (PT1AD0)

1. Read: Anytime.
Write: Anytime.

Table 2-66. PT1AD0 Register Field Descriptions

Field	Description
7-0 PT1AD0	**Port AD0 general purpose input/output data**—Data Register This register is associated with ATD0 analog inputs AN[7:0] on PAD[7:0], respectively. When not used with the alternative function, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.71 Port AD0 Data Direction Register 0 (DDR0AD0)

Address 0x0272 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDR0AD07	DDR0AD06	DDR0AD05	DDR0AD04	DDR0AD03	DDR0AD02	DDR0AD01	DDR0AD00
Reset	0	0	0	0	0	0	0	0

Figure 2-69. Port AD0 Data Direction Register 0 (DDR0AD0)

[1] Read: Anytime.
Write: Anytime.

Table 2-67. DDR0AD0 Register Field Descriptions

Field	Description
7-0 DDR0AD0	**Port AD0 data direction—** This register controls the data direction of pins 15 through 8. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PT0AD0 registers, when changing the DDR0AD0 register.

NOTE

To use the digital input function on Port AD0 the ATD Digital Input Enable Register (ATD0DIEN1) has to be set to logic level “1”.

2.3.72 Port AD0 Data Direction Register 1 (DDR1AD0)

Address 0x0273 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDR1AD07	DDR1AD06	DDR1AD05	DDR1AD04	DDR1AD03	DDR1AD02	DDR1AD01	DDR1AD00
Reset	0	0	0	0	0	0	0	0

Figure 2-70. Port AD0 Data Direction Register 1 (DDR1AD0)

[1] Read: Anytime.
Write: Anytime.

Table 2-68. DDR1AD0 Register Field Descriptions

Field	Description
7-0 DDR1AD0	**Port AD0 data direction—** This register controls the data direction of pins 7 through 0. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PT0AD0 registers, when changing the DDR1AD0 register.

NOTE

To use the digital input function on Port AD0 the ATD Digital Input Enable Register (ATD0DIEN1) has to be set to logic level “1”.

2.3.73 Port AD0 Reduced Drive Register 0 (RDR0AD0)

Address 0x0274 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDR0AD07	RDR0AD06	RDR0AD05	RDR0AD04	RDR0AD03	RDR0AD02	RDR0AD01	RDR0AD00
Reset	0	0	0	0	0	0	0	0

Figure 2-71. Port AD0 Reduced Drive Register 0 (RDR0AD0)

[1] Read: Anytime.
Write: Anytime.

Table 2-69. RDR0AD0 Register Field Descriptions

Field	Description
7-0 RDR0AD0	**Port AD0 reduced drive**—Select reduced drive for Port AD0 outputs This register configures the drive strength of Port AD0 output pins 15 through 8 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.74 Port AD0 Reduced Drive Register 1 (RDR1AD0)

Address 0x0275 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDR1AD07	RDR1AD06	RDR1AD05	RDR1AD04	RDR1AD03	RDR1AD02	RDR1AD01	RDR1AD00
Reset	0	0	0	0	0	0	0	0

Figure 2-72. Port AD0 Reduced Drive Register 1 (RDR1AD0)

[1] Read: Anytime.
Write: Anytime.

Table 2-70. RDR1AD0 Register Field Descriptions

Field	Description
7-0 RDR1AD0	**Port AD0 reduced drive**—Select reduced drive for Port AD0 outputs This register configures the drive strength of Port AD0 output pins 7 through 0 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.75 Port AD0 Pull Up Enable Register 0 (PER0AD0)

Address 0x0276 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PER0AD07	PER0AD06	PER0AD05	PER0AD04	PER0AD03	PER0AD02	PER0AD01	PER0AD00
Reset	0	0	0	0	0	0	0	0

Figure 2-73. Port AD0 Pull Device Up Register 0 (PER0AD0)

[1] Read: Anytime.
Write: Anytime.

Table 2-71. PER0AD0 Register Field Descriptions

Field	Description
7-0 PER0AD0	**Port AD0 pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset no pull device is enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.76 Port AD0 Pull Up Enable Register 1 (PER1AD0)

Address 0x0277 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PER1AD07	PER1AD06	PER1AD05	PER1AD04	PER1AD03	PER1AD02	PER1AD01	PER1AD00
Reset	0	0	0	0	0	0	0	0

[1] Read: Anytime.
Write: Anytime.

Figure 2-74. Port AD0 Pull Up Enable Register 1 (PER1AD0)

Table 2-72. PER1AD0 Register Field Descriptions

Field	Description
7-0 PER1AD0	**Port AD0 pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset no pull device is enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.77 Port AD1 Data Register 0 (PT0AD1)

Address 0x0278 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PT0AD17	PT0AD16	PT0AD15	PT0AD14	PT0AD13	PT0AD12	PT0AD11	PT0AD10
Altern. Function	AN15	AN14	AN13	AN12	AN11	AN10	AN9	AN8
Reset	0	0	0	0	0	0	0	0

[1] Read: Anytime.
Write: Anytime.

Figure 2-75. Port AD1 Data Register 0 (PT0AD1)

Table 2-73. PT0AD1 Register Field Descriptions

Field	Description
7-0 PT0AD1	**Port AD1 general purpose input/output data**—Data Register This register is associated with ATD1 analog inputs AN[15:8] on PAD[31:24], respectively. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.78 Port AD1 Data Register 1 (PT1AD1)

Address 0x0279 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PT1AD17	PT1AD16	PT1AD15	PT1AD14	PT1AD13	PT1AD12	PT1AD11	PT1AD10
Altern. Function	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
Reset	0	0	0	0	0	0	0	0

Figure 2-76. Port AD1 Data Register 1 (PT1AD1)

[1] Read: Anytime.
Write: Anytime.

Table 2-74. PT1AD1 Register Field Descriptions

Field	Description
7-0 PT1AD1	**Port AD1 general purpose input/output data**—Data Register This register is associated with ATD1 analog inputs AN[7:0] on PAD[23:16], respectively. When not used with the alternative function, these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.79 Port AD1 Data Direction Register 0 (DDR0AD1)

Address 0x027A Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDR0AD17	DDR0AD16	DDR0AD15	DDR0AD14	DDR0AD13	DDR0AD12	DDR0AD11	DDR0AD10
Reset	0	0	0	0	0	0	0	0

Figure 2-77. Port AD1 Data Direction Register 0 (DDR0AD1)

[1] Read: Anytime.
Write: Anytime.

Table 2-75. DDR0AD1 Register Field Descriptions

Field	Description
7-0 DDR0AD1	**Port AD1 data direction**— This register controls the data direction of pins 15 through 8. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PT0AD1 registers, when changing the DDR0AD1 register.

NOTE

To use the digital input function on Port AD1 the ATD Digital Input Enable Register (ATD1DIEN1) has to be set to logic level “1”.

2.3.80 Port AD1 Data Direction Register 1 (DDR1AD1)

Address 0x027B

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDR1AD17	DDR1AD16	DDR1AD15	DDR1AD14	DDR1AD13	DDR1AD12	DDR1AD11	DDR1AD10
Reset	0	0	0	0	0	0	0	0

Figure 2-78. Port AD1 Data Direction Register 1 (DDR1AD1)

[1] Read: Anytime.
Write: Anytime.

Table 2-76. DDR1AD1 Register Field Descriptions

Field	Description
7-0 DDR1AD1	**Port AD1 data direction—** This register controls the data direction of pins 7 through 0. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PT0AD1 registers, when changing the DDR1AD1 register.

NOTE

To use the digital input function on Port AD1 the ATD Digital Input Enable Register (ATD1DIEN1) has to be set to logic level “1”.

2.3.81 Port AD1 Reduced Drive Register 0 (RDR0AD1)

Address 0x027C Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDR0AD17	RDR0AD16	RDR0AD15	RDR0AD14	RDR0AD13	RDR0AD12	RDR0AD11	RDR0AD10
Reset	0	0	0	0	0	0	0	0

Figure 2-79. Port AD1 Reduced Drive Register 0 (RDR0AD1)

[1] Read: Anytime.
Write: Anytime.

Table 2-77. RDR0AD1 Register Field Descriptions

Field	Description
7-0 RDR0AD1	**Port AD1 reduced drive**—Select reduced drive for Port AD1 outputs This register configures the drive strength of Port AD1 output pins 15 through 8 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.82 Port AD1 Reduced Drive Register 1 (RDR1AD1)

Address 0x027D Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDR1AD17	RDR1AD16	RDR1AD15	RDR1AD14	RDR1AD13	RDR1AD12	RDR1AD11	RDR1AD10
Reset	0	0	0	0	0	0	0	0

Figure 2-80. Port AD1 Reduced Drive Register 1 (RDR1AD1)

[1] Read: Anytime.
Write: Anytime.

Table 2-78. RDR1AD1 Register Field Descriptions

Field	Description
7-0 RDR1AD1	**Port AD1 reduced drive**—Select reduced drive for Port AD1 outputs This register configures the drive strength of Port AD1 output pins 7 through 0 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.83 Port AD1 Pull Up Enable Register 0 (PER0AD1)

Address 0x027E

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PER0AD17	PER0AD16	PER0AD15	PER0AD14	PER0AD13	PER0AD12	PER0AD11	PER0AD10
Reset	0	0	0	0	0	0	0	0

Figure 2-81. Port AD1 Pull Device Up Register 0 (PER0AD1)

[1] Read: Anytime.
Write: Anytime.

Table 2-79. PER0AD1 Register Field Descriptions

Field	Description
7-0 PER0AD1	**Port AD1 pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset no pull device is enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.84 Port AD1 Pull Up Enable Register 1 (PER1AD1)

Address 0x027F

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PER1AD17	PER1AD16	PER1AD15	PER1AD14	PER1AD13	PER1AD12	PER1AD11	PER1AD10
Reset	0	0	0	0	0	0	0	0

Figure 2-82. Port AD1 Pull Up Enable Register 1 (PER1AD1)

[1] Read: Anytime.
Write: Anytime.

Table 2-80. PER1AD1 Register Field Descriptions

Field	Description
7-0 PER1AD1	**Port AD1 pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset no pull device is enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.85 Port R Data Register (PTR)

Address 0x0368 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PTR7	PTR6	PTR5	PTR4	PTR3	PTR2	PTR1	PTR0
Altern. Function	TIMIOC7	TIMIOC6	TIMIOC5	TIMIOC4	TIMIOC3	TIMIOC2	TIMIOC1	TIMIOC0
Reset	0	0	0	0	0	0	0	0

Figure 2-83. Port R Data Register (PTR)

1. Read: Anytime.
   Write: Anytime.

Table 2-81. PTR Register Field Descriptions

Field	Description
7-0 PTR	**Port R general purpose input/output data**—Data Register Port R pins 7 through 0 are associated with TIM channels TIMIOC7 through TIMIOC0. When not used with the alternative function, these pins can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.86 Port R Input Register (PTIR)

Address 0x0369 Access: User read[1]

	7	6	5	4	3	2	1	0
R	PTIR7	PTIR6	PTIR5	PTIR4	PTIR3	PTIR2	PTIR1	PTIR0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	u	u	u	u	u	u	u	u

▨ = Unimplemented or Reserved u = Unaffected by reset

Figure 2-84. Port R Input Register (PTIR)

1. Read: Anytime.
   Write:Never, writes to this register have no effect.

Table 2-82. PTIR Register Field Descriptions

Field	Description
7-0 PTIR	**Port R input data**— This register always reads back the buffered state of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

2.3.87 Port R Data Direction Register (DDRR)

Address 0x036A Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRR7	DDRR6	DDRR5	DDRR4	DDRR3	DDRR2	DDRR1	DDRR0
Reset	0	0	0	0	0	0	0	0

Figure 2-85. Port R Data Direction Register (DDRR)

1. Read: Anytime.
   Write: Anytime.

Table 2-83. DDRR Register Field Descriptions

Field	Description
7-0 DDRR	**Port R data direction—** This register controls the data direction of pins 7 through 0. The TIM forces the I/O state to be an output for each timer port associated with an enabled output compare. In this case the data direction bits will not change. The data direction bits revert to controlling the I/O direction of a pin when the associated timer output compare is disabled. The timer Input Capture always monitors the state of the pin. 1 Associated pin is configured as output. 0 Associated pin is configured as high-impedance input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTR or PTIR registers, when changing the DDRR register.

2.3.88 Port R Reduced Drive Register (RDRR)

Address 0x036B Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDRR7	RDRR6	RDRR5	RDRR4	RDRR3	RDRR2	RDRR1	RDRR0
Reset	0	0	0	0	0	0	0	0

Figure 2-86. Port R Reduced Drive Register (RDRR)

1. Read: Anytime.
   Write: Anytime.

Table 2-84. RDRR Register Field Descriptions

Field	Description
7-0 RDRR	**Port R reduced drive**—Select reduced drive for outputs This register configures the drive strength of output pins 7 through 0 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.89 Port R Pull Device Enable Register (PERR)

Address 0x036C

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PERR7	PERR6	PERR5	PERR4	PERR3	PERR2	PERR1	PERR0
Reset	0	0	0	0	0	0	0	0

Figure 2-87. Port R Pull Device Enable Register (PERR)

1. Read: Anytime.
   Write: Anytime.

Table 2-85. PERR Register Field Descriptions

Field	Description
7-0 PERR	**Port R pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset no pull device is enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.90 Port R Polarity Select Register (PPSR)

Address 0x036D

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PPSR7	PPSR6	PPSR5	PPSR4	PPSR3	PPSR2	PPSR1	PPSR0
Reset	0	0	0	0	0	0	0	0

Figure 2-88. Port R Polarity Select Register (PPSR)

1. Read: Anytime.
   Write: Anytime.

Table 2-86. PPSR Register Field Descriptions

Field	Description
7-0 PPSR	**Port R pull device select**—Determine pull device polarity on input pins This register selects whether a pull-down or a pull-up device is connected to the pin. 1 A pull-down device is connected to the associated pin, if enabled and if the pin is used as input. 0 A pull-up device is connected to the associated pin, if enabled and if the pin is used as input.

2.3.91 PIM Reserved Register

Address 0x036E

Access: User read[1]

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

▨ = Unimplemented or Reserved

Figure 2-89. PIM Reserved Register

[1] Read: Always reads 0x00
Write: Unimplemented

2.3.92 Port R Routing Register (PTRRR)

Address 0x036F

Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PTRRR7	PTRRR6	PTRRR5	PTRRR4	PTRRR3	PTRRR2	PTRRR1	PTRRR0
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 2-90. Port R Routing Register (PTRRR)

[1] Read: Anytime.
Write: Anytime.

Table 2-87. PTR Routing Register Field Descriptions

Field	Description
7 PTRRR	**Port R routing—** This register configures the re-routing of the associated TIM channel. 1 TIMIOC7 is available on PP7 0 TIMIOC7 is available on PR7
6 PTRRR	**Port R routing—** This register configures the re-routing of the associated TIM channel. 1 TIMIOC6 is available on PP6 0 TIMIOC6 is available on PR6

Table 2-87. PTR Routing Register Field Descriptions (continued)

Field	Description
5 PTRRR	**Port R routing—** This register configures the re-routing of the associated TIM channel. 1 TIMIOC5 is available on PP5 0 TIMIOC5 is available on PR5
4 PTRRR	**Port R routing—** This register configures the re-routing of the associated TIM channel. 1 TIMIOC4 is available on PP4 0 TIMIOC4 is available on PR4
3 PTRRR	**Port R routing—** This register configures the re-routing of the associated TIM channel. 1 TIMIOC3 is available on PP3 0 TIMIOC3 is available on PR3
2 PTRRR	**Port R routing—** This register configures the re-routing of the associated TIM channel. 1 TIMIOC2 is available on PP2 0 TIMIOC2 is available on PR2
1 PTRRR	**Port R routing—** This register configures the re-routing of the associated TIM channel. 1 TIMIOC1 is available on PP1 0 TIMIOC1 is available on PR1
0 PTRRR	**Port R routing—** This register configures the re-routing of the associated TIM channel. 1 TIMIOC0 is available on PP0 0 TIMIOC0 is available on PR0

2.3.93 Port L Data Register (PTL)

Address 0x0370 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PTL7	PTLT6	PTL5	PTL4	PTL3	PTL2	PTL1	PTL0
Altern. Function	(TXD7)	(RXD7)	(TXD6)	(RXD6)	(TXD5)	(RXD5)	(TXD4)	(RXD4)
Reset	0	0	0	0	0	0	0	0

Figure 2-91. Port L Data Register (PTL)

[1] Read: Anytime.
Write: Anytime.

Table 2-88. PTL Register Field Descriptions

Field	Description
7 PTL	**Port L general purpose input/output data**—Data Register Port L pin 7 is associated with the TXD signal of the SCI7 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
6 PTL	**Port L general purpose input/output data**—Data Register Port L pin 6 is associated with the RXD signal of the SCI7 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
5 PTL	**Port L general purpose input/output data**—Data Register Port L pin 5 is associated with the TXD signal of the SCI6 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
4 PTL	**Port L general purpose input/output data**—Data Register Port L pin 4 is associated with the RXD signal of the SCI6 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
3 PTL	**Port L general purpose input/output data**—Data Register Port L pin 3 is associated with the TXD signal of the SC5 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
2 PTL	**Port L general purpose input/output data**—Data Register Port L pin 2 is associated with the RXD signal of the SCI5 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
1 PTL	**Port L general purpose input/output data**—Data Register Port L pin 3 is associated with the TXD signal of the SCI4 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
0 PTL	**Port L general purpose input/output data**—Data Register Port L pin 2 is associated with the RXD signal of the SCI4 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.94 Port L Input Register (PTIL)

Address 0x0371 Access: User read[1]

	7	6	5	4	3	2	1	0
R	PTIL7	PTIL6	PTIL5	PTIL4	PTIL3	PTIL2	PTIL1	PTIL0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	u	u	u	u	u	u	u	u

▨ = Unimplemented or Reserved u = Unaffected by reset

Figure 2-92. Port L Input Register (PTIL)

1. Read: Anytime.
   Write:Never, writes to this register have no effect.

Table 2-89. PTIL Register Field Descriptions

Field	Description
7-0 PTIL	**Port L input data—** This register always reads back the buffered state of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

2.3.95 Port L Data Direction Register (DDRL)

Address 0x0372 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRL7	DDRL6	DDRL5	DDRL4	DDRL3	DDRL2	DDRL1	DDRL0
Reset	0	0	0	0	0	0	0	0

Figure 2-93. Port L Data Direction Register (DDRL)

1. Read: Anytime.
   Write: Anytime.

Table 2-90. DDRL Register Field Descriptions

Field	Description
7-0 DDRL	**Port L data direction—** This register controls the data direction of pins 7 through 0.This register configures each Port L pin as either input or output. If SPI0 is enabled, the SPI0 determines the pin direction. *Refer to SPI section for details*. If the associated SCI transmit or receive channel is enabled this register has no effect on the pins. The pin is forced to be an output if a SCI transmit channel is enabled, it is forced to be an input if the SCI receive channel is enabled. The data direction bits revert to controlling the I/O direction of a pin when the associated channel is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTL or PTIL registers, when changing the DDRL register.

2.3.96 Port L Reduced Drive Register (RDRL)

Address 0x0373 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDRL7	RDRL6	RDRL5	RDRL4	RDRL3	RDRL2	RDRL1	RDRL0
Reset	0	0	0	0	0	0	0	0

Figure 2-94. Port L Reduced Drive Register (RDRL)

[1] Read: Anytime.
Write: Anytime.

Table 2-91. RDRL Register Field Descriptions

Field	Description
7-0 RDRL	**Port L reduced drive**—Select reduced drive for outputs This register configures the drive strength of output pins 7 through 0 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.97 Port L Pull Device Enable Register (PERL)

Address 0x0374 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PERL7	PERL6	PERL5	PERL4	PERL3	PERL2	PERL1	PERL0
Reset	1	1	1	1	1	1	1	1

Figure 2-95. Port L Pull Device Enable Register (PERL)

[1] Read: Anytime.
Write: Anytime.

Table 2-92. PERL Register Field Descriptions

Field	Description
7-0 PERL	**Port L pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset all pull devices are enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.98 Port L Polarity Select Register (PPSL)

Address 0x0375 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PPSL7	PPSL6	PPSL5	PPSL4	PPSL3	PPSL2	PPSL1	PPSL0
Reset	0	0	0	0	0	0	0	0

Figure 2-96. Port L Polarity Select Register (PPSL)

[1] Read: Anytime.
Write: Anytime.

Table 2-93. PPSL Register Field Descriptions

Field	Description
7-0 PPSL	**Port L pull device select**—Determine pull device polarity on input pins This register selects whether a pull-down or a pull-up device is connected to the pin. 1 A pull-down device is connected to the associated pin, if enabled and if the pin is used as input. 0 A pull-up device is connected to the associated pin, if enabled and if the pin is used as input.

2.3.99 Port L Wired-Or Mode Register (WOML)

Address 0x0376 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	WOML7	WOML6	WOML5	WOML4	WOML3	WOML2	WOML1	WOML0
Reset	0	0	0	0	0	0	0	0

Figure 2-97. Port L Wired-Or Mode Register (WOML)

[1] Read: Anytime.
Write: Anytime.

Table 2-94. WOML Register Field Descriptions

Field	Description
7-0 WOML	**Port L wired-or mode**—Enable wired-or functionality This register configures the output pins as wired-or independent of the function used on the pins. If enabled the output is driven active low only (open-drain). A logic level of “1” is not driven.This allows a multipoint connection of several serial modules. These bits have no influence on pins used as inputs. 1 Output buffers operate as open-drain outputs. 0 Output buffers operate as push-pull outputs.

2.3.100 Port L Routing Register (PTLRR)

Address 0x0377 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	PTLRR7	PTLRR6	PTLRR5	PTLRR4	0	0	0	0
W					▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 2-98. Port L Routing Register (PTLRR)

1. Read: Anytime.
   Write: Anytime.

This register configures the re-routing of SCI7, SCI6, SCI5, and SCI4 on alternative ports.

Table 2-95. Port L Routing Summary

Module	PTLRR				Related Pins	
	7	6	5	4		
					TXD	RXD
SCI7	0	x	x	x	PH3	PH2
	1	x	x	x	PL7	PL6
SCI6	x	0	x	x	PH1	PH0
	x	1	x	x	PL5	PL4
SCI5	x	x	0	x	PH7	PH6
	x	x	1	x	PL3	PL2
SCI4	x	x	x	0	PH5	PH4
	x	x	x	1	PL1	PL0

2.3.101 Port F Data Register (PTF)

Address 0x0378 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PTF7	PTFT6	PTF5	PTF4	PTF3	PTF2	PTF1	PTF0
Altern. Function	(TXD3)	(RXD3)	(SCL0)	(SDA0)	($\overline{\text{CS3}}$)	($\overline{\text{CS2}}$)	($\overline{\text{CS1}}$)	($\overline{\text{CS0}}$)
Reset	0	0	0	0	0	0	0	0

Figure 2-99. Port F Data Register (PTF)

1. Read: Anytime.
   Write: Anytime.

Table 2-96. PTF Register Field Descriptions

Field	Description
7 PTF	**Port F general purpose input/output data**—Data Register Port F pin 7 is associated with the TXD signal of the SCI3 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
6 PTF	**Port F general purpose input/output data**—Data Register Port F pin 6 is associated with the RXD signal of the SCI3 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
5 PTF	**Port F general purpose input/output data**—Data Register Port F pin 5 is associated with the TXD signal of the SCI6 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
4 PTF	**Port F general purpose input/output data**—Data Register Port F pin 4 is associated with the RXD signal of the SCI6 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
3 PTF	**Port F general purpose input/output data**—Data Register Port F pin 3 is associated with the TXD signal of the SC5 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
2 PTF	**Port F general purpose input/output data**—Data Register Port F pin 2 is associated with the RXD signal of the SCI5 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
1 PTF	**Port F general purpose input/output data**—Data Register Port F pin 3 is associated with the TXD signal of the SCI4 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
0 PTF	**Port F general purpose input/output data**—Data Register Port F pin 2 is associated with the RXD signal of the SCI4 module. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

2.3.102 Port F Input Register (PTIF)

Address 0x0379 Access: User read[1]

	7	6	5	4	3	2	1	0
R	PTIF7	PTIF6	PTIF5	PTIF4	PTIF3	PTIF2	PTIF1	PTIF0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	u	u	u	u	u	u	u	u

▨ = Unimplemented or Reserved u = Unaffected by reset

Figure 2-100. Port F Input Register (PTIF)

1. Read: Anytime.
   Write:Never, writes to this register have no effect.

Table 2-97. PTIF Register Field Descriptions

Field	Description
7-0 PTIF	**Port F input data**— This register always reads back the buffered state of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

2.3.103 Port F Data Direction Register (DDRF)

Address 0x037A Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRF7	DDRF6	DDRF5	DDRF4	DDRF3	DDRF2	DDRF1	DDRF0
Reset	0	0	0	0	0	0	0	0

Figure 2-101. Port F Data Direction Register (DDRF)

1. Read: Anytime.
   Write: Anytime.

Table 2-98. DDRF Register Field Descriptions

Field	Description
7-0 DDRF	**Port F data direction**— This register controls the data direction of pins 7 through 0.This register configures each Port F pin as either input or output. If SPI0 is enabled, the SPI0 determines the pin direction. *Refer to SPI section for details*. If the associated SCI transmit or receive channel is enabled this register has no effect on the pins. The pin is forced to be an output if a SCI transmit channel is enabled, it is forced to be an input if the SCI receive channel is enabled. The data direction bits revert to controlling the I/O direction of a pin when the associated channel is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

NOTE

Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTF or PTIF registers, when changing the DDRF register.

2.3.104 Port F Reduced Drive Register (RDRF)

Address 0x037B Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDRF7	RDRF6	RDRF5	RDRF4	RDRF3	RDRF2	RDRF1	RDRF0
Reset	0	0	0	0	0	0	0	0

Figure 2-102. Port F Reduced Drive Register (RDRF)

[1] Read: Anytime.
Write: Anytime.

Table 2-99. RDRF Register Field Descriptions

Field	Description
7-0 RDRF	**Port F reduced drive**—Select reduced drive for outputs This register configures the drive strength of output pins 7 through 0 as either full or reduced independent of the function used on the pins. If a pin is used as input this bit has no effect. 1 Reduced drive selected (approx. 1/5 of the full drive strength). 0 Full drive strength enabled.

2.3.105 Port F Pull Device Enable Register (PERF)

Address 0x037C Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PERF7	PERF6	PERF5	PERF4	PERF3	PERF2	PERF1	PERF0
Reset	1	1	1	1	1	1	1	1

Figure 2-103. Port F Pull Device Enable Register (PERF)

[1] Read: Anytime.
Write: Anytime.

Table 2-100. PERF Register Field Descriptions

Field	Description
7-0 PERF	**Port F pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset all pull devices are enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.106 Port F Polarity Select Register (PPSF)

Address 0x037D Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PPSF7	PPSF6	PPSF5	PPSF4	PPSF3	PPSF2	PPSF1	PPSF0
Reset	0	0	0	0	0	0	0	0

Figure 2-104. Port F Polarity Select Register (PPSF)

[1] Read: Anytime.
Write: Anytime.

Table 2-101. PPSF Register Field Descriptions

Field	Description
7-0 PPSF	**Port F pull device select**—Determine pull device polarity on input pins This register selects whether a pull-down or a pull-up device is connected to the pin. 1 A pull-down device is connected to the associated pin, if enabled and if the pin is used as input. 0 A pull-up device is connected to the associated pin, if enabled and if the pin is used as input.

2.3.107 PIM Reserved Register

Address 0x037E Access: User read[1]

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

▨ = Unimplemented or Reserved

Figure 2-105. PIM Reserved Register

[1] Read: Always reads 0x00
Write: Unimplemented

2.3.108 Port F Routing Register (PTFRR)

Address 0x037F Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	0	0	PTFRR5	PTFRR4	PTFRR3	PTFRR2	PTFRR1	PTFRR0
W	▨	▨						
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 2-106. Port F Routing Register (PTFRR)

[1] Read: Anytime.
Write: Anytime.

This register configures the re-routing of SCI3, IIC0, $\overline{\text{CS}}$[3:0] on alternative ports.

Table 2-102. Port F Routing Summary

Module	PTFRR						Related Pins	
	5	4	3	2	1	0		
							TXD	**RXD**
SCI3	0	x	x	x	x	x	PM7	PM6
	1	x	x	x	x	x	PF7	PF6
							SCL	**SDA**
IIC0	x	0	x	x	x	x	PJ7	PJ6
	x	1	x	x	x	x	PF5	PF4
							$\overline{\text{CS}}$	
$\overline{\text{CS3}}$	x	x	0	x	x	x	PJ0	
	x	x	1	x	x	x	PF3	
$\overline{\text{CS2}}$	x	x	x	0	x	x	PJ5	
	x	x	x	1	x	x	PF2	
$\overline{\text{CS1}}$	x	x	x	x	0	x	PJ2	
	x	x	x	x	1	x	PF1	
$\overline{\text{CS0}}$	x	x	x	x	x	0	PJ4	
	x	x	x	x	x	1	PF0	

2.4 Functional Description

2.4.1 General

Each pin except PE0, PE1, and BKGD can act as general purpose I/O. In addition each pin can act as an output from the external bus interface module or a peripheral module or an input to the external bus interface module or a peripheral module.

2.4.2 Registers

A set of configuration registers is common to all ports with exceptions in the expanded bus interface and ATD ports (Table 2-103). All registers can be written at any time, however a specific configuration might not become active.

Example 2-1. Selecting a pull-up device

```
This device does not become active while the port is used as a push-pull output.
```

Table 2-103. Register availability per port[1]

Port	Data	Input	Data Direction	Reduced Drive	Pull Enable	Polarity Select	Wired-Or Mode	Interrupt Enable	Interrupt Flag	Routing
A	yes	-	yes	yes	yes	-	-	-	-	-
B	yes	-	yes			-	-	-	-	-
C	yes	-	yes			-	-	-	-	-
D	yes	-	yes			-	-	-	-	-
E	yes	-	yes			-	-	-	-	-
K	yes	-	yes			-	-	-	-	-
T	yes	yes	yes	yes	yes	yes	-	-	-	-
S	yes	yes	yes	yes	yes	yes	yes	-	-	yes
M	yes	yes	yes	yes	yes	yes	yes	-	-	yes
P	yes	yes	yes	yes	yes	yes	-	yes	yes	-
H	yes	yes	yes	yes	yes	yes	-	yes	yes	-
J	yes	yes	yes	yes	yes	yes	-	yes	yes	-
AD0	yes	-	yes	yes	yes	-	-	-	-	-
AD1	yes	-	yes	yes	yes	-	-	-	-	-
R	yes	yes	yes	yes	yes	yes	-	-	-	-
L	yes	yes	yes	yes	yes	yes	yes	-	-	yes
F	yes	yes	yes	yes	yes	yes	-	-	-	yes

[1] Each cell represents one register with individual configuration bits

2.4.2.1 Data register (PORTx, PTx)

This register holds the value driven out to the pin if the pin is used as a general purpose I/O.

Writing to this register has only an effect on the pin if the pin is used as general purpose output. When reading this address, the buffered state of the pin is returned if the associated data direction register bit is set to “0”.

If the data direction register bits are set to logic level “1”, the contents of the data register is returned. This is independent of any other configuration (Figure 2-107).

2.4.2.2 Input register (PTIx)

This is a read-only register and always returns the buffered state of the pin (Figure 2-107).

2.4.2.3 Data direction register (DDRx)

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

If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 2-107).

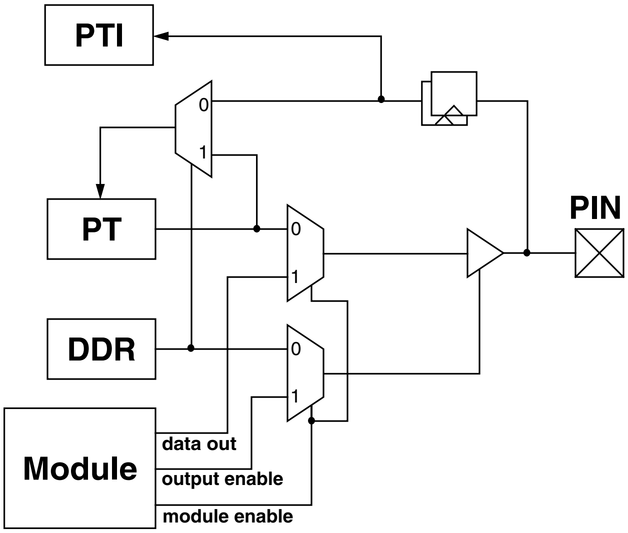


Figure 2-107. Illustration of I/O pin functionality

2.4.2.4 Reduced drive register (RDRx)

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

2.4.2.5 Pull device enable register (PERx)

This register turns on a pull-up or pull-down device.

It becomes active only if the pin is used as an input or as a wired-or output.

2.4.2.6 Polarity select register (PPSx)

This register selects either a pull-up or pull-down device if enabled.

It becomes only active if the pin is used as an input. A pull-up device can be activated if the pin is used as a wired-or output.

2.4.2.7 Wired-or mode register (WOMx)

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

2.4.2.8 Interrupt enable register (PIEx)

If the pin is used as an interrupt input this register serves as a mask to the interrupt flag to enable/disable the interrupt.

2.4.2.9 Interrupt flag register (PIFx)

If the pin is used as an interrupt input this register holds the interrupt flag after a valid pin event.

2.4.2.10 Module routing register (MODRR, PTRRR, PTLRR, PTFRR)

This register supports the re-routing of the CAN0, CAN4, SPI2-0, SCI7-3, IIC0, TIM and CS[3:0] pins to alternative ports. This allows a software re-configuration of the pinouts of the different package options with respect to above peripherals.

2.4.3 Pins and Ports

NOTE

> Please refer to the SOC Guide to determine the pin availability in the different package options.

2.4.3.1 BKGD pin

The BKGD pin is associated with the S12X_BDM and S12X_EBI modules.

During reset, the BKGD pin is used as MODC input.

2.4.3.2 Port A, B

Port A pins PA[7:0] and Port B pins PB[7:0] can be used for either general-purpose I/O with the external bus interface. In this case Port A and Port B are associated with the external address bus outputs ADDR15-ADDR8 and ADDR7-ADDR0, respectively. PB0 is the ADDR0 or $\overline{\mathrm{UDS}}$ output.

2.4.3.3 Port C, D

Port C pins PC[7:0] and Port D pins PD[7:0] can be used for either general-purpose I/O 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).

2.4.3.4 Port E

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

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] can 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[6] can be used for either general-purpose I/O, as $\overline{\text{TAGHI}}$ input or as MODB input during reset.

Port E pin PE[5] can be used for either general-purpose I/O, as $\overline{\text{TAGLO}}$ input, $\overline{\text{RE}}$ output or as MODA input during reset.

Port E pin PE[4] can 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[3] can be used for either general-purpose I/O, as $\overline{\text{LSTRB}}$ or $\overline{\text{LDS}}$ output, or as EROMCTL input during reset.

Port E pin PE[2] can be used for either general-purpose I/O, or as R$\overline{\text{W}}$ or $\overline{\text{WE}}$ output.

Port E pin PE[1] can be used for either general-purpose input or as the level- or falling edge-sensitive $\overline{\text{IRQ}}$ interrupt input. $\overline{\text{IRQ}}$ will be enabled by setting the IRQEN configuration bit (2.3.17/119) and clearing the I-bit in the CPU condition code register. It is inhibited at reset so this pin is initially configured as a simple input with a pull-up.

Port E pin PE[0] can be used for either general-purpose input or as the level-sensitive $\overline{\text{XIRQ}}$ interrupt input. $\overline{\text{XIRQ}}$ can be enabled by clearing the X-bit in the CPU 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).

2.4.3.5 Port K

Port K pins PK[7:0] can be used for either general-purpose I/O, or 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 $\overline{\text{EWAIT}}$ input.

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

2.4.3.6 Port T

This port is associated with the ECT module.

Port T pins PT[7:0] can be used for either general-purpose I/O, or with the channels of the Enhanced Capture Timer.

2.4.3.7 Port S

This port is associated with SCI0, SCI1 and SPI0.

Port S pins PS[7:4] can be used either for general-purpose I/O, or with the SPI0 subsystem.

Port S pins PS[3:2] can be used either for general-purpose I/O, or with the SCI1 subsystem.

Port S pins PS[1:0] can be used either for general-purpose I/O, or with the SCI0 subsystem.

The SPI0 pins can be re-routed.

2.4.3.8 Port M

This port is associated with the SCI3 CAN4-0 and SPI0.

Port M pins PM[7:6] can be used for either general purpose I/O, or with the CAN3 subsystem.

Port M pins PM[5:4] can be used for either general purpose I/O, or with the CAN2 subsystem.

Port M pins PM[3:2] can be used for either general purpose I/O, or with the CAN1 subsystem.

Port M pins PM[1:0] can be used for either general purpose I/O, or with the CAN0 subsystem.

Port M pins PM[5:2] can be used for either general purpose I/O, or with the SPI0 subsystem.

The CAN0, CAN4 and SPI0 pins can be re-routed.

2.4.3.9 Port P

This port is associated with the PWM, SPI1, SPI2 and TIM.

Port P pins PP[7:0] can be used for either general purpose I/O, or with the PWM or with the channels of the standard Timer.subsystem.

Port P pins PP[7:4] can be used for either general purpose I/O, or with the SPI2 subsystem.

Port P pins PP[3:0] can be used for either general purpose I/O, or with the SPI1 subsystem.

2.4.3.10 Port H

This port is associated with the SPI1, SPI2, and SCI7-4.

Port H pins PH[7:4] can be used for either general purpose I/O, or with the SPI2 subsystem.

Port H pins PH[3:0] can be used for either general purpose I/O, or with the SPI1 subsystem.

Port H pins PH[7:6] can be used for either general purpose I/O, or with the SCI5 subsystem.

Port H pins PH[5:4] can be used for either general purpose I/O, or with the SCI4 subsystem.

Port H pins PH[3:2] can be used for either general purpose I/O, or with the SCI7 subsystem.

Port H pins PH[1:0] can be used for either general purpose I/O, or with the SCI6 subsystem.

2.4.3.11 Port J

This port is associated with the chip selects $\overline{\text{CS}}$[3:0] as well as with CAN4, CAN0, IIC1, IIC0, and SCI2.

Port J pins PJ[7:6] can be used for either general purpose I/O, or with the CAN4, IIC0 or CAN0 subsystems.

Port J pins PJ[5:4] can be used for either general purpose I/O, or with the IIC1 subsystem or as chip select outputs.

Port J pin PJ[3] can be used for general purpose I/O.

Port J pin PJ[2] can be used for either general purpose I/O or as chip select output.

Port J pin PJ[1] can be used for either general purpose I/O, or with the SCI2 subsystem.

Port J pin PJ[0] can be used for either general purpose I/O, or with the SCI2 subsystem or as chip select output.

2.4.3.12 Port AD0

This port is associated with the ATD0.

Port AD0 pins PAD[15:0] can be used for either general purpose I/O, or with the ATD0 subsystem.

2.4.3.13 Port AD1

This port is associated with the ATD1.

Port AD1 pins PAD[31:16] can be used for either general purpose I/O, or with the ATD1 subsystem.

2.4.3.14 Port R

This port is associated with the TIM module.

Port R pins PR[7:0] can be used for either general-purpose I/O, or with the channels of the standard Timer.

The TIM channels can be re-routed.

2.4.3.15 Port L

This port is associated with SCI7-4.

Port L pins PL[7:6] can be used for either general purpose I/O, or with SCI7 subsystem.

Port L pins PL[5:4] can be used for either general purpose I/O, or with SCI6 subsystem.

Port L pins PL[3:2] can be used for either general purpose I/O, or with SCI5 subsystem.

Port L pins PL[1:0] can be used for either general purpose I/O, or with SCI4 subsystem.

2.4.3.16 Port F

This port is associated with SCI3, IIC0 and chip selects.

Port L pins PL[7:6] can be used for either general purpose I/O, or with SCI3 subsystem.

Port L pins PL[5:4] can be used for either general purpose I/O, or with IIC0 subsystem.

Port L pins PL[3:0] can be used for either general purpose I/O, or with chip selects.

2.4.4 Pin interrupts

Ports P, H and J offer pin interrupt capability. The interrupt enable as well as the sensitivity to rising or falling edges can be individually configured on per-pin basis. All bits/pins in a port share the same interrupt vector. Interrupts can be used with the pins configured as inputs or outputs.

An interrupt is generated when a bit in the port interrupt flag register and its corresponding port interrupt enable bit are both set. The pin interrupt feature is also capable to wake up the CPU when it is in STOP or WAIT mode.

A digital filter on each pin prevents pulses (Figure 2-109) shorter than a specified time from generating an interrupt. The minimum time varies over process conditions, temperature and voltage (Figure 2-108 and Table 2-104).

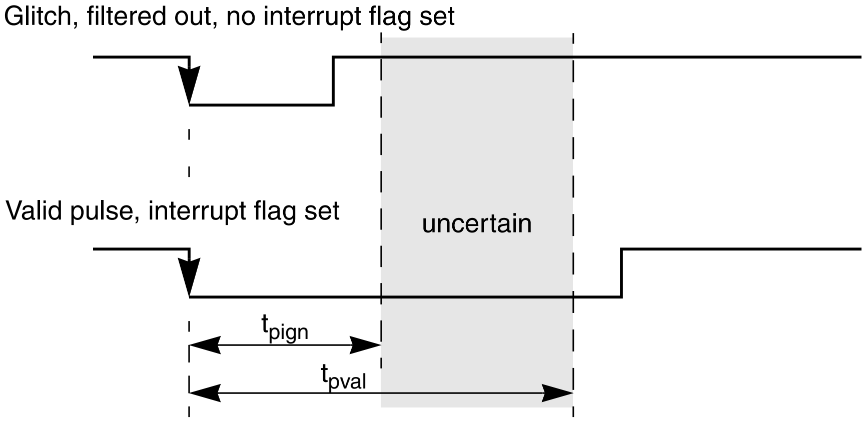

Glitch, filtered out, no interrupt flag set
Valid pulse, interrupt flag set
uncertain
t_{pign}
t_{pval}

Figure 2-108. Interrupt Glitch Filter on Port P, H and J (PPS=0)

Table 2-104. Pulse Detection Criteria

Pulse	**Mode**		
	$\overline{\text{STOP}}$		**STOP**[1]
		Unit	
Ignored	$t_{pulse} \le 3$	bus clocks	$t_{pulse} \le t_{pign}$
Uncertain	$3 < t_{pulse} < 4$	bus clocks	$t_{pign} < t_{pulse} < t_{pval}$
Valid	$t_{pulse} \ge 4$	bus clocks	$t_{pulse} \ge t_{pval}$

[1] These values include the spread of the oscillator frequency over temperature, voltage and process.

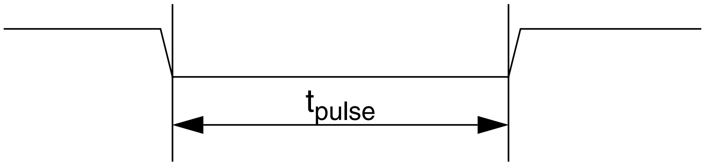


Figure 2-109. Pulse Illustration

A valid edge on an input is detected if 4 consecutive samples of a passive level are followed by 4 consecutive samples of an active level directly or indirectly.

The filters are continuously clocked by the bus clock in RUN and WAIT mode. In STOP mode the clock is generated by an RC-oscillator in the Port Integration Module. To maximize current saving the RC oscillator runs only if the following condition is true on any pin individually:

Sample count <= 4 and interrupt enabled (PIE=1) and interrupt flag not set (PIF=0)

2.5 Initialization Information

2.5.1 Port Data and Data Direction Register writes

It is not recommended to write PORTx/PTx and DDRx in a word access. When changing the register pins from inputs to outputs, the data may have extra transitions during the write access. Initialize the port data register before enabling the outputs.

Chapter 3
Memory Mapping Control (S12XMMCV4)

Table 3-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V04.04	26 Oct 2005		- Reorganization of MEMCTL0 register bits.
V04.05	26 Jul 2006	3.4.2.4/3-212	- Updated XGATE Memory Map
V04.06	15 Nov 2006		- Adding AUTOSAR Compliance concerning illegal CPU accesses

3.1 Introduction

This section describes the functionality of the module mapping control (MMC) sub-block of the S12X platform. The block diagram of the MMC is shown in Figure 3-1.

The MMC module controls the multi-master priority accesses, the selection of internal resources and external space. Internal buses, including internal memories and peripherals, are controlled in this module. The local address space for each master is translated to a global memory space.

3.1.1 Terminology

Table 3-2. Acronyms and Abbreviations

Logic level “1”	Voltage that corresponds to Boolean true state
Logic level “0”	Voltage that corresponds to Boolean false state
0x	Represents hexadecimal number
x	Represents logic level ’don’t care’
byte	8-bit data
word	16-bit data
local address	based on the 64 KBytes Memory Space (16-bit address)
global address	based on the 8 MBytes Memory Space (23-bit address)
Aligned address	Address on even boundary
Mis-aligned address	Address on odd boundary
Bus Clock	System Clock. Refer to CRG Block Guide.
expanded modes	Normal Expanded Mode Emulation Single-Chip Mode Emulation Expanded Mode Special Test Mode
single-chip modes	Normal Single-Chip Mode Special Single-Chip Mode
emulation modes	Emulation Single-Chip Mode Emulation Expanded Mode
normal modes	Normal Single-Chip Mode Normal Expanded Mode
special modes	Special Single-Chip Mode Special Test Mode
NS	Normal Single-Chip Mode
SS	Special Single-Chip Mode
NX	Normal Expanded Mode
ES	Emulation Single-Chip Mode
EX	Emulation Expanded Mode
ST	Special Test Mode
Unimplemented areas	Areas which are accessible by the pages (RPAGE,PPAGE,EPAGE) and not implemented
External Space	Area which is accessible in the global address range 14_0000 to 3F_FFFF
external resource	Resources (Emulator, Application) connected to the MCU via the external bus on expanded modes (Unimplemented areas and External Space)
PRR	Port Replacement Registers
PRU	Port Replacement Unit located on the emulator side
MCU	MicroController Unit
NVM	Non-volatile Memory; Flash EEPROM or ROM

3.1.2 Features

The main features of this block are:

- Paging capability to support a global 8 Mbytes memory address space
- Bus arbitration between the masters CPU, BDM and XGATE

- Simultaneous accesses to different resources[1] (internal, external, and peripherals) (see Figure 3-1)
- Resolution of target bus access collision
- MCU operation mode control
- MCU security control
- Separate memory map schemes for each master CPU, BDM and XGATE
- ROM control bits to enable the on-chip FLASH or ROM selection
- Port replacement registers access control
- Generation of system reset when CPU accesses an unimplemented address (i.e., an address which does not belong to any of the on-chip modules) in single-chip modes

3.1.3 S12X Memory Mapping

The S12X architecture implements a number of memory mapping schemes including

- a CPU 8 MByte global map, defined using a global page (GPAGE) register and dedicated 23-bit address load/store instructions.
- a BDM 8 MByte global map, defined using a global page (BDMGPR) register and dedicated 23-bit address load/store instructions.
- a (CPU or BDM) 64 KByte local map, defined using specific resource page (RPAGE, EPAGE and PPAGE) registers and the default instruction set. The 64 KBytes visible at any instant can be considered as the local map accessed by the 16-bit (CPU or BDM) address.
- The XGATE 64 Kbyte local map.

The MMC module performs translation of the different memory mapping schemes to the specific global (physical) memory implementation.

3.1.4 Modes of Operation

This subsection lists and briefly describes all operating modes supported by the MMC.

3.1.4.1 Power Saving Modes

- Run mode

  MMC is functional during normal run mode.
- Wait mode

  MMC is functional during wait mode.
- Stop mode

  MMC is inactive during stop mode.

3.1.4.2 Functional Modes

- Single chip modes

  In normal and special single chip mode the internal memory is used. External bus is not active.

[1] Resources are also called targets.

- Expanded modes

  Address, data, and control signals are activated in normal expanded and special test modes when accessing the external bus. Access to internal resources will not cause activity on the external bus.

- Emulation modes

  External bus is active to emulate, via an external tool, the normal expanded or the normal single chip mode.}

3.1.5 Block Diagram

Figure 3-1[1] shows a block diagram of the MMC.

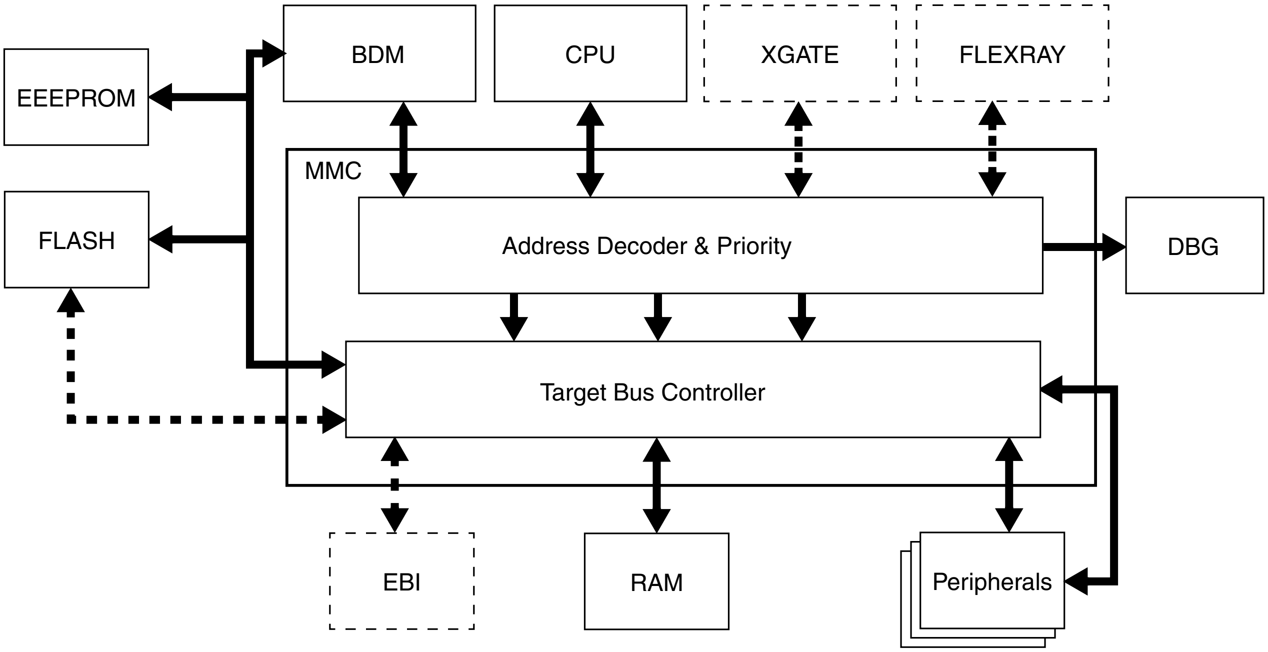


Figure 3-1. MMC Block Diagram

3.2 External Signal Description

The user is advised to refer to the device overview for port configuration and location of external bus signals. Some pins may not be bonded out in all implementations.

Table 3-3 and Table 3-4 outline the pin names and functions. It also provides a brief description of their operation.

[1] Doted blocks and lines are optional. Please refer to the Device User Guide for their availlibilities.

Table 3-3. External Input Signals Associated with the MMC

Signal	I/O	Description	Availability
MODC	I	Mode input	Latched after $\overline{\text{RESET}}$ (active low)
MODB	I	Mode input	Latched after $\overline{\text{RESET}}$ (active low)
MODA	I	Mode input	Latched after $\overline{\text{RESET}}$ (active low)
EROMCTL	I	EROM control input	Latched after $\overline{\text{RESET}}$ (active low)
ROMCTL	I	ROM control input	Latched after $\overline{\text{RESET}}$ (active low)

Table 3-4. External Output Signals Associated with the MMC

Signal	I/O	Description	Available in Modes					
			NS	SS	NX	ES	EX	ST
CS0	O	Chip select line 0	(see Table 3-5)					
CS1	O	Chip select line 1						
CS2	O	Chip select line 2						
CS3	O	Chip select line 3						

3.3 Memory Map and Registers

3.3.1 Module Memory Map

A summary of the registers associated with the MMC block is shown in Figure 3-2. Detailed descriptions of the registers and bits are given in the subsections that follow.

Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000A	MMCCTL0	R	CS3E1	CS3E0	CS2E1	CS2E0	CS1E1	CS1E0	CS0E1	CS0E0
		W								
0x000B	MODE	R	MODC	MODB	MODA	0	0	0	0	0
		W				▨	▨	▨	▨	▨
0x0010	GPAGE	R	0	GP6	GP5	GP4	GP3	GP2	GP1	GP0
		W	▨							
0x0011	DIRECT	R	DP15	DP14	DP13	DP12	DP11	DP10	DP9	DP8
		W								
0x0012	Reserved	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x0013	MMCCTL1	R	TGMRAMON	0	EEEIFRON	PGMIFRON	RAMHM	EROMON	ROMHM	ROMON
		W		▨						
0x0014	Reserved	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x0015	PPAGE	R	PIX7	PIX6	PIX5	PIX4	PIX3	PIX2	PIX1	PIX0
		W								
0x0016	RPAGE	R	RP7	RP6	RP5	RP4	RP3	RP2	RP1	RP0
		W								
0x0017	EPAGE	R	EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
		W								

▨ = Unimplemented or Reserved

Figure 3-2. MMC Register Summary

3.3.2 Register Descriptions

3.3.2.1 MMC Control Register (MMCCTL0)

Address: 0x000A PRR

	7	6	5	4	3	2	1	0
R W	CS3E1	CS3E0	CS2E1	CS2E0	CS1E1	CS1E0	CS0E1	CS0E0
Reset	0	0	0	0	0	0	0	$\overline{\text{ROMON}}$[1]

[1] ROMON is bit[0] of the register MMCTL1 (see Figure 3-10)

▨ = Unimplemented or Reserved

Figure 3-3. MMC Control Register (MMCCTL0)

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

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

Table 3-5. Chip Selects Function Activity

Register Bit	Chip Modes					
	NS	SS	NX	ES	EX	ST
CS0E[1:0], CS1E[1:0], CS2E[1:0], CS3E[1:0]	Disabled[1]	Disabled	Enabled[2]	Disabled	Enabled	Disabled

[1] Disabled: feature always inactive.

[2] Enabled: activity is controlled by the appropriate register bit value.

The MMCCTL0 register is used to control external bus functions, like:

- Availability of chip selects. (See Table 3-5 and Table 3-6)
- Control of different external stretch mechanism. For more detail refer to the S12X_EBI BlockGuide.

CAUTION

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

Table 3-6. MMCCTL0 Field Descriptions

Field	Description
7–6 CS3E[1:0]	**Chip Select 3 Enables** — These bits enable the external chip select $\overline{\text{CS3}}$ output which is asserted during accesses to specific external addresses. The associated global address range is shown in Table 3-7 and Figure 3-17. Chip select 3 is only active if enabled in Normal Expanded mode, Emulation Expanded mode. The function disabled in all other operating modes. 00 Chip select 3 is disabled 01,10,11 Chip select 3 is enabled
5–4 CS2E[1:0]	**Chip Select 2 Enables** — These bits enable the external chip select $\overline{\text{CS2}}$ output which is asserted during accesses to specific external addresses. The associated global address range is shown in Table 3-7 and Figure 3-17. Chip select 2 is only active if enabled in Normal Expanded mode, Emulation Expanded mode. The function disabled in all other operating modes. 00 Chip select 2 is disabled 01,10,11 Chip select 2 is enabled
3–2 CS1E[1:0]	**Chip Select 1 Enables** — These bits enable the external chip select $\overline{\text{CS1}}$ output which is asserted during accesses to specific external addresses. The associated global address range is shown in Table 3-7 and Figure 3-17. Chip select 1 is only active if enabled in Normal Expanded mode, Emulation Expanded mode. The function disabled in all other operating modes. 00 Chip select 1 is disabled 01,10,11 Chip select 1 is enabled
1–0 CS0E[1:0]	**Chip Select 0 Enables** — These bits enable the external chip select $\overline{\text{CS0}}$ output which is asserted during accesses to specific external addresses. The associated global address range is shown in Table 3-7 and Figure 3-17. Chip select 0 is only active if enabled in Normal Expanded mode, Emulation Expanded mode. The function disabled in all other operating modes. 00 Chip select 0 is disabled 01,10,11 Chip select 0 is enabled

Table 3-7 shows the address boundaries of each chip select and the relationship with the implemented resources (internal) parameters.

Table 3-7. Global Chip Selects Memory Space

Chip Selects	Bottom Address	Top Address
$\overline{\text{CS3}}$	0x00_0800	0x0F_FFFF minus RAMSIZE(1)
$\overline{\text{CS2}}$(2)	0x14_0000	0x1F_FFFF
$\overline{\text{CS1}}$	0x20_0000	0x3F_FFFF
$\overline{\text{CS0}}$(3)	0x40_0000	0x7F_FFFF minus FLASHSIZE(4)

1. External RPAGE accesses in (NX, EX)
2. When ROMHM is set (see ROMHM in Table 3-16) the $\overline{\text{CS2}}$ is asserted in the space occupied by this on-chip memory block.
3. When the internal NVM is enabled (see ROMON in Section 3.3.2.5, "MMC Control Register (MMCCTL1)) the $\overline{\text{CS0}}$ is not asserted in the space occupied by this on-chip memory block.
4. External PPAGE accesses in (NX, EX)

3.3.2.2 Mode Register (MODE)

Address: 0x000B PRR

	7	6	5	4	3	2	1	0
R	MODC	MODB	MODA	0	0	0	0	0
W				▨	▨	▨	▨	▨
Reset	MODC[1]	MODB[1]	MODA[1]	0	0	0	0	0

[1] External signal (see Table 3-3).

▨ = Unimplemented or Reserved

Figure 3-4. Mode Register (MODE)

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

Write: Only if a transition is allowed (see Figure 3-5). In emulation modes write operations will be also directed to the external bus.

The MODE bits of the MODE register are used to establish the MCU operating mode.

CAUTION

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

Table 3-8. MODE Field Descriptions

Field	Description
7–5 MODC, MODB, MODA	**Mode Select Bits** — These bits control the current operating mode during $\overline{\text{RESET}}$ high (inactive). The external mode pins MODC, MODB, and MODA determine the operating mode during $\overline{\text{RESET}}$ low (active). The state of the pins is latched into the respective register bits after the $\overline{\text{RESET}}$ signal goes inactive (see Figure 3-4). Write restrictions exist to disallow transitions between certain modes. Figure 3-5 illustrates all allowed mode changes. Attempting non authorized transitions will not change the MODE bits, but it will block further writes to these register bits except in special modes. Both transitions from normal single-chip mode to normal expanded mode and from emulation single-chip to emulation expanded mode are only executed by writing a value of 3’b101 (write once). Writing any other value will not change the MODE bits, but will block further writes to these register bits. Changes of operating modes are not allowed when the device is secured, but it will block further writes to these register bits except in special modes. In emulation modes reading this address returns data from the external bus which has to be driven by the emulator. It is therefore responsibility of the emulator hardware to provide the expected value (i.e. a value corresponding to normal single chip mode while the device is in emulation single-chip mode or a value corresponding to normal expanded mode while the device is in emulation expanded mode).

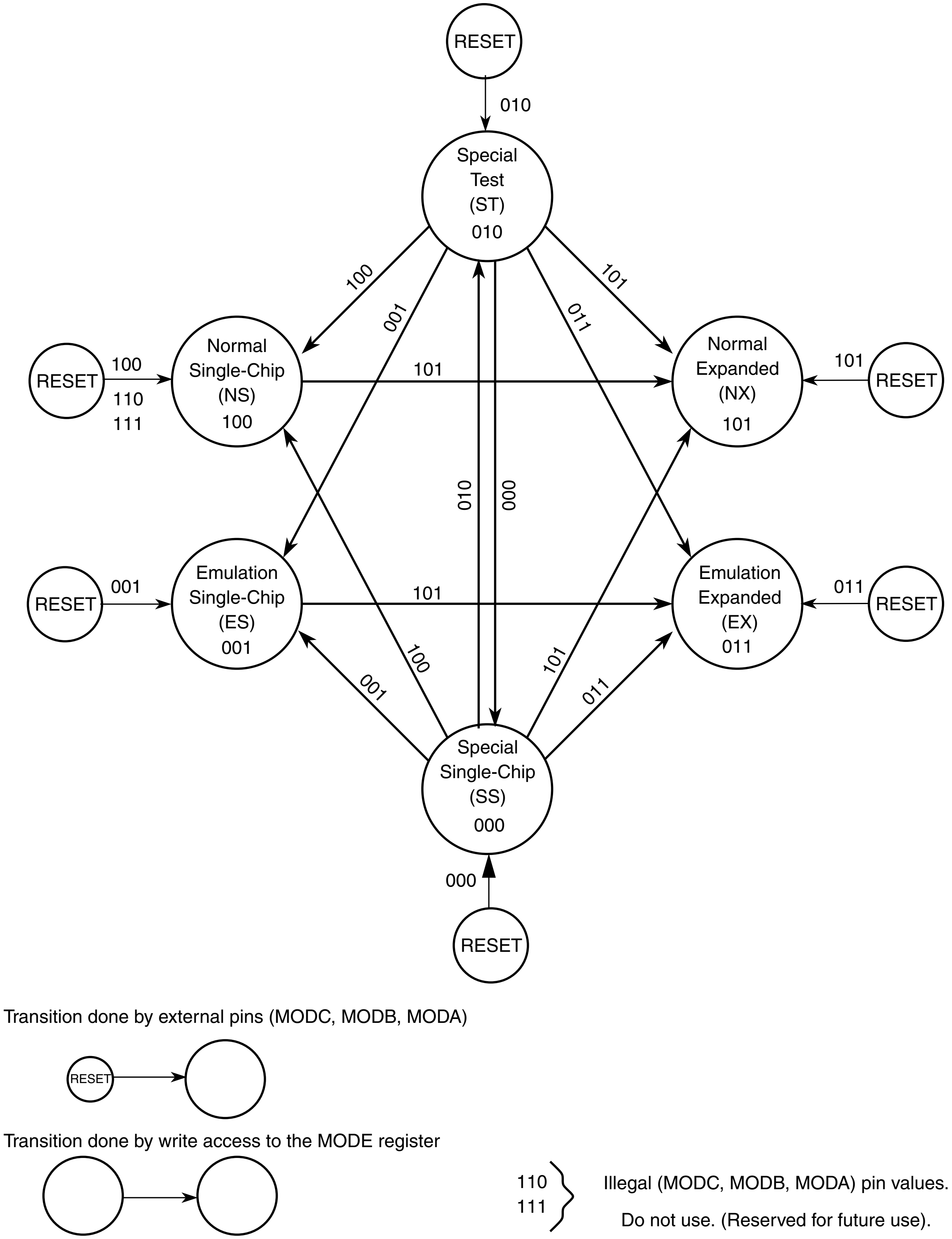


Figure 3-5. Mode Transition Diagram when MCU is Unsecured

3.3.2.3 Global Page Index Register (GPAGE)

Address: 0x0010

	7	6	5	4	3	2	1	0
R	0	GP6	GP5	GP4	GP3	GP2	GP1	GP0
W	▨							
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 3-6. Global Page Index Register (GPAGE)

Read: Anytime

Write: Anytime

The global page index register is used to construct a 23 bit address in the global map format. It is only used when the CPU is executing a global instruction (GLDAA, GLDAB, GLDD, GLDS, GLDX, GLDY,GSTAA, GSTAB, GSTD, GSTS, GSTX, GSTY) (see CPU Block Guide). The generated global address is the result of concatenation of the CPU local address [15:0] with the GPAGE register [22:16] (see Figure 3-7).

CAUTION

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

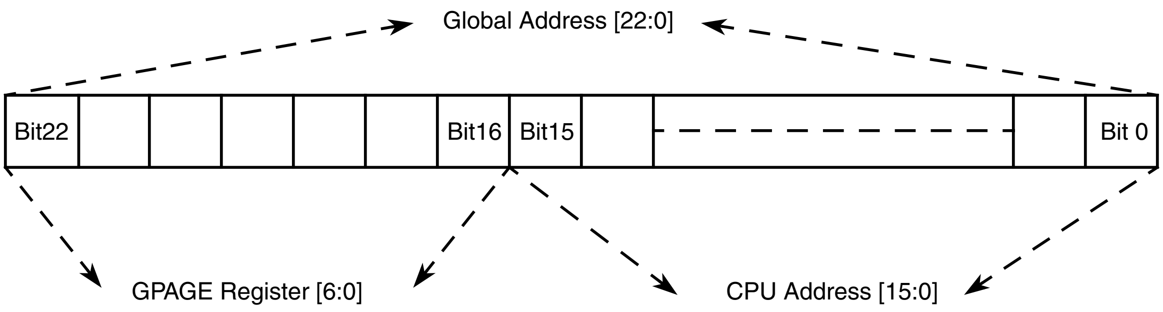


Figure 3-7. GPAGE Address Mapping

Table 3-9. GPAGE Field Descriptions

Field	Description
6–0 GP[6:0]	**Global Page Index Bits 6–0** — These page index bits are used to select which of the 128 64-kilobyte pages is to be accessed.

Example 3-1. This example demonstrates usage of the GPAGE register

```
LDX      #0x5000           ;Set GPAGE offset to the value of 0x5000
MOVB     #0x14, GPAGE      ;Initialize GPAGE register with the value of 0x14
GLDAA    X                 ;Load Accu A from the global address 0x14_5000
```

3.3.2.4 Direct Page Register (DIRECT)

Address: 0x0011

	7	6	5	4	3	2	1	0
R W	DP15	DP14	DP13	DP12	DP11	DP10	DP9	DP8
Reset	0	0	0	0	0	0	0	0

Figure 3-8. Direct Register (DIRECT)

Read: Anytime

Write: anytime in special modes, one time only in other modes.

This register determines the position of the 256 Byte direct page within the memory map.It is valid for both global and local mapping scheme.

Table 3-10. DIRECT Field Descriptions

Field	Description
7–0 DP[15:8]	**Direct Page Index Bits 15–8** — These bits are used by the CPU when performing accesses using the direct addressing mode. The bits from this register form bits [15:8] of the address (see Figure 3-9).

CAUTION

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

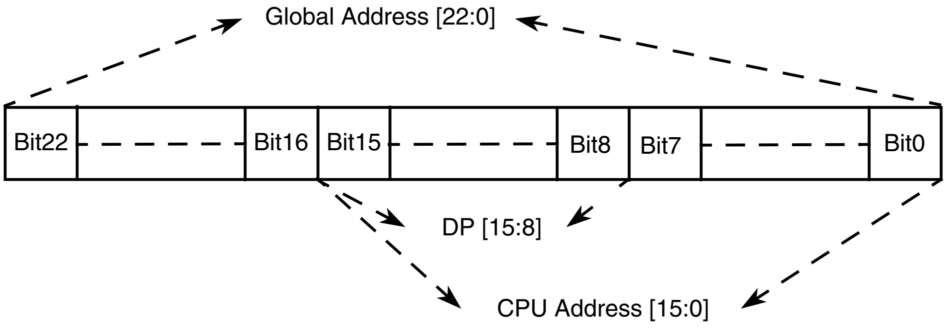


Figure 3-9. DIRECT Address Mapping

Bits [22:16] of the global address will be formed by the GPAGE[6:0] bits in case the CPU executes a global instruction in direct addressing mode or by the appropriate local address to the global address expansion (refer to Section 3.4.2.1.1, “Expansion of the Local Address Map).

Example 3-2. This example demonstrates usage of the Direct Addressing Mode

```
        MOVB     #0x80,DIRECT       ;Set DIRECT register to 0x80. Write once only.
                                    ;Global data accesses to the range 0xXX_80XX can be direct.
                                    ;Logical data accesses to the range 0x80XX are direct.

        LDY      <00                ;Load the Y index register from 0x8000 (direct access).
                                    ;< operator forces direct access on some assemblers but in
```

```
;many cases assemblers are “direct page aware” and can
;automatically select direct mode.
```

3.3.2.5 MMC Control Register (MMCCTL1)

Address: 0x0013   PRR

	7	6	5	4	3	2	1	0
R	TGMRAMON	0	EEEIFRON	PGMIFRON	RAMHM	EROMON	ROMHM	ROMON
W		▨						
Reset	0	0	0	0	0	EROMCTL	0	ROMCTL

▨ = Unimplemented or Reserved

Figure 3-10. MMC Control Register (MMCCTL1)

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: Refer to each bit description. In emulation modes write operations will also be directed to the external bus.

CAUTION

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

Table 3-11. MMCCTL1 Field Descriptions

Field	Description
7 TGMRAMON	**EEE Tag RAM and FTM SCRATCH RAM visible in the memory map** Write: Anytime This bit is used to made the EEE Tag RAM nd FTM SCRATCH RAM visible in the global memory map. 0 Not visible in the memory map. 1 Visible in the memory map.
5 EEEIFRON	**EEE IFR visible in the memory map** Write: Anytime This bit is used to made the IFR sector of EEE DATA FLASH visible in the global memory map. 0 Not visible in the memory map. 1 Visible in the memory map.
4 PGMIFRON	**Program IFR visible in the memory map** Write: Anytime This bit is used to made the IFR sector of the Program Flash visible in the global memory map. 0 Not visible in the memory map. 1 Visible in the memory map.
3 RAMHM	**RAM only in higher Half of the memory map** Write: Once in normal and emulation modes and anytime in special modes 0 Accesses to $4000–$7FFF will be mapped to 14_4000-14_7FFF in the global memory space (external access). 1 Accesses to $4000–$7FFF will be mapped to $0F_C000-$0F_FFFF in the global memory space (RAM area).

Table 3-11. MMCCTL1 Field Descriptions (continued)

Field	Description
2 EROMON	**Enables emulated Flash or ROM memory in the memory map** Write: Never This bit is used in some modes to define the placement of the Emulated Flash or ROM (Refer to Table 3-12) 0 Disables the emulated Flash or ROM in the memory map. 1 Enables the emulated Flash or ROM in the memory map.
1 ROMHM	**FLASH or ROM only in higher Half of Memory Map** Write: Once in normal and emulation modes and anytime in special modes 0 The fixed page of Flash or ROM can be accessed in the lower half of the memory map. Accesses to 0x4000–0x7FFF will be mapped to 0x7F_4000-0x7F_7FFF in the global memory space. 1 Disables access to the Flash or ROM in the lower half of the memory map.These physical locations of the Flash or ROM can still be accessed through the program page window. Accesses to 0x4000–0x7FFF will be mapped to 0x14_4000-0x14_7FFF in the global memory space (external access).
0 ROMON	**Enable FLASH or ROM in the memory map** Write: Once in normal and emulation modes and anytime in special modes. This bit is used in some modes to define the placement of the ROM (Refer to Table 3-12) 0 Disables the Flash or ROM from the memory map. 1 Enables the Flash or ROM in the memory map.

EROMON and ROMON control the visibility of the Flash in the memory map for CPU or BDM (not for XGATE). Both local and global memory maps are affected.

Table 3-12. Data Sources when CPU or BDM is Accessing Flash Area

Chip Modes	ROMON	EROMON	DATA SOURCE[1]	Stretch[2]
Normal Single Chip	X	X	Internal Flash	N
Special Single Chip				
Emulation Single Chip	X	0	Emulation Memory	N
	X	1	Internal Flash	
Normal Expanded	0	X	External Application	Y
	1	X	Internal Flash	N
Emulation Expanded	0	X	External Application	Y
	1	0	Emulation Memory	N
	1	1	Internal Flash	
Special Test	0	X	External Application	N
	1	X	Internal Flash	

[1] Internal Flash means Flash resources inside the MCU are read/written.
Emulation memory means resources inside the emulator are read/written (PRU registers, flash replacement, RAM, EEPROM and register space are always considered internal).
External application means resources residing outside the MCU are read/written.

[2] The external access stretch mechanism is part of the EBI module (refer to EBI Block Guide for details).

3.3.2.6 Program Page Index Register (PPAGE)

Address: 0x0015

	7	6	5	4	3	2	1	0
R W	PIX7	PIX6	PIX5	PIX4	PIX3	PIX2	PIX1	PIX0
Reset	1	1	1	1	1	1	1	0

Figure 3-11. Program Page Index Register (PPAGE)

Read: Anytime

Write: Anytime

These eight index bits are used to page 16 KByte blocks into the Flash page window located in the local (CPU or BDM) memory map from address 0x8000 to address 0xBFFF (see Figure 3-12). This supports accessing up to 4 Mbytes of Flash (in the Global map) within the 64 KByte Local map. The PPAGE register is effectively used to construct paged Flash addresses in the Local map format. The CPU has special access to read and write this register directly during execution of CALL and RTC instructions..

CAUTION

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

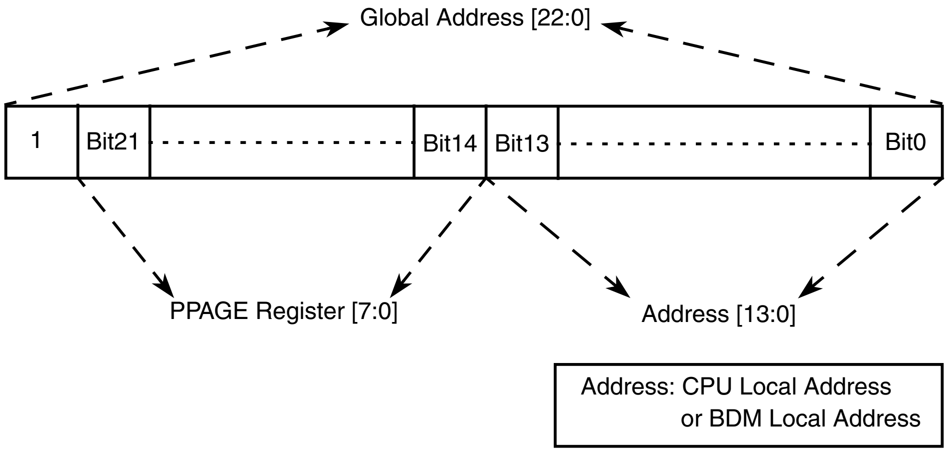


Figure 3-12. PPAGE Address Mapping

NOTE

Writes to this register using the special access of the CALL and RTC instructions will be complete before the end of the instruction execution.

Table 3-13. PPAGE Field Descriptions

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

The fixed 16K page from 0x4000–0x7FFF (when ROMHM = 0) is the page number 0xFD.

The reset value of 0xFE ensures that there is linear Flash space available between addresses 0x4000 and 0xFFFF out of reset.

The fixed 16K page from 0xC000-0xFFFF is the page number 0xFF.

3.3.2.7 RAM Page Index Register (RPAGE)

Address: 0x0016

	7	6	5	4	3	2	1	0
R W	RP7	RP6	RP5	RP4	RP3	RP2	RP1	RP0
Reset	1	1	1	1	1	1	0	1

Figure 3-13. RAM Page Index Register (RPAGE)

Read: Anytime

Write: Anytime

These eight index bits are used to page 4 KByte blocks into the RAM page window located in the local (CPU or BDM) memory map from address 0x1000 to address 0x1FFF (see Figure 3-14). This supports accessing up to 1022 KByte of RAM (in the Global map) within the 64 KByte Local map. The RAM page index register is effectively used to construct paged RAM addresses in the Local map format.

CAUTION

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

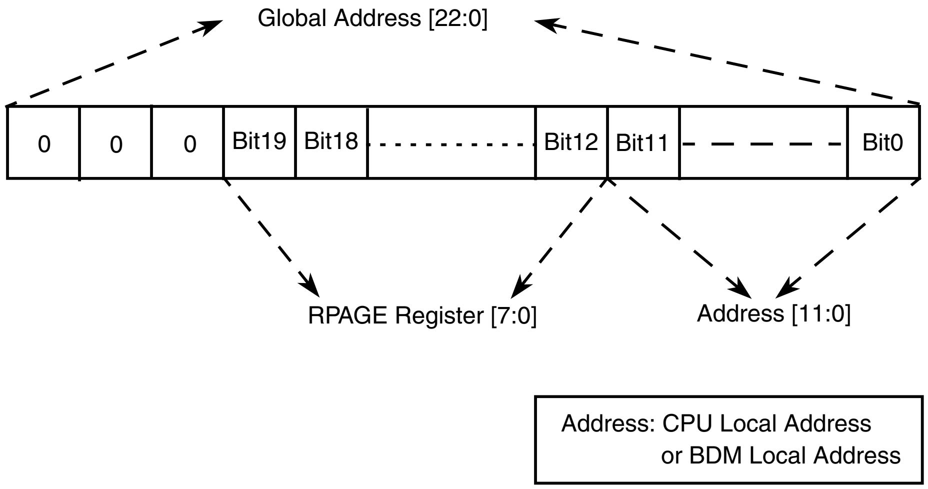


Figure 3-14. RPAGE Address Mapping

NOTE

Because RAM page 0 has the same global address as the register space, it is possible to write to registers through the RAM space when RPAGE = 0x00.

Table 3-14. RPAGE Field Descriptions

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

The reset value of 0xFD ensures that there is a linear RAM space available between addresses 0x1000 and 0x3FFF out of reset.

The fixed 4K page from 0x2000–0x2FFF of RAM is equivalent to page 254 (page number 0xFE).

The fixed 4K page from 0x3000–0x3FFF of RAM is equivalent to page 255 (page number 0xFF).

3.3.2.8 EEPROM Page Index Register (EPAGE)

Address: 0x0017

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

Figure 3-15. EEPROM Page Index Register (EPAGE)

Read: Anytime

Write: Anytime

These eight index bits are used to page 1 KByte blocks into the EEPROM page window located in the local (CPU or BDM) memory map from address 0x0800 to address 0x0BFF (see Figure 3-16). This supports accessing up to 256 KByte of EEPROM (in the Global map) within the 64 KByte Local map. The EEPROM page index register is effectively used to construct paged EEPROM addresses in the Local map format.

CAUTION

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

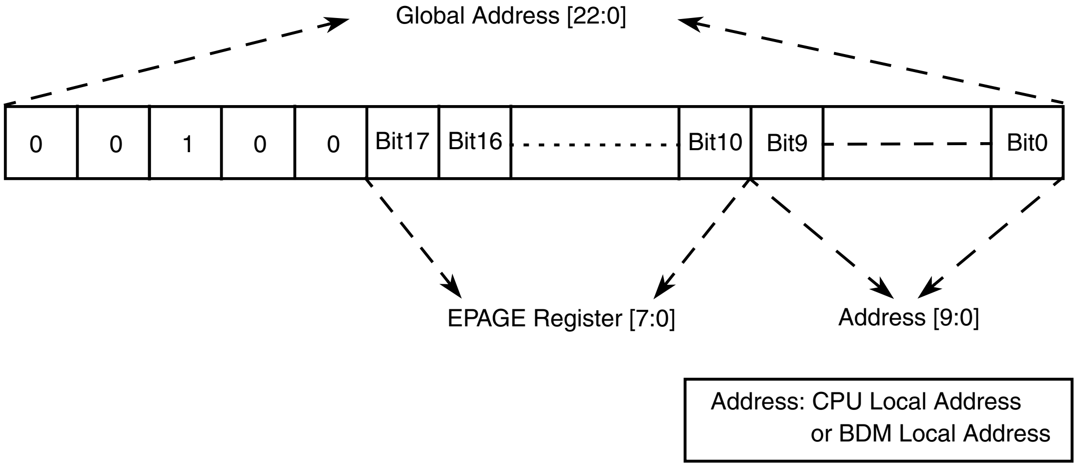


Figure 3-16. EPAGE Address Mapping

Table 3-15. EPAGE Field Descriptions

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

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

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

3.4 Functional Description

The MMC block performs several basic functions of the S12X sub-system operation: MCU operation modes, priority control, address mapping, select signal generation and access limitations for the system. Each aspect is described in the following subsections.

3.4.1 MCU Operating Mode

- Normal single-chip mode

  There is no external bus in this mode. The MCU program is executed from the internal memory and no external accesses are allowed.
- Special single-chip mode

  This mode is generally used for debugging single-chip operation, boot-strapping or security related operations. The active background debug mode is in control of the CPU code execution and the BDM firmware is waiting for serial commands sent through the BKGD pin. There is no external bus in this mode.

- Emulation single-chip mode

  Tool vendors use this mode for emulation systems in which the user's target application is normal single-chip mode. Code is executed from external or internal memory depending on the set-up of the EROMON bit (see Section 3.3.2.5, "MMC Control Register (MMCCTL1)). The external bus is active in both cases to allow observation of internal operations (internal visibility).
- Normal expanded mode

  The external bus interface is configured as an up to 23-bit address bus, 8 or 16-bit data bus with dedicated bus control and status signals. This mode allows 8 or 16-bit external memory and peripheral devices to be interfaced to the system. The fastest external bus rate is half of the internal bus rate. An external signal can be used in this mode to cause the external bus to wait as desired by the external logic.
- Emulation expanded mode

  Tool vendors use this mode for emulation systems in which the user's target application is normal expanded mode.
- Special test mode

  This mode is an expanded mode for factory test.

3.4.2 Memory Map Scheme

3.4.2.1 CPU and BDM Memory Map Scheme

The BDM firmware lookup tables and BDM register memory locations share addresses with other modules; however they are not visible in the memory map during user's code execution. The BDM memory resources are enabled only during the READ_BD and WRITE_BD access cycles to distinguish between accesses to the BDM memory area and accesses to the other modules. (Refer to BDM Block Guide for further details).

When the MCU enters active BDM mode, the BDM firmware lookup tables and the BDM registers become visible in the local memory map in the range 0xFF00-0xFFFF (global address 0x7F_FF00 - 0x7F_FFFF) and the CPU begins execution of firmware commands or the BDM begins execution of hardware commands. The resources which share memory space with the BDM module will not be visible in the memory map during active BDM mode.

Please note that after the MCU enters active BDM mode the BDM firmware lookup tables and the BDM registers will also be visible between addresses 0xBF00 and 0xBFFF if the PPAGE register contains value of 0xFF.

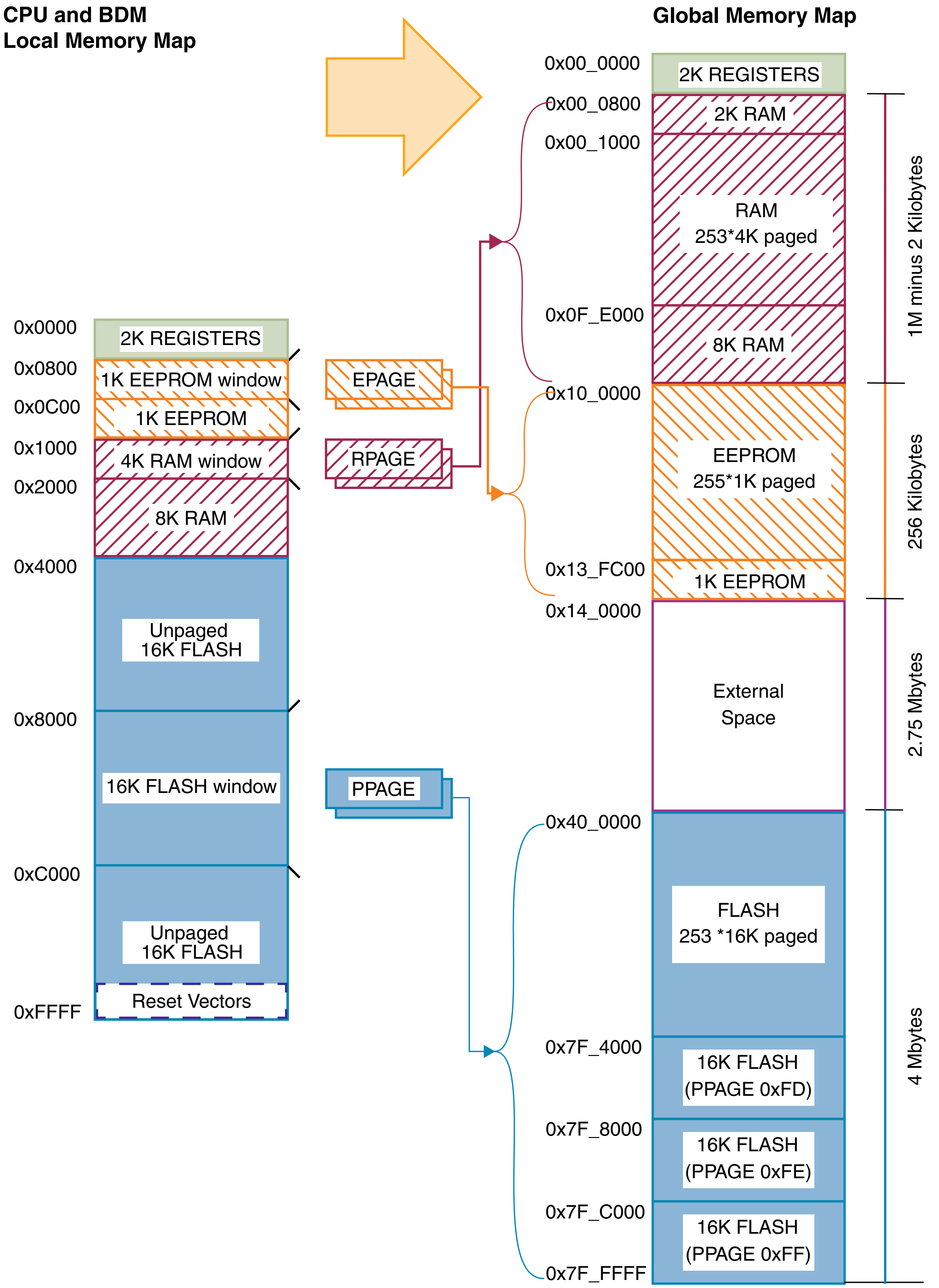


Figure 3-17. Expansion of the Local Address Map

3.4.2.1.1 Expansion of the Local Address Map

Expansion of the CPU Local Address Map

The program page index register in MMC allows accessing up to 4 Mbyte of FLASH or ROM in the global memory map by using the eight page index bits to page 256 16 Kbyte blocks into the program page window located from address 0x8000 to address 0xBFFF in the local CPU memory map.

The page value for the program page window is stored in the PPAGE register. The value of the PPAGE register can be read or written by normal memory accesses as well as by the CALL and RTC instructions (see Section 3.5.1, “CALL and RTC Instructions).

Control registers, vector space and parts of the on-chip memories are located in unpaged portions of the 64-kilobyte local CPU address space.

The starting address of an interrupt service routine must be located in unpaged memory unless the user is certain that the PPAGE register will be set to the appropriate value when the service routine is called. However an interrupt service routine can call other routines that are in paged memory. The upper 16-kilobyte block of the local CPU memory space (0xC000–0xFFFF) is unpaged. It is recommended that all reset and interrupt vectors point to locations in this area or to the other unpaged sections of the local CPU memory map.

Table 3-16 summarizes mapping of the address bus in Flash/External space based on the address, the PPAGE register value and value of the ROMHM bit in the MMCCTL1 register.

Table 3-16. Global FLASH/ROM Allocated

Local CPU Address	ROMHM	External Access	Global Address
0x4000–0x7FFF	0	No	0x7F_4000 –0x7F_7FFF
	1	Yes	0x14_4000–0x14_7FFF
0x8000–0xBFFF	N/A	No[1]	0x40_0000–0x7F_FFFF
	N/A	Yes[1]	
0xC000–0xFFFF	N/A	No	0x7F_C000–0x7F_FFFF

[1] The internal or the external bus is accessed based on the size of the memory resources implemented on-chip. Please refer to Figure 1-23 for further details.

The RAM page index register allows accessing up to 1 Mbyte –2 Kbytes of RAM in the global memory map by using the eight RPAGE index bits to page 4 Kbyte blocks into the RAM page window located in the local CPU memory space from address 0x1000 to address 0x1FFF. The EEPROM page index register EPAGE allows accessing up to 256 Kbytes of EEPROM in the system by using the eight EPAGE index bits to page 1 Kbyte blocks into the EEPROM page window located in the local CPU memory space from address 0x0800 to address 0x0BFF.

Expansion of the BDM Local Address Map

PPAGE, RPAGE, and EPAGE registers are also used for the expansion of the BDM local address to the global address. These registers can be read and written by the BDM.

The BDM expansion scheme is the same as the CPU expansion scheme.

3.4.2.2 Global Addresses Based on the Global Page

CPU Global Addresses Based on the Global Page

The seven global page index bits allow access to the full 8 Mbyte address map that can be accessed with 23 address bits. This provides an alternative way to access all of the various pages of FLASH, RAM and EEE as well as additional external memory.

The GPAGE Register is used only when the CPU is executing a global instruction (see Section 3.3.2.3, “Global Page Index Register (GPAGE)). The generated global address is the result of concatenation of the CPU local address [15:0] with the GPAGE register [22:16] (see Figure 3-7).

BDM Global Addresses Based on the Global Page

The seven BDMGPR Global Page index bits allow access to the full 8 Mbyte address map that can be accessed with 23 address bits. This provides an alternative way to access all of the various pages of FLASH, RAM and EEE as well as additional external memory.

The BDM global page index register (BDMGPR) is used only in the case the CPU is executing a firmware command which uses a global instruction (like GLDD, GSTD) or by a BDM hardware command (like WRITE_W, WRITE_BYTE, READ_W, READ_BYTE). See the BDM Block Guide for further details.

The generated global address is a result of concatenation of the BDM local address with the BDMGPR register [22:16] in the case of a hardware command or concatenation of the CPU local address and the BDMGPR register [22:16] in the case of a firmware command (see Figure 3-18).

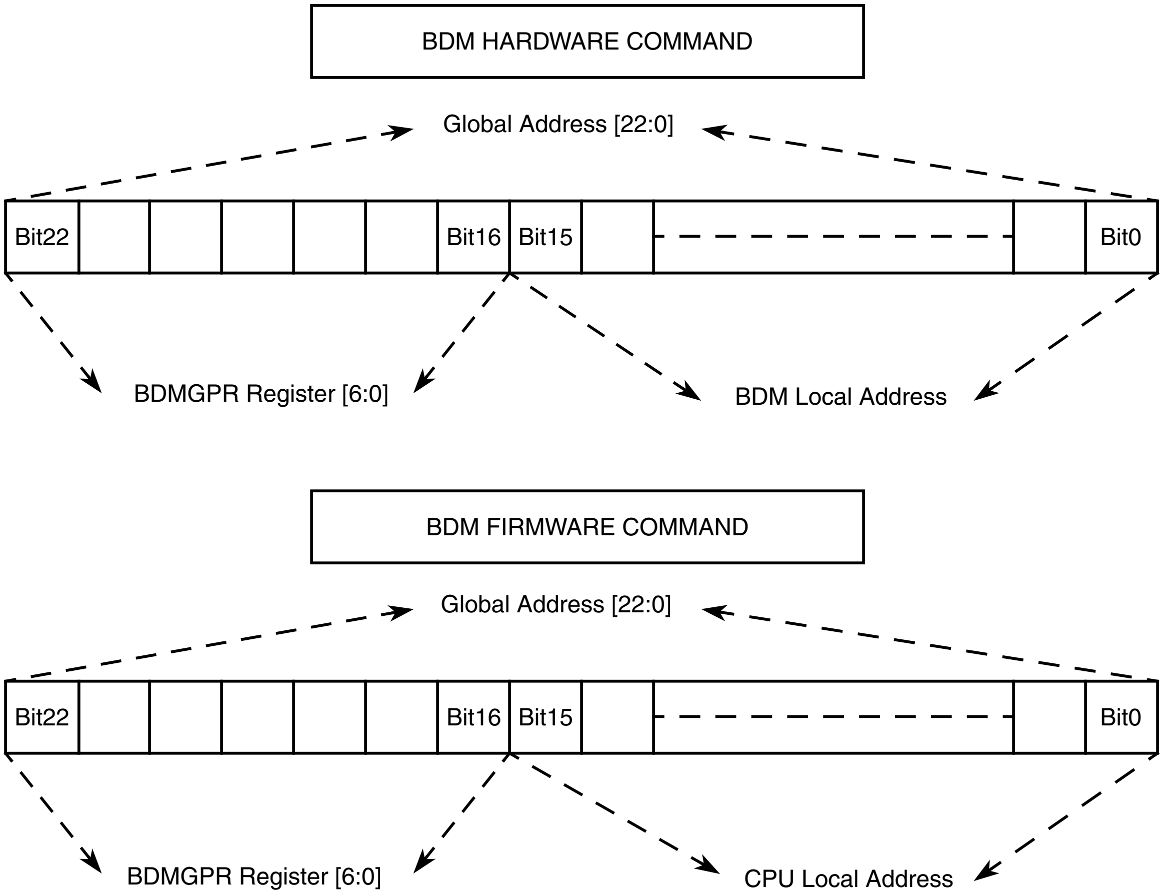


Figure 3-18. BDMGPR Address Mapping

3.4.2.3 Implemented Memory Map

The global memory spaces reserved for the internal resources (RAM, EEE, and FLASH) are not determined by the MMC module. Size of the individual internal resources are however fixed in the design of the device cannot be changed by the user. Please refer to the Device User Guide for further details. Figure 3-19 and Table 3-17 show the memory spaces occupied by the on-chip resources. Please note that the memory spaces have fixed top addresses.

Table 3-17. Global Implemented Memory Space

Internal Resource	$Address
RAM	RAM_LOW = 0x10_0000 minus RAMSIZE[1]
FLASH	FLASH_LOW = 0x80_0000 minus FLASHSIZE[2]

[1] RAMSIZE is the hexadecimal value of RAM SIZE in bytes

[2] FLASHSIZE is the hexadecimal value of FLASH SIZE in bytes

When the device is operating in expanded modes except emulation single-chip mode, accesses to global addresses which are not occupied by the on-chip resources (unimplemented areas or external memory space) result in accesses to the external bus (see Figure 3-19).

In emulation single-chip mode, accesses to global addresses which are not occupied by the on-chip resources (unimplemented areas) result in accesses to the external bus. CPU accesses to global addresses which are occupied by external memory space result in an illegal access reset (system reset) in case of no MPU error. BDM accesses to the external space are performed but the data will be undefined.

In single-chip modes accesses by the CPU (except for firmware commands) to any of the unimplemented areas (see Figure 3-19) will result in an illegal access reset (system reset) in case of no MPU error. BDM accesses to the unimplemented areas are allowed but the data will be undefined.

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

Misaligned word access to the last location of RAM is performed but the data will be undefined.

Misaligned word access to the last location of any global page (64 Kbyte) by any global instruction, 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 3-19), which are only active in normal expanded, emulation expanded (see Section 3.3.2.1, “MMC Control Register (MMCCTL0)).

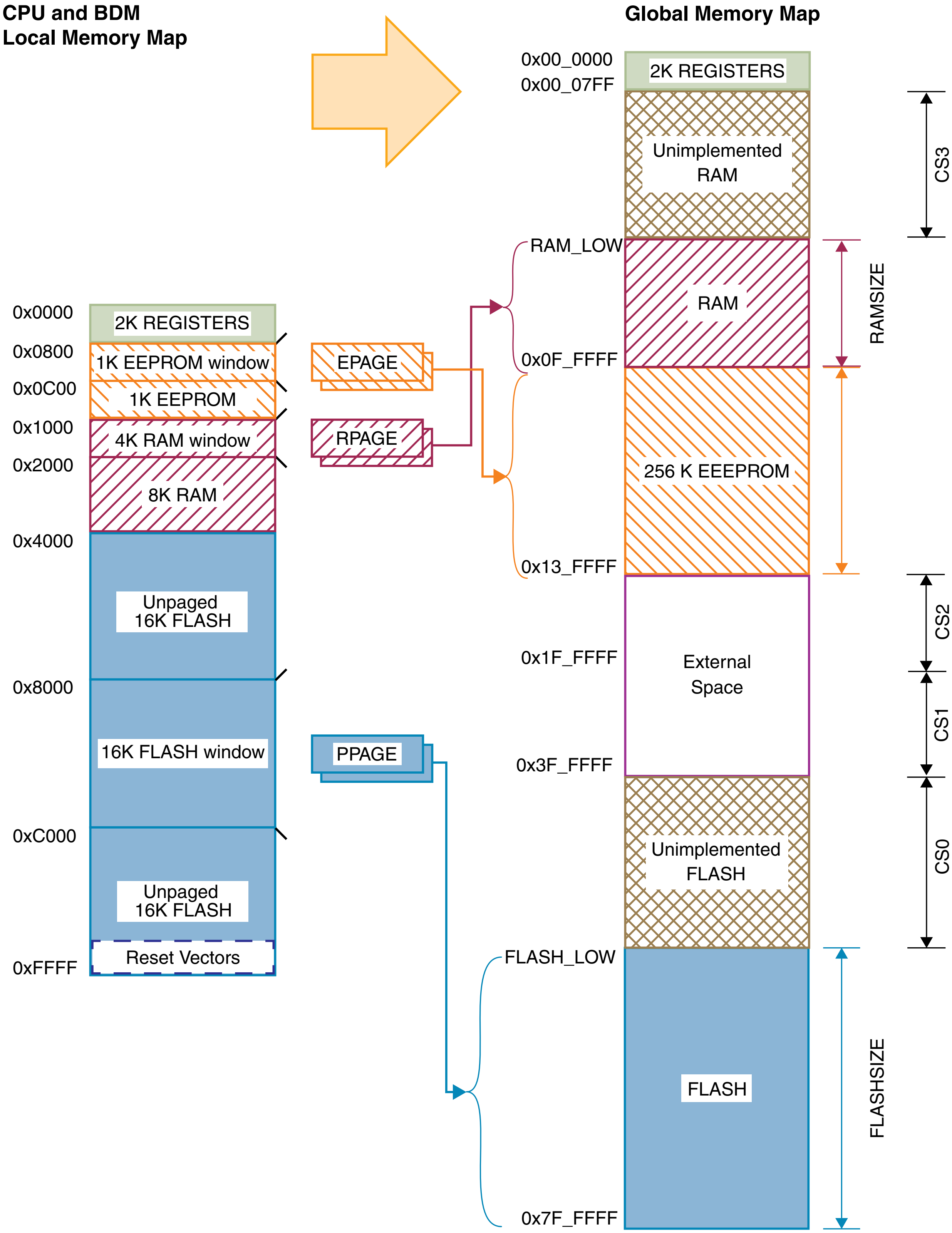


Figure 3-19. S12XE CPU & BDM Global Address Mapping

3.4.2.4 XGATE Memory Map Scheme

3.4.2.4.1 Expansion of the XGATE Local Address Map

The XGATE 64 Kbyte memory space allows access to internal resources only (Registers, RAM, and FLASH). The 2 Kilobyte register address range is the same register address range as for the CPU and the BDM module (see Table 3-18).

XGATE can access the FLASH in single chip modes, even when the MCU is secured. In expanded modes, XGATE can not access the FLASH when MCU is secured.

The local address of the XGATE RAM access is translated to the global RAM address range. The XGATE shares the RAM resource with the CPU and the BDM module (see Table 3-18).

XGATE RAM size (XGRAMSIZE) may be lower or equal to the MCU RAM size (RAMSIZE).In case of XGATE RAM size less than 32 Kbytes (see Figure 3-20), the gap in the xgate local memory map will result in an illegal RAM access (see Section 3.4.3.1, “Illegal XGATE Accesses)

The local address of the XGATE FLASH access is always translated to the global address 0x78_0800 - 0x78_7FFF.

Example 3-3. is a general example of the XGATE memory map implementation.

Table 3-18. XGATE Implemented Memory Space

Internal Resource	$Address
XGATE RAM	XGRAM_LOW = 0x0F_0000 plus (0x1_0000 minus XGRAMSIZE)[1]

[1] XGRAMSIZE is the hexadecimal value of XGATE RAM SIZE in bytes.

Example 3-3.

The MCU FLASHSIZE is 64 Kbytes (0x10000) and MCU RAMSIZE is 32 Kbytes (0x8000).

The XGATE RAMSIZE is 16 Kbytes (0x4000).

The space occupied by the XGATE RAM in the global address space will be:

Bottom address: (0x10_0000 minus 0x4000) = 0x0F_C000

Top address: 0x0F_FFFF

XGATE accesses to local address range 0x0800–0x7FFF will result always in accesses to the following FLASH block in the global address space:

Bottom address: 0x78_0800

Top address: 0x78_7FFF

The gap range in the local memory map 0x8000–0xBFFF will be translated in the global address space:

0x0F_8000 - 0x0F_BFFF (illegal xgate access to system RAM).

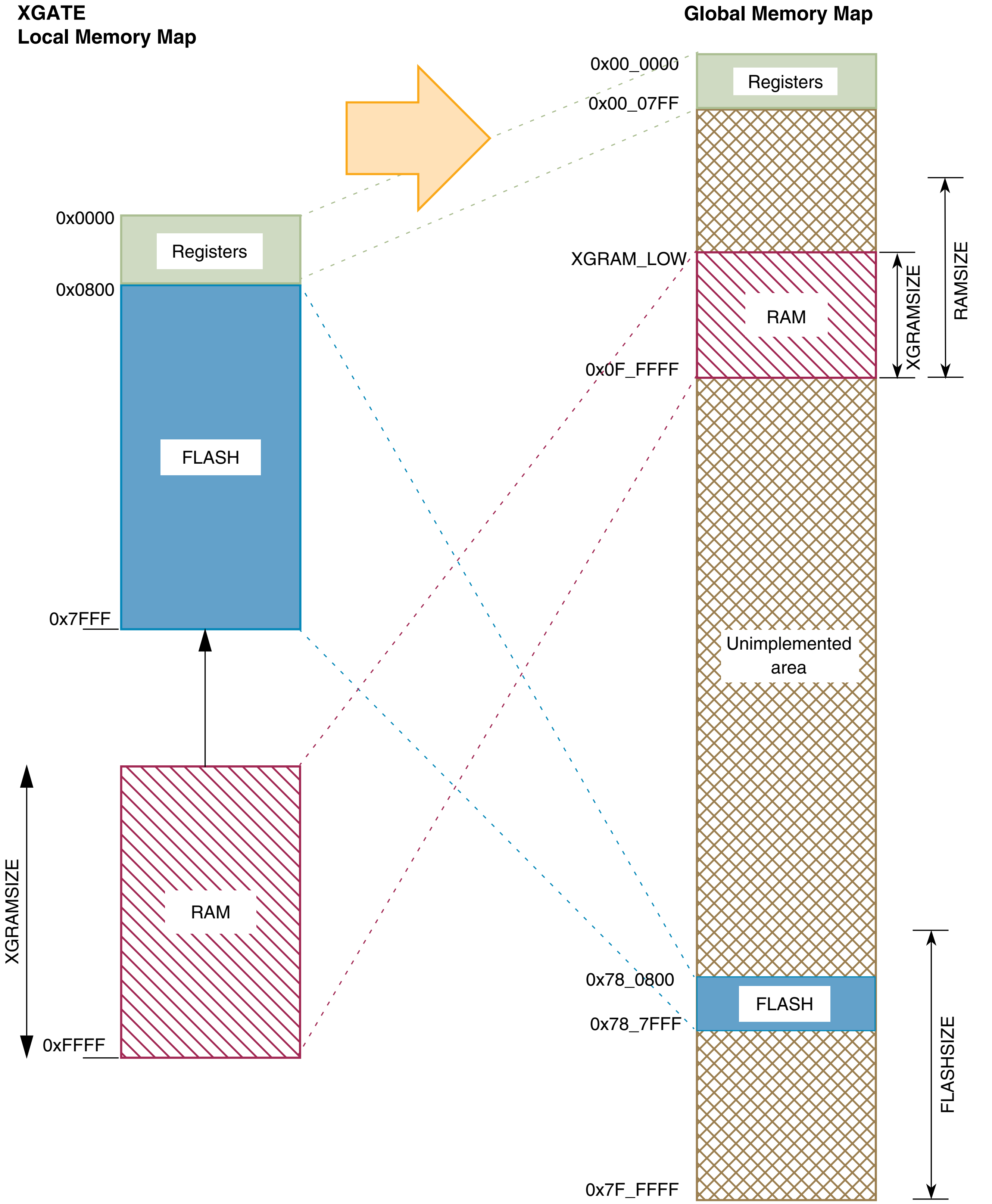


Figure 3-20. XGATE Global Address Mapping

3.4.2.5 Memory Configuration

Two bits in the MMCCTL1 register (ROMHM, RAMHM) configure the mapping of the local address (0x4000-0x7FFF) in the global memory map.

ROMHM, RAMHM are write once in normal and emulation modes and anytime in special modes.

Three areas are identified (See Figure 3-21):

- Program FLASH (0x7F_4000-0x7F_7FFF) when ROMHM = 0.
- External Space (0x14_4000-0x14_7FFF) when ROMHM = 1 and RAMHM = 0.
- XSRAM Space (0x0F_C000-0x0F_FFFF) when ROMHM = 1 and RAMHM = 1.

Table 3-19 shows the translation from the local memory map to the global memory map taking in consideration the different configurations of ROMHM and RAMHM.

Table 3-19. ROMHM and RAMHM Address Location

Local Address	ROMHM	RAMHM	Global Address	Location
0x4000 - 0x7FFF	0	X	0x7F_4000 - 0x7F_7FFF	Internal Flash
	1	0	0x14_4000 - 0x14_7FFF	External Space
	1	1	0x0F_C000 - 0x0F_FFFF	Bottom of the Implemented RAM
0x2000 - 0x3FFF			0x0F_A000 - 0x0F_BFFF	Fixed up to 8K RAM
0x2000 - 0x3FFF	1	0	0x0F_E000 - 0x0F_FFFF	Fixed up to 8K RAM

Table 3-20 describes the application note of the RAM configuration and its dedicated global address.

Table 3-20. RAM Configuration

phase	RPAGE	ROMHM	RAMHM	RAM AREA	Global Address
After reset	RPAGE = 0xFD (Reset value)	0	0	12 Kilobytes	0x0F_D000 - 0x0F_FFFF
During setup	RPAGE = 0xFD (Reset value)	1	1	24 Kilobytes	0x0F_A000 - 0x0F_FFFF
Normal Operation	(0x00 <= RPAGE <= 0xF9)	1	1	28 Kilobytes	0x00_0000 - 0x0F_9FFF
	(0xFA <= RPAGE <= 0xFF)	1	1	24 Kilobytes	0x0F_A000 - 0x0F_FFFF

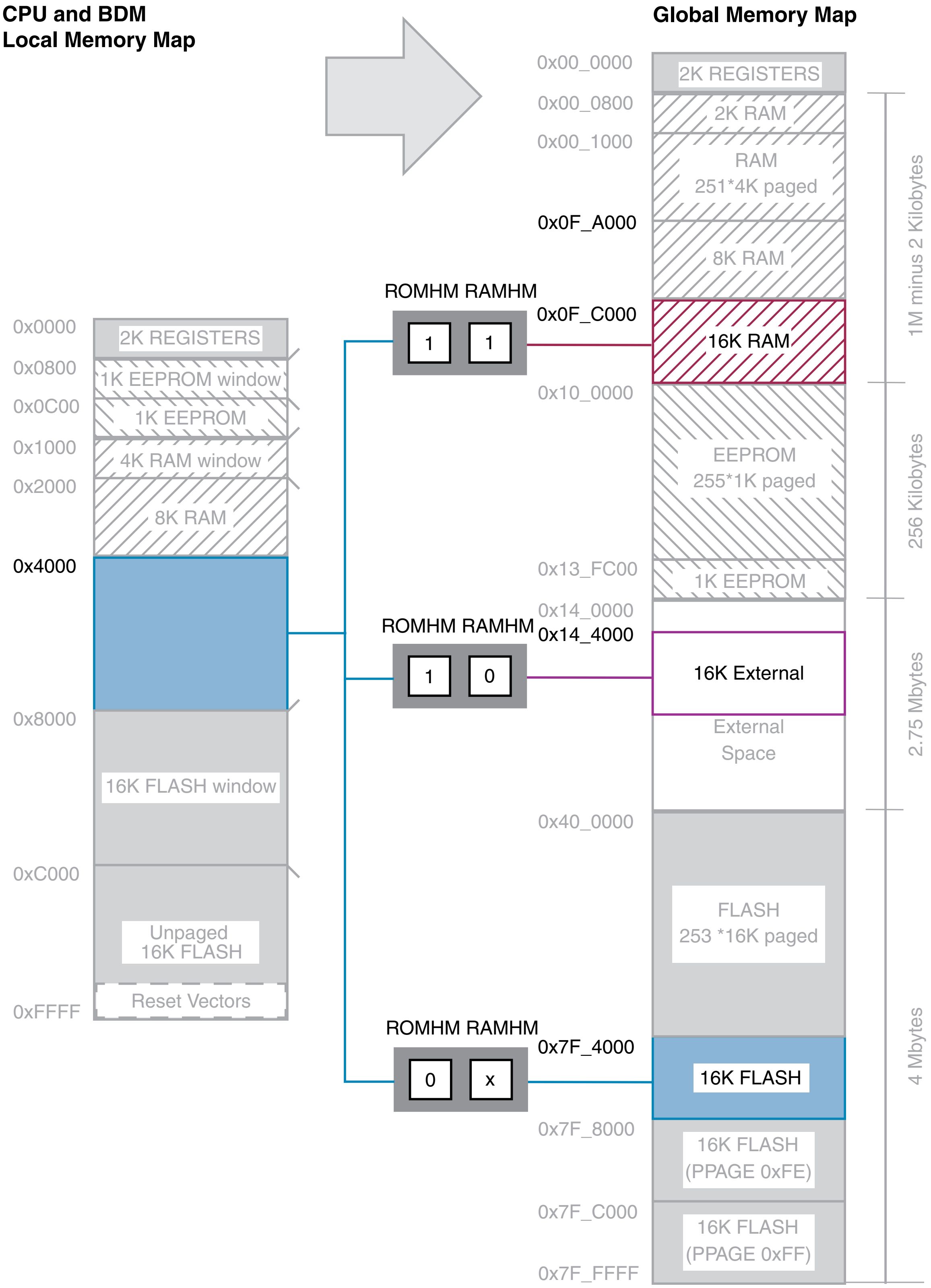


Figure 3-21. ROMHM, RAMHM Memory Configuration

3.4.2.5.1 System XSRAM

System XSRAM has two ways to be accessed by the CPU. One is by the programming of RPAGE and the fixed XSRAM areas configured by the values of ROMHM, RAMHM, or by the usage of the global instruction and the usage of GPAGE.

Figure 3-22 shows the memory map for the implemented XSRAM. The size of the implemented XSRAM is done by the device definition and denoted by RAMSIZE.

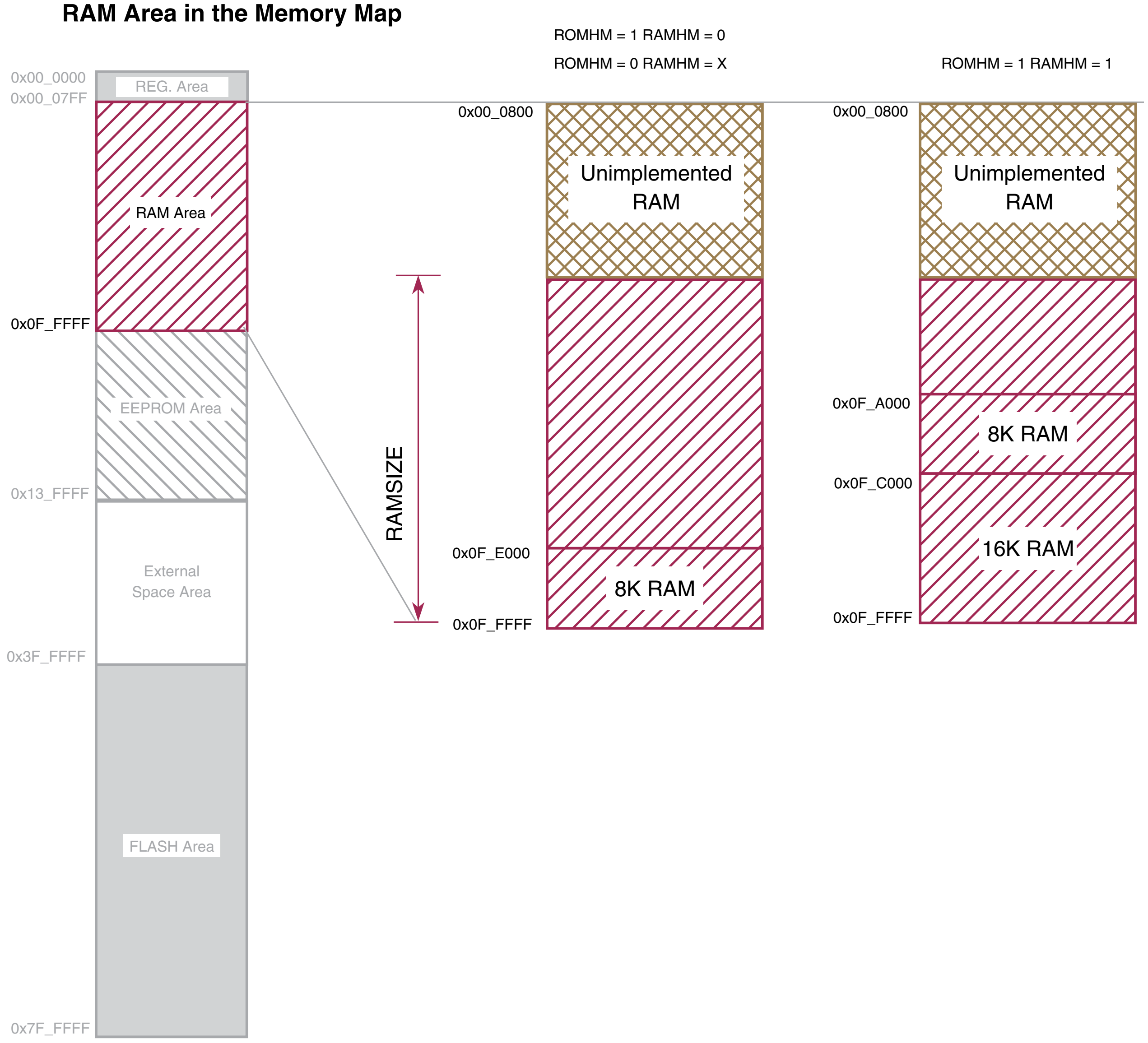


Figure 3-22. S12XE System RAM in the Memory Map

3.4.3 Chip Access Restrictions

CPU and XGATE accesses are watched in the memory protection unit (See MPU Block Guide). In case of access violation, the suspect master is acknowledged with an indication of an error; the victim target will not be accessed.

Other violations MPU is not handling are listed below.

3.4.3.1 Illegal XGATE Accesses

A possible access error is flagged by the MMC and signalled to XGATE under the following conditions:

- XGATE performs misaligned word (in case of load-store or opcode or vector fetch accesses).
- XGATE accesses the register space (in case of opcode or vector fetch).
- XGATE performs a write to Flash in any modes (in case of load-store access).
- XGATE performs an access to a secured Flash in expanded modes (in case of load-store or opcode or vector fetch accesses).

For further details refer to the XGATE Block Guide.

3.4.4 Chip Bus Control

The MMC controls the address buses and the data buses that interface the S12X masters (CPU, BDM and XGATE) with the rest of the system (master buses). In addition the MMC handles all CPU read data bus swapping operations. All internal and external resources are connected to specific target buses (see Figure 3-23[1]).

[1] Doted blocks and lines are optional. Please refer to the Device User Guide for their availlibilities.

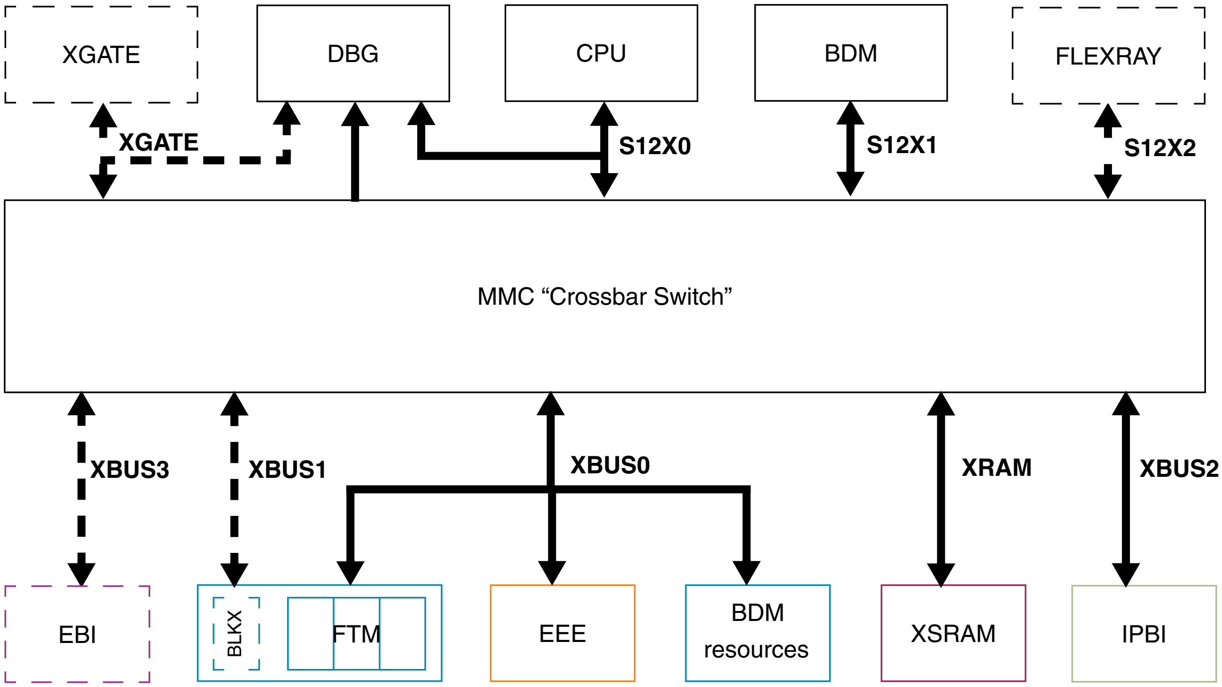


Figure 3-23. MMC Block Diagram

3.4.4.1 Master Bus Prioritization regarding access conflicts on Target Buses

The arbitration scheme allows only one master to be connected to a target at any given time. The following rules apply when prioritizing accesses from different masters to the same target bus:

- CPU always has priority over BDM and XGATE.
- XGATE access to PRU registers constitutes a special case. It is always granted and stalls the CPU for its duration.
- XGATE has priority over BDM.
- BDM has priority over CPU and XGATE when its access is stalled for more than 128 cycles. In the later case the suspect master will be stalled after finishing the current operation and the BDM will gain access to the bus.
- In emulation modes all internal accesses are visible on the external bus as well and the external bus is used during access to the PRU registers.

3.5 Initialization/Application Information

3.5.1 CALL and RTC Instructions

CALL and RTC instructions are uninterruptable CPU instructions that automate page switching in the program page window. The CALL instruction is similar to the JSR instruction, but the subroutine that is called can be located anywhere in the local address space or in any Flash or ROM page visible through the program page window. The CALL instruction calculates and stacks a return address, stacks the current PPAGE value and writes a new instruction-supplied value to the PPAGE register. The PPAGE value controls which of the 256 possible pages is visible through the 16 Kbyte program page window in the 64 Kbyte local CPU memory map. Execution then begins at the address of the called subroutine.

During the execution of the CALL instruction, the CPU performs the following steps:

1. Writes the current PPAGE value into an internal temporary register and writes the new instruction-supplied PPAGE value into the PPAGE register
2. Calculates the address of the next instruction after the CALL instruction (the return address) and pushes this 16-bit value onto the stack
3. Pushes the temporarily stored PPAGE value onto the stack
4. Calculates the effective address of the subroutine, refills the queue and begins execution at the new address

This sequence is uninterruptable. There is no need to inhibit interrupts during the CALL instruction execution. A CALL instruction can be performed from any address to any other address in the local CPU memory space.

The PPAGE value supplied by the instruction is part of the effective address of the CPU. For all addressing mode variations (except indexed-indirect modes) the new page value is provided by an immediate operand in the instruction. In indexed-indirect variations of the CALL instruction a pointer specifies memory locations where the new page value and the address of the called subroutine are stored. Using indirect addressing for both the new page value and the address within the page allows usage of values calculated at run time rather than immediate values that must be known at the time of assembly.

The RTC instruction terminates subroutines invoked by a CALL instruction. The RTC instruction unstacks the PPAGE value and the return address and refills the queue. Execution resumes with the next instruction after the CALL instruction.

During the execution of an RTC instruction the CPU performs the following steps:

1. Pulls the previously stored PPAGE value from the stack
2. Pulls the 16-bit return address from the stack and loads it into the PC
3. Writes the PPAGE value into the PPAGE register
4. Refills the queue and resumes execution at the return address

This sequence is uninterruptable. The RTC can be executed from anywhere in the local CPU memory space.

The CALL and RTC instructions behave like JSR and RTS instruction, they however require more execution cycles. Usage of JSR/RTS instructions is therefore recommended when possible and CALL/RTC instructions should only be used when needed. The JSR and RTS instructions can be used to access subroutines that are already present in the local CPU memory map (i.e. in the same page in the program memory page window for example). However calling a function located in a different page requires usage of the CALL instruction. The function must be terminated by the RTC instruction. Because the RTC instruction restores contents of the PPAGE register from the stack, functions terminated with the RTC instruction must be called using the CALL instruction even when the correct page is already present in the memory map. This is to make sure that the correct PPAGE value will be present on stack at the time of the RTC instruction execution.

3.5.2 Port Replacement Registers (PRRs)

Registers used for emulation purposes must be rebuilt by the in-circuit emulator hardware to achieve full emulation of single chip mode operation. These registers are called port replacement registers (PRRs) (see Table 1-25). PRRs are accessible from CPU, BDM and XGATE using different access types (word aligned, word-misaligned and byte).

Each access to PRRs will be extended to 2 bus cycles for write or read accesses independent of the operating mode. In emulation modes all write operations result in simultaneous writing to the internal registers (peripheral access) and to the emulated registers (external access) located in the PRU in the emulator. All read operations are performed from external registers (external access) in emulation modes. In all other modes the read operations are performed from the internal registers (peripheral access).

Due to internal visibility of CPU accesses the CPU will be halted during XGATE or BDM access to any PRR. This rule applies also in normal modes to ensure that operation of the device is the same as in emulation modes.

A summary of PRR accesses:

- An aligned word access to a PRR will take 2 bus cycles.
- A misaligned word access to a PRRs will take 4 cycles. If one of the two bytes accessed by the misaligned word access is not a PRR, the access will take only 3 cycles.
- A byte access to a PRR will take 2 cycles.

Table 3-21. PRR Listing

PRR Name	PRR Local Address	PRR Location
PORTA	0x0000	PIM
PORTB	0x0001	PIM
DDRA	0x0002	PIM
DDRB	0x0003	PIM
PORTC	0x0004	PIM
PORTD	0x0005	PIM
DDRC	0x0006	PIM
DDRD	0x0007	PIM
PORTE	0x0008	PIM
DDRE	0x0009	PIM
MMCCTL0	0x000A	MMC
MODE	0x000B	MMC
PUCR	0x000C	PIM
RDRIV	0x000D	PIM
EBICTL0	0x000E	EBI
EBICTL1	0x000F	EBI
Reserved	0x0012	MMC
MMCCTL1	0x0013	MMC
ECLKCTL	0x001C	PIM
Reserved	0x001D	PIM
PORTK	0x0032	PIM
DDRK	0x0033	PIM

3.5.3 On-Chip ROM Control

The MCU offers two modes to support emulation. In the first mode (called generator) the emulator provides the data instead of the internal FLASH and traces the CPU actions. In the other mode (called observer) the internal FLASH provides the data and all internal actions are made visible to the emulator.

3.5.3.1 ROM Control in Single-Chip Modes

In single-chip modes the MCU has no external bus. All memory accesses and program fetches are internal (see Figure 3-24).

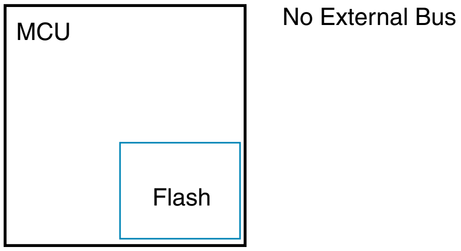


Figure 3-24. ROM in Single Chip Modes

3.5.3.2 ROM Control in Emulation Single-Chip Mode

In emulation single-chip mode the external bus is connected to the emulator. If the EROMON bit is set, the internal FLASH provides the data and the emulator can observe all internal CPU actions on the external bus. If the EROMON bit is cleared, the emulator provides the data (generator) and traces the all CPU actions (see Figure 3-25).

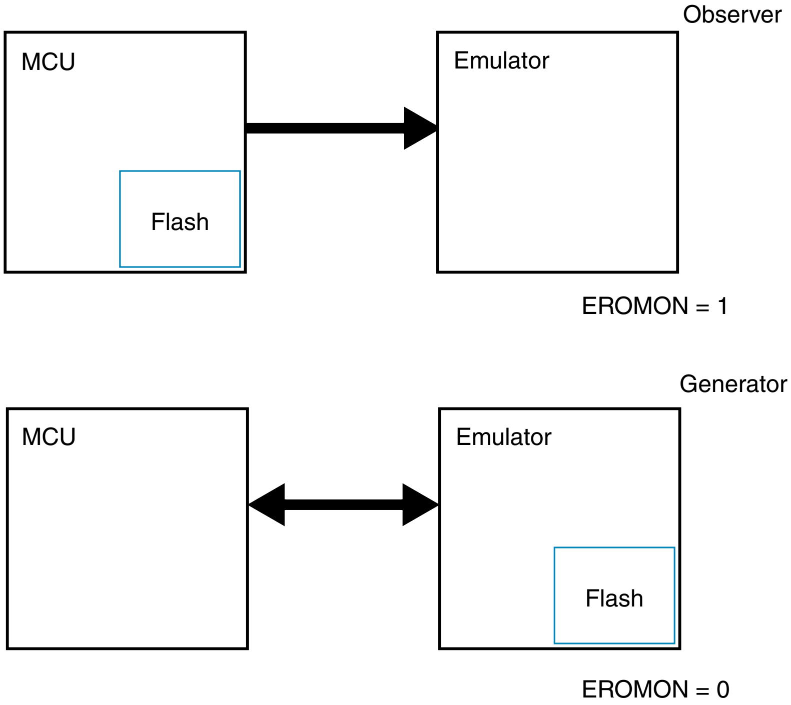


Figure 3-25. ROM in Emulation Single-Chip Mode

3.5.3.3 ROM Control in Normal Expanded Mode

In normal expanded mode the external bus will be connected to the application. If the ROMON bit is set, the internal FLASH provides the data. If the ROMON bit is cleared, the application memory provides the data (see Figure 3-26).

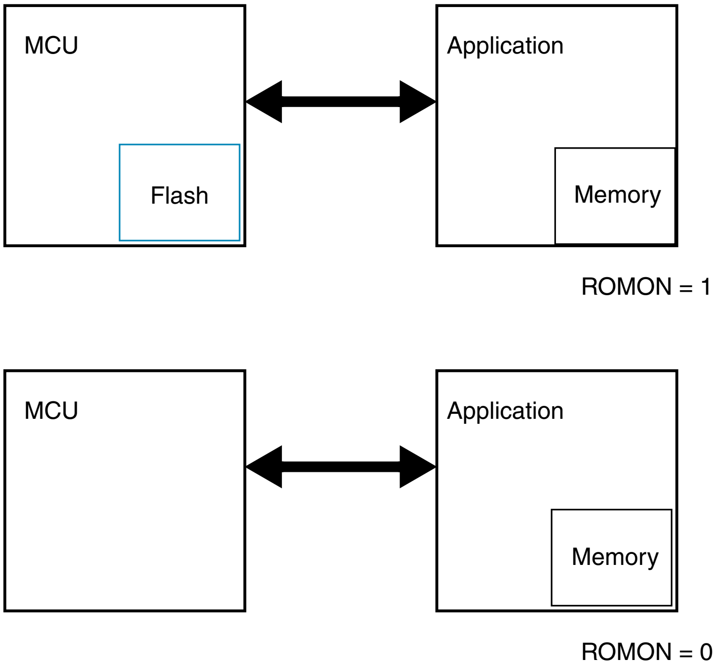


Figure 3-26. ROM in Normal Expanded Mode

3.5.3.4 ROM Control in Emulation Expanded Mode

In emulation expanded mode the external bus will be connected to the emulator and to the application. If the ROMON bit is set, the internal FLASH provides the data. If the EROMON bit is set as well the emulator observes all CPU internal actions, otherwise the emulator provides the data and traces all CPU actions (see Figure 3-27). When the ROMON bit is cleared, the application memory provides the data and the emulator will observe the CPU internal actions (see Figure 3-28).

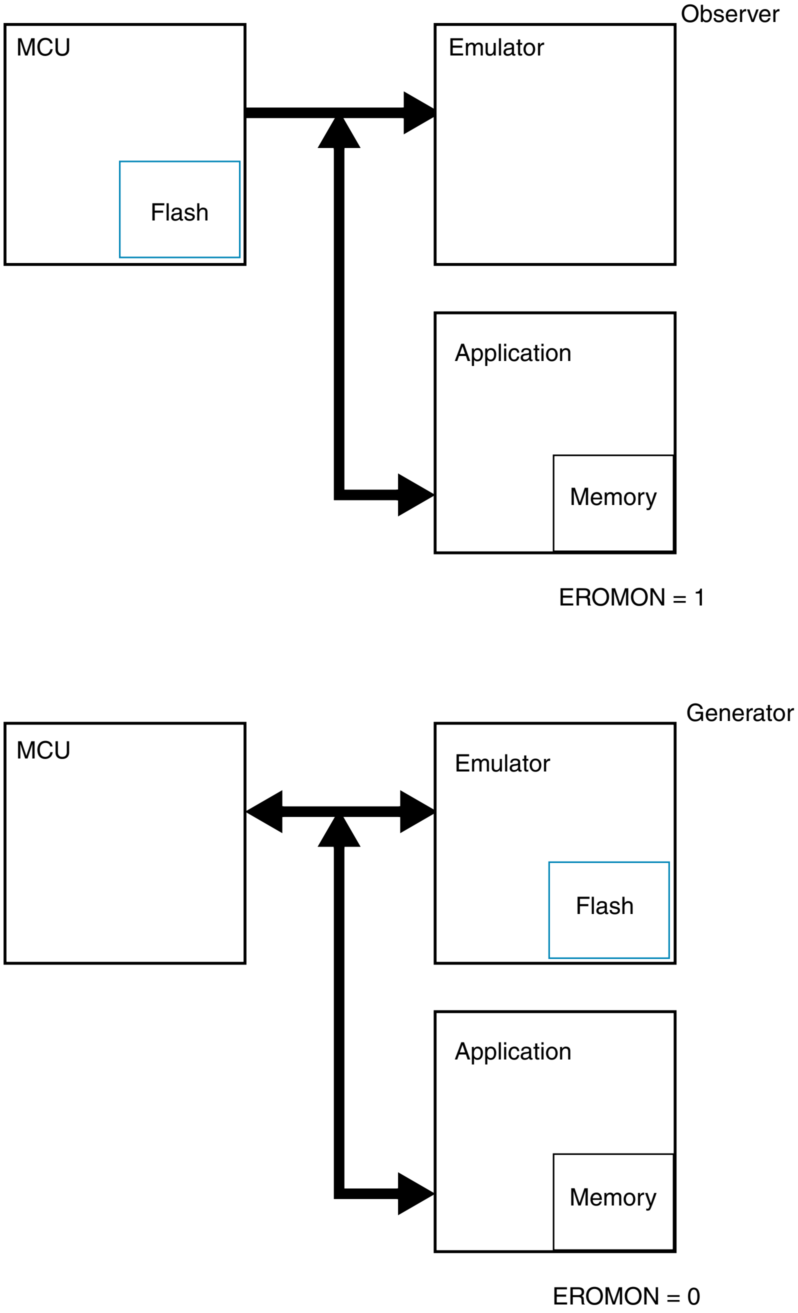


Figure 3-27. ROMON = 1 in Emulation Expanded Mode

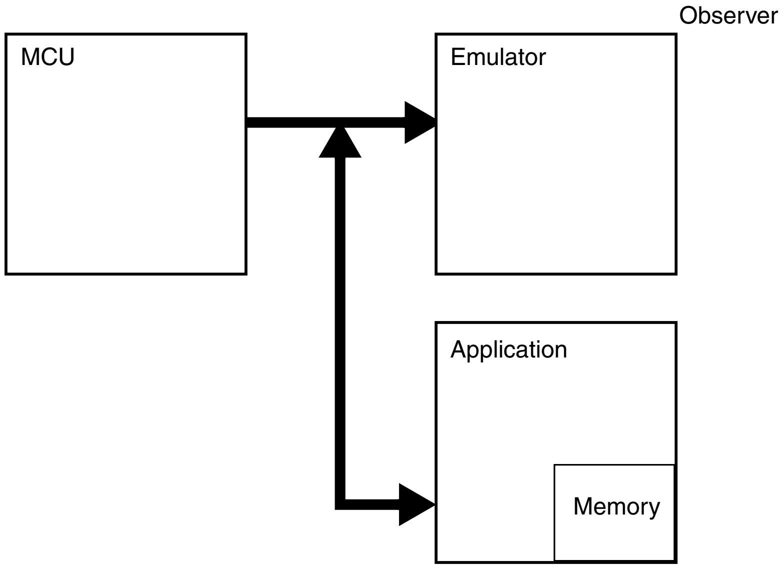


Figure 3-28. ROMON = 0 in Emulation Expanded Mode

3.5.3.5 ROM Control in Special Test Mode

In special test mode the external bus is connected to the application. If the ROMON bit is set, the internal FLASH provides the data, otherwise the application memory provides the data (see Figure 3-29).

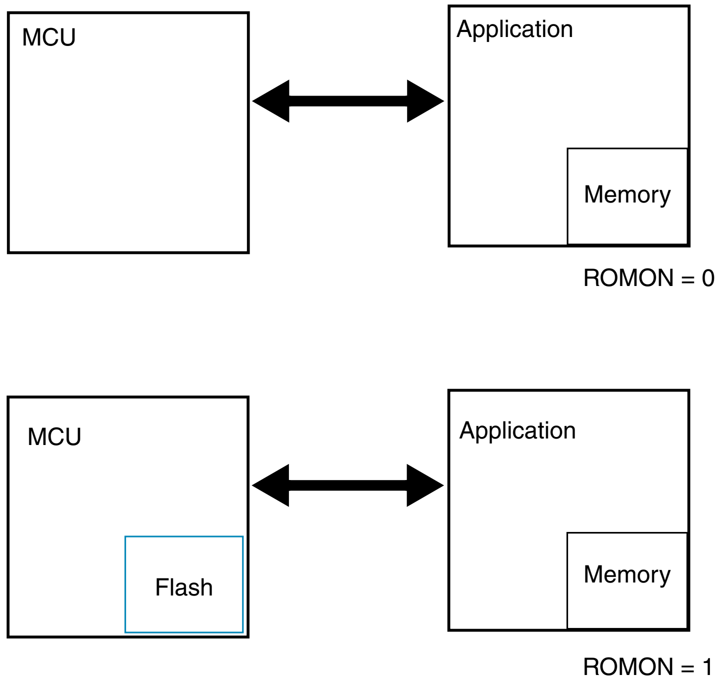


Figure 3-29. ROM in Special Test Mode

Chapter 4
Memory Protection Unit (S12XMPUV1)

Table 4-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V01.04	14 Sep 2005	4.3.1.1/4-231 4.4.1/4-237	- Added note to only use the CPU to clear the AE flag. - Added disclaimer to avoid changing descriptors while they are in use because of other bus-masters doing accesses.
V01.05	14 Mar 2006	4.3.1.1/4-231 4.4/4-237	- Clarified that interrupt generation is independent of AEF bit state. - Corrected preliminary statement about execution of violating accesses.
V01.06	09 Oct 2006		- Made Revision History entries public.

4.1 Introduction

The MPU module provides basic functionality required to protect memory mapped resources from undesired accesses. Multiple address range comparators compare memory accesses against eight memory protection descriptors located in the MPU module to determine if each access is valid or not. The comparison is sensitive to which bus master generates the access and the type of the access.

The MPU module can be used to isolate memory ranges accessible by different bus masters. It can be also be used by an operating system or software kernel to isolate the regions of memory “legally” available to specific software tasks, with the kernel re-configuring the task specific memory protection descriptors in supervisor state during task-switching.

4.1.1 Preface

The following terms and abbreviations are used in the document.

Table 4-2. Terminology

Term	Meaning
MCU	Micro-Controller Unit
MPU	Memory Protection Unit
CPU	S12X Central Processing Unit (see S12XCPU Reference Manual)
XGATE	XGATE Co-processor (see XGATE chapter)
supervisor state	refers to the supervisor state of the S12XCPU (see S12XCPU Reference Manual)
user state	refers to the user state of the S12XCPU (see S12XCPU Reference Manual)

4.1.2 Overview

The MPU module monitors the bus activity of each bus master. The data describing each access is fed into multiple address range comparators. The output of the comparators is used to determine if a particular access is allowed or represents an access violation. If an access violation caused by the S12X CPU is detected, the MPU module raises an access violation interrupt. If the MPU module detects an access violation caused by a bus master other than the S12X CPU, it flags an access error condition to the respective master. In addition to the restrictions defined for memory ranges in the MPU descriptors, accesses to memory not covered by any MPU descriptor (even read accesses!) are considered access violations.

Figure 4-1 shows a block diagram of the MPU module.

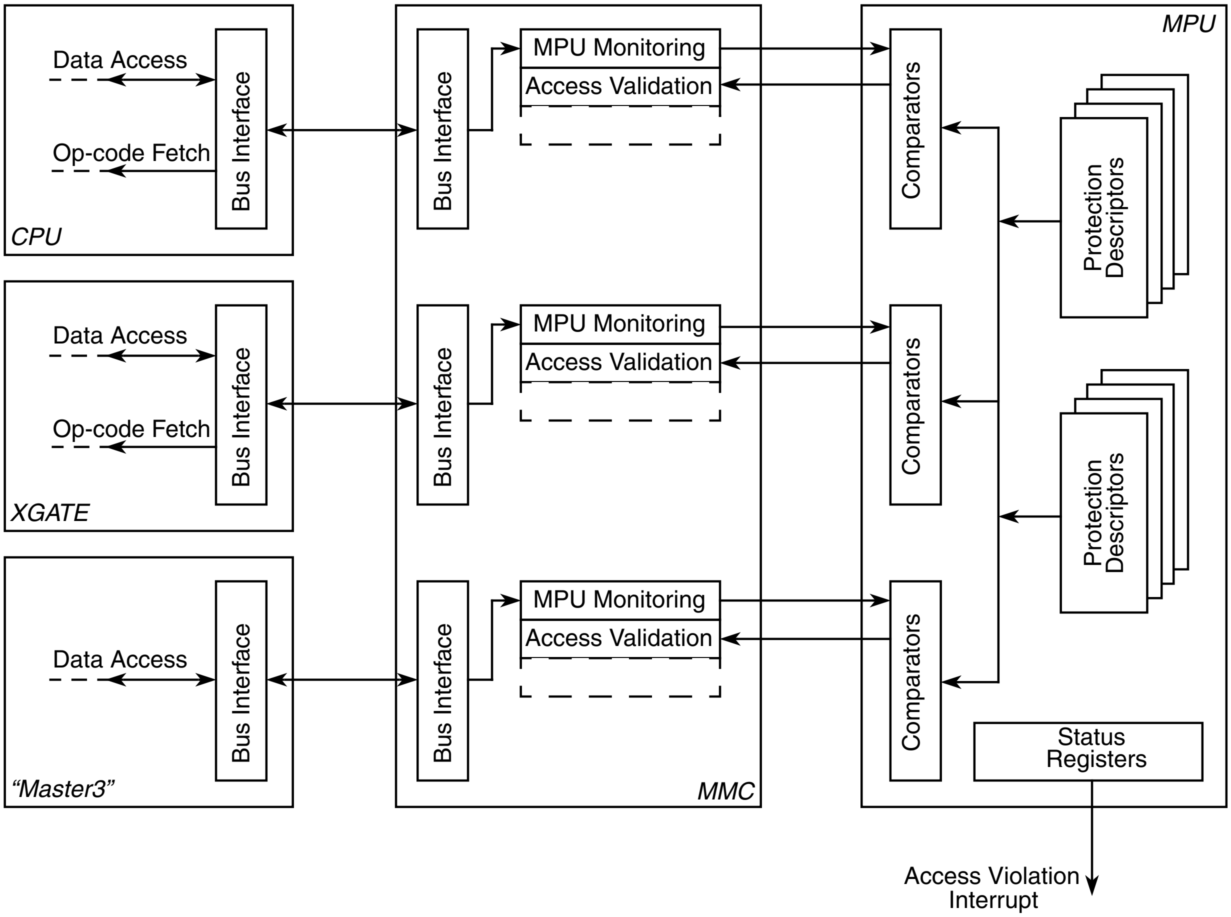


Figure 4-1. Block Diagram

4.1.3 Features

- Protects memory from undesired accesses coming from up to 3 bus masters[1]
- Eight memory protection descriptors
  - — each descriptor can cover the full global memory map (8 MBytes)
  - — each descriptor has a granularity of 8 Bytes

[1] Master 3 can be implemented or left out depending the chip configuration. Please refer to the Device Reference Manual for information about the availability and function of Master 3.

- Each descriptor can be configured to allow one of four types of access privilege for the defined memory region
  - Bus master has full access (read, write and execute enabled)
  - Bus master can read and execute (write illegal)
  - Bus master can read and write (execution illegal)
  - Bus master can only read (write and execution illegal)
- Accesses to memory not covered by any protection descriptor will cause an access violation

4.1.4 Modes of Operation

The MPU module can be used in all MCU modes.

4.2 External Signal Description

The MPU module has no external signals.

4.3 Memory Map and Register Definition

This section provides a detailed description of address space and registers used by the MPU module.

4.3.1 Register Descriptions

This section describes in address order all the MPU module registers and their individual bits.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 MPUFLG	R	AEF	WPF	NEXF	0	0	0	0	SVSF
	W		▨	▨	▨	▨	▨	▨	▨
0x0001 MPUASTAT0	R	0	ADDR[22:16]						
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0002 MPUASTAT1	R	ADDR[15:8]							
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0003 MPUASTAT2	R	ADDR[7:0]							
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0004 Reserved	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0005 MPUSEL	R	SVSEN	0	0	0	0	SEL[2:0]		
	W		▨	▨	▨	▨			
0x0006 MPUDESC0[1]	R	MSTR0	MSTR1	MSTR2	MSTR3	LOW_ADDR[22:19]			
	W								
0x0007 MPUDESC1[1]	R	LOW_ADDR[18:11]							
	W								
0x0008 MPUDESC2[1]	R	LOW_ADDR[10:3]							
	W								
0x0009 MPUDESC3[1]	R	WP	NEX	0	0	HIGH_ADDR[22:19]			
	W			▨	▨				
0x000A MPUDESC4[1]	R	HIGH_ADDR[18:11]							
	W								
0x000B MPUDESC5[1]	R	HIGH_ADDR[10:3]							
	W								

▨ = Unimplemented or Reserved

[1] The module addresses 0x0006–0x000B represent a window in the register map through which different descriptor registers are visible.

Figure 4-2. MPU Register Summary

4.3.1.1 MPU Flag Register (MPUFLG)

Address: Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	AEF	WPF	NEXF	0	0	0	0	SVSF
W								
Reset	0	0	0	0	0	0	0	0

Figure 4-3. MPU Flag Register (MPUFLG)

Read: Anytime

Write: Write of 1 clears flag, write of 0 ignored

Table 4-3. MPUFLG Field Descriptions

Field	Description
7 AEF	**Access Error Flag** — This bit is the CPU access error interrupt flag. It is set if a CPU access violation has occurred. At the same time this bit is set, all the other status flags in this register and the access violation address bits in the MPUASTATn registers are captured. Clear this flag by writing a one. **Note:** If a CPU access error is flagged and both the WPF bit and the NEXF bit are zero, the access violation was caused by an access to memory not covered by the MPU descriptors. **Note:** While this bit is set, the CPU in supervisor state ("Master 0") can read from and write to the peripheral register space even if there is no memory protection descriptor explicitly allowing this. This is to prevent the case that the CPU cannot clear the AEF bit if the registers are write protected for the CPU in supervisor state. **Note:** This bit should only be cleared by an access from the S12X CPU. Otherwise, when using one of the other masters (such as the XGATE) to clear this bit, the status flags and the address status registers may not get updated correctly if a CPU access causes a violation in the same bus cycle.
6 WPF	**Write-Protect Violation Flag** — This flag is set if the current CPU access violation has occurred because of an attempt to write to memory configured as read-only. The WPF bit is read-only; it will be automatically updated when the next access violation is flagged with the AEF bit.
5 NEXF	**No-Execute Violation Flag** — This bit is set if the current CPU access violation has occurred because of an attempt to fetch code from memory configured as No-Execute. The NEXF bit is read-only; it will be automatically updated when the next access violation is flagged with the AEF bit.
0 SVSF	**Supervisor State Flag** — This bit is set if the current CPU access violation occurred while the CPU was in supervisor state. This bit is cleared if the current CPU access violation occurred while the CPU was in user state. The supervisor state flag is read-only; it will be automatically updated when the next CPU access violation is flagged with the AEF bit.

If the AEF bit is set further violations are not captured into the MPU status registers. The status of the AEF bit has no effect on the access restrictions, i.e. access restrictions for all masters are still enforced if the AEF bit is set. Also, the non-maskable hardware interrupt for violating accesses coming from the S12X CPU is generated regardless of the state of the AEF bit.

4.3.1.2 MPU Address Status Register 0 (MPUASTAT0)

Address: Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	0	ADDR[22:16]						
W								
Reset	0	0	0	0	0	0	0	0

Figure 4-4. MPU Address Status Register 0 (MPUASTAT0)

Read: Anytime

Write: Never

Table 4-4. MPUASTAT0 Field Descriptions

Field	Description
6–0 ADDR[22:16]	**Access violation address bits** — The ADDR[22:16] bits contain bits [22:16] of the global address which caused the current access violation interrupt. These bits are undefined if the access error flag bit (AEF) in the MPUFLG register is not set.

4.3.1.3 MPU Address Status Register 1 (MPUASTAT1)

Address: Module Base + 0x0002

	7	6	5	4	3	2	1	0
R	ADDR[15:8]							
W								
Reset	0	0	0	0	0	0	0	0

Figure 4-5. MPU Address Status Register 1 (MPUASTAT1)

Read: Anytime

Write: Never

Table 4-5. MPUASTAT1 Field Descriptions

Field	Description
7–0 ADDR[15:8]	**Access violation address bits** — The ADDR[15:8] bits contain bits [15:8] of the global address which caused the current access violation interrupt. These bits are undefined if the access error flag bit (AEF) in the MPUFLG register is not set.

4.3.1.4 MPU Address Status Register 2 (MPUASTAT2)

Address: Module Base + 0x0003

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

Figure 4-6. MPU Address Status Register (MPUASTAT2)

Read: Anytime

Write: Never

Table 4-6. MPUASTAT2 Field Descriptions

Field	Description
7–0 ADDR[7:0]	**Access violation address bits** — The ADDR[7:0] bits contain bits [7:0] of the global address which caused the current access violation interrupt. These bits are undefined if the access error flag bit (AEF) in the MPUFLG register is not set.

4.3.1.5 MPU Descriptor Select Register (MPUSEL)

Address: Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	SVSEN	0	0	0	0	SEL[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

Figure 4-7. MPU Descriptor Select Register (MPUSEL)

Read: Anytime

Write: Anytime

Table 4-7. MPUSEL Field Descriptions

Field	Description
7 SVSEN	**MPU supervisor state enable bit** — This bit enables the memory protection for the CPU in supervisor state. If this bit is cleared, the MPU does not affect any accesses coming from the CPU in supervisor state. This is to prevent the CPU from locking out itself while configuring the protection descriptors (during initialization after a system reset and during the update of the protection descriptors for a task switch). The memory protection functionality for the other bus-masters is unaffected by this bit. 0 MPU is disabled for the CPU in supervisor state 1 MPU is enabled for the CPU in supervisor state
2–0 SEL[2:0]	**Descriptor select bits** — The SEL[2:0] bits select which descriptor is visible in the MPU Descriptor Register window (MPUDESC0—MPUDESC5).

4.3.1.6 MPU Descriptor Register 0 (MPUDESC0)

Address: Module Base + 0x0006

	7	6	5	4	3	2	1	0
R W	MSTR0	MSTR1	MSTR2	MSTR3	LOW_ADDR[22:19]			
Reset	1[1]	1[1]	1[1]	1[2]	0	0	0	0

1. initialized as set for descriptor 0 only, cleared for all others
2. initialized as set for descriptor 0 only, if MSTR3 is implemented on the device

Figure 4-8. MPU Descriptor Register 0 (MPUDESC0)

Read: Anytime

Write: Anytime

Table 4-8. MPUDESC0 Field Descriptions

Field	Description
7 MSTR0	**Master 0 select bit** — If this bit is set the descriptor is valid for bus master 0 (CPU in supervisor state).
6 MSTR1	**Master 1 select bit** — If this bit is set the descriptor is valid for bus master 1 (CPU in user state).
5 MSTR2	**Master 2 select bit** — If this bit is set the descriptor is valid for bus master 2 (XGATE).
4 MSTR3	**Master 3 select bit** — If this bit is set the descriptor is valid for bus master 3.
3–0 LOW_ADDR[22:19]	**Memory range lower boundary address bits** — The LOW_ADDR[22:19] bits represent bits [22:19] of the global memory address that is used as the lower boundary for the described memory range.

A descriptor can be configured as valid for more than one bus-master at the same time by setting multiple Master select bits to one. Setting all Master select bits of a descriptor to zero disables the descriptor.

4.3.1.7 MPU Descriptor Register 1 (MPUDESC1)

Address: Module Base + 0x0007

	7	6	5	4	3	2	1	0
R W	LOW_ADDR[18:11]							
Reset	0	0	0	0	0	0	0	0

Figure 4-9. MPU Descriptor Register 1 (MPUDESC1)

Read: Anytime

Write: Anytime

Table 4-9. MPUDESC1 Field Descriptions

Field	Description
7–0 LOW_ADDR[18:11]	**Memory range lower boundary address bits** — The LOW_ADDR[18:11] bits represent bits [18:11] of the global memory address that is used as the lower boundary for the described memory range.

4.3.1.8 MPU Descriptor Register 2 (MPUDESC2)

Address: Module Base + 0x0008

	7	6	5	4	3	2	1	0
R W	LOW_ADDR[10:3]							
Reset	0	0	0	0	0	0	0	0

Figure 4-10. MPU Descriptor Register 2 (MPUDESC2)

Read: Anytime

Write: Anytime

Table 4-10. MPUDESC2 Field Descriptions

Field	Description
7–0 LOW_ADDR[10:3]	**Memory range lower boundary address bits** — The LOW_ADDR[10:3] bits represent bits [10:3] of the global memory address that is used as the lower boundary for the described memory range.

4.3.1.9 MPU Descriptor Register 3 (MPUDESC3)

Address: Module Base + 0x0009

	7	6	5	4	3	2	1	0
R	WP	NEX	0	0	HIGH_ADDR[22:19]			
W			▨	▨				
Reset	0	0	0	0	1	1	1	1

Figure 4-11. MPU Descriptor Register 3 (MPUDESC3)

Read: Anytime

Write: Anytime

Table 4-11. MPUDESC3 Field Descriptions

Field	Description
7 WP	**Write-Protect bit** — The WP bit causes the described memory range to be treated as write-protected. If this bit is set every attempt to write in the described memory range causes an access violation.

Field	Description
6 NEX	**No-Execute bit** — The NEX bit prevents the described memory range from being used as code memory. If this bit is set every Op-code fetch in this memory range causes an access violation.
3–0 HIGH_ADDR[22:19]	**Memory range upper boundary address bits** — The HIGH_ADDR[22:19] bits represent bits [22:19] of the global memory address that is used as the upper boundary for the described memory range.

4.3.1.10 MPU Descriptor Register 4 (MPUDESC4)

Address: Module Base + 0x000A

	7	6	5	4	3	2	1	0
R W	HIGH_ADDR[18:11]							
Reset	1	1	1	1	1	1	1	1

Figure 4-12. MPU Descriptor Register 4 (MPUDESC4)

Read: Anytime

Write: Anytime

Table 4-12. MPUDESC4 Field Descriptions

Field	Description
7–0 HIGH_ADDR[18:11]	**Memory range upper boundary address bits** — The HIGH_ADDR[18:11] bits represent bits [18:11] of the global memory address that is used as the upper boundary for the described memory range.

4.3.1.11 MPU Descriptor Register 5 (MPUDESC5)

Address: Module Base + 0x000B

	7	6	5	4	3	2	1	0
R W	HIGH_ADDR[10:3]							
Reset	1	1	1	1	1	1	1	1

Figure 4-13. MPU Descriptor Register 5 (MPUDESC5)

Read: Anytime

Write: Anytime

Table 4-13. MPUDESC5 Field Descriptions

Field	Description
7–0 HIGH_ADDR[10:3]	**Memory range upper boundary address bits** — The HIGH_ADDR[10:3] bits represent bits [10:3] of the global memory address that is used as the upper boundary for the described memory range.

4.4 Functional Description

The MPU module provides memory protection for accesses coming from multiple masters in the system. This is done by monitoring bus traffic of each master and compare this with the configuration information from a set of eight programmable descriptors located in the MPU module. If the MPU module detects an access violation caused by the S12X CPU, it will assert the CPU access violation interrupt signal. If the MPU module detects an access violation caused by a bus master other than the S12X CPU, it raises an access error signal. Please refer to the documentation chapter of the individual master modules (i.e. XGATE, etc.) for more information about the access error condition.

Violating accesses are not executed. The return value of a violating read access is undefined for both 8 bit and 16 bit accesses.

NOTE

Accesses from BDM are not restricted. BDM hardware accesses always bypass the MPU module. During execution of BDM firmware code S12X CPU accesses are masked from the MPU module as well.

4.4.1 Protection Descriptors

Each of the eight protection descriptors can be used to restrict the allowed types of memory accesses for a given memory range. Each of these memory ranges can cover up the entire 23 bits global memory range (8 MBytes).

The descriptors are banked in the MPU module register map.

Each descriptor can be selected for modifying using the SEL bits in the MPU Descriptor Select (MPUSEL) register.

Table 4-14 gives an overview of the types of accesses that can be configured using the protection descriptors.

Table 4-14. Access Types

WP	NEX	Meaning
0	0	read, write and execute
0	1	read, write
1	0	read and execute
1	1	read only

The granularity of each descriptor is 8 bytes. This means the protection comparators in the MPU module cover only address bits [22:3] of each access. The lower address bits [2:0] are ignored.

NOTE

A mis-aligned word access to the upper boundary address of a descriptor is always flagged as an access violation.

NOTE

Configuring the lower boundary address of a descriptor to be higher than the upper boundary address of a descriptor causes this descriptor to be ignored by the comparator block. This effectively disables the descriptor.

NOTE

Avoid changing descriptors while they are in active use to validate accesses from bus-masters. This can be done by temporarily disabling the affected master during the update (XGATE, Master 3, switch S12X CPU states). Otherwise accesses from bus-masters affected by a descriptor which is updated concurrently could yield undefined results.

4.4.1.1 Overlapping Descriptors

If the memory ranges of two protection descriptors defined for the same bus-master overlap, the access restrictions for the overlapped memory range are accumulated. For example:

- a memory protection descriptor defines memory range 0x40_0000–0x41_FFFF as WP=1, NEX=0 (read and execute)
- another descriptor defines memory range 0x41_0000–0x43_FFFF as WP=0, NEX=1 (read and write)
- the resulting access rights for the overlapping range 0x41_0000–0x41_FFFF are WP=1, NEX=1 (read only)

4.4.1.2 Implicitly defined memory descriptors

As mentioned in the bit description of the Access Error Flag (AEF) in the MPUFLG register (Table 4-3), there is an additional memory range implicitly defined only while the AEF bit is set: The CPU in supervisor state can read from and write to the peripheral register space even if there is no memory protection descriptor explicitly allowing this. This is to prevent the case that the CPU cannot clear the AEF bit if the registers are write protected for the CPU in supervisor state.

The register address space containing the PAGE registers (EPAGE, RPAGE, GPAGE, PPAGE) at 0x0010–0x0017 gets special treatment. It is defined like this:

- The S12X CPU can always read and write these registers, regardless of the configuration in the descriptors.
- XGATE or Master3 (if available) are never allowed to read or write these registers, even if the descriptor configuration allows accesses for other masters than the S12X CPU.

4.4.1.3 Op-code pre-fetch cycles and the NEX bit

Some bus-masters (CPU, XGATE) do a pre-fetch of program-code past the current instruction. The S12XCPU pre-fetches two words past the current instruction, the XGATE pre-fetches one word, even if the pre-fetched code is not executed. The MPU module has no way of knowing this at the time when the pre-fetch cycles occur. Therefore this will result in an access violation if the op-code pre-fetch accesses a memory range marked as “No-Execute” (NEX=1). This must be taken into account when defining memory ranges with the NEX bit set adjacent to memory used for program code. The best way to do this would be to leave some fill-bytes between the memory ranges in this case, i.e. do not set the upper memory boundary to the address of the last op-code but to a following address which is at least two words (four bytes) away.

4.4.2 Interrupts

This section describes all interrupts originated by the MPU module.

4.4.2.1 Description of Interrupt Operation

The MPU module generates one interrupt request. It cannot be masked locally in the MPU module and is meant to be used as the source of a non-maskable hardware interrupt request for the S12X CPU

Table 4-15. Interrupt vectors

Interrupt Source	CCR Mask	Local Enable
S12X CPU access error interrupt (AEF)	–	–

4.4.2.2 CPU Access Error Interrupt

An S12X CPU access error interrupt request is generated if the MPU module has detected an illegal memory access originating from the S12X CPU. This is a non-maskable hardware interrupt. Due to the non-maskable nature of this interrupt, the de-assertion of this interrupt request is coupled to the S12X CPU interrupt vector fetch instead of the local access error flag (AEF). This means leaving the access error flag (AEF) in the MPUFLG register set will not cause the same interrupt to be serviced again after leaving the interrupt service routine with “RTI”. Instead, the interrupt request will be asserted again only when the next illegal S12X CPU access is detected.

4.5 Initialization/Application Information

4.5.1 Initialization

After reset the MPU module is in an unconfigured state, with all eight protection descriptors covering the whole memory map. The master bits are all set for descriptor “0” and cleared for all other descriptors. The S12XCPU in supervisor state can access everything because the SVSEN bit in the MPUSEL register is cleared by a system reset. After system reset every master has full access to the memory map because of descriptor “0”.

In order to use the MPU module to protect memory ranges from undesired accesses, software needs to:

- Initialize the protection descriptors.
- Make sure there are meaningful interrupt service routines defined for the Access Violation interrupts because these are non-maskable (See S12XINT chapter for details).
- Initialize peripherals and other masters for use (i.e. set-up XGATE, Master3 if applicable).
- Enable the MPU protection for the S12X CPU in supervisor state, if desired.
- Switch the S12X CPU to user state, if desired.

Chapter 5
External Bus Interface (S12XEBIV4)

Table 5-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V04.01	12 Sep 2005		- Added CSx stretch description.
V04.02	23 May 2006		- Internal updates
V04.03	24 Jul 2006		- Removed term IVIS

5.1 Introduction

This document describes the functionality of the XEBI block controlling the external bus interface.

The XEBI controls the functionality of a non-multiplexed external bus (a.k.a. ‘expansion bus’) in relationship with the chip operation modes. Dependent on the mode, the external bus can be used for data exchange with external memory, peripherals or PRU, and provide visibility to the internal bus externally in combination with an emulator.

5.1.1 Glossary or Terms

bus clock	System Clock. Refer to CRG Block Guide.
expanded modes	Normal Expanded Mode Emulation Single-Chip Mode Emulation Expanded Mode Special Test Mode
single-chip modes	Normal Single-Chip Mode Special Single-Chip Mode
emulation modes	Emulation Single-Chip Mode Emulation Expanded Mode
normal modes	Normal Single-Chip Mode Normal Expanded Mode
special modes	Special Single-Chip Mode Special Test Mode
NS	Normal Single-Chip Mode
SS	Special Single-Chip Mode
NX	Normal Expanded Mode
ES	Emulation Single-Chip Mode
EX	Emulation Expanded Mode
ST	Special Test Mode
external resource	Addresses outside MCU
PRR	Port Replacement Registers
PRU	Port Replacement Unit
EMULMEM	External emulation memory
access source	CPU or BDM or XGATE

5.1.2 Features

The XEBI includes the following features:

- Output of up to 23-bit address bus and control signals to be used with a non-muxed external bus
- Bidirectional 16-bit external data bus with option to disable upper half
- Visibility of internal bus activity

5.1.3 Modes of Operation

- Single-chip modes

  The external bus interface is not available in these modes.
- Expanded modes

  Address, data, and control signals are activated on the external bus in normal expanded mode and special test mode.
- Emulation modes

  The external bus is activated to interface to an external tool for emulation of normal expanded mode or normal single-chip mode applications.

Refer to the S12X_MMC section for a detailed description of the MCU operating modes.

5.1.4 Block Diagram

Figure 5-1 is a block diagram of the XEBI with all related I/O signals.

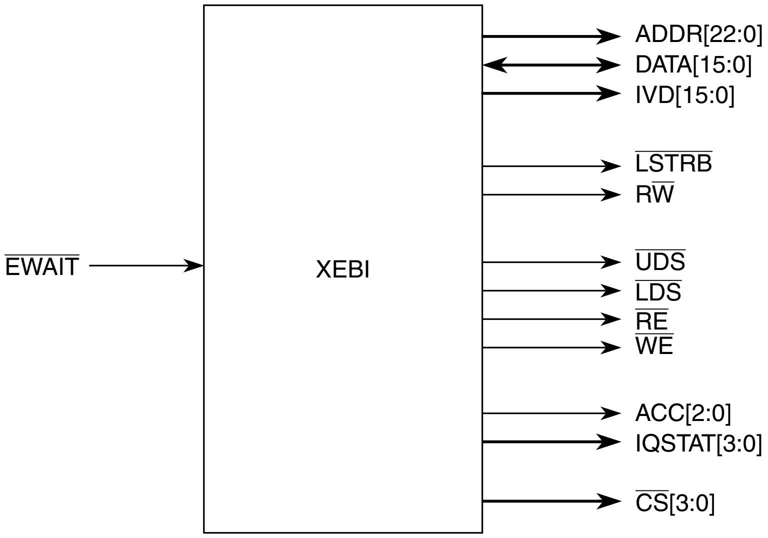


Figure 5-1. XEBI Block Diagram

5.2 External Signal Description

The user is advised to refer to the SoC section for port configuration and location of external bus signals.

NOTE

The following external bus related signals are described in other sections:
ECLK, ECLKX2 (free-running clocks) — PIM section
$\overline{\text{TAGHI}}$, $\overline{\text{TAGLO}}$ (tag inputs) — PIM section, S12X_DBG section

Table 5-2 outlines the pin names and gives a brief description of their function. Refer to the SoC section and PIM section for reset states of these pins and associated pull-ups or pull-downs.

Table 5-2. External System Signals Associated with XEBI

Signal	I[1]/O	EBI Signal Multiplex (T)ime[2] (F)unction[3]		Description	Available in Modes					
					NS	SS	NX	ES	EX	ST
$\overline{RE}$	O	—	—	Read Enable, indicates external read access	No	No	Yes	No	No	No
ADDR[22:20]	O	T	—	External address	No	No	Yes	Yes	Yes	Yes
ACC[2:0]	O		—	Access source	No	No	No	Yes	Yes	Yes
ADDR[19:16]	O	T	—	External address	No	No	Yes	Yes	Yes	Yes
IQSTAT[3:0]	O		—	Instruction Queue Status	No	No	No	Yes	Yes	Yes
ADDR[15:1]	O	T	—	External address	No	No	Yes	Yes	Yes	Yes
IVD[15:1]	O		—	Internal visibility read data	No	No	No	Yes	Yes	Yes
ADDR0	O	T	F	External address	No	No	No	Yes	Yes	Yes
IVD0	O			Internal visibility read data	No	No	No	Yes	Yes	Yes
$\overline{UDS}$	O	—		Upper Data Select, indicates external access to the high byte DATA[15:8]	No	No	Yes	No	No	No
$\overline{LSTRB}$	O	—	F	Low Strobe, indicates valid data on DATA[7:0]	No	No	No	Yes	Yes	Yes
$\overline{LDS}$	O	—		Lower Data Select, indicates external access to the low byte DATA[7:0]	No	No	Yes	No	No	No
R$\overline{W}$	O	—	F	Read/Write, indicates the direction of internal data transfers	No	No	No	Yes	Yes	Yes
$\overline{WE}$	O	—		Write Enable, indicates external write access	No	No	Yes	No	No	No
$\overline{CS}$[3:0]	O	—	—	Chip select	No	No	Yes	No	Yes	No
DATA[15:8]	I/O	—	—	Bidirectional data (even address)	No	No	Yes	Yes	Yes	Yes
DATA[7:0]	I/O	—	—	Bidirectional data (odd address)	No	No	Yes	Yes	Yes	Yes
$\overline{EWAIT}$	I	—	—	External control for external bus access stretches (adding wait states)	No	No	Yes	No	Yes	No

[1] All inputs are capable of reducing input threshold level

[2] Time-multiplex means that the respective signals share the same pin on chip level and are active alternating in a dedicated time slot (in modes where applicable).

[3] Function-multiplex means that one of the respective signals sharing the same pin on chip level continuously uses the pin depending on configuration and reset state.

5.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the XEBI.

5.3.1 Module Memory Map

The registers associated with the XEBI block are shown in Figure 5-2.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0E EBICTL0	R	ITHRS	0	HDBE	ASIZ4	ASIZ3	ASIZ2	ASIZ1	ASIZ0
	W		▨						
0x0F EBICTL1	R	0	EXSTR12	EXSTR11	EXSTR10	0	EXSTR02	EXSTR01	EXSTR00
	W	▨				▨			

▨ = Unimplemented or Reserved

Figure 5-2. XEBI Register Summary

5.3.2 Register Descriptions

The following sub-sections provide a detailed description of each register and the individual register bits.

All control bits can be written anytime, but this may have no effect on the related function in certain operating modes. This allows specific configurations to be set up before changing into the target operating mode.

NOTE

Depending on the operating mode an available function may be enabled, disabled or depend on the control register bit. Reading the register bits will reflect the status of related function only if the current operating mode allows user control. Please refer the individual bit descriptions.

5.3.2.1 External Bus Interface Control Register 0 (EBICTL0)

Module Base +0x000E (PRR)

	7	6	5	4	3	2	1	0
R	ITHRS	0	HDBE	ASIZ4	ASIZ3	ASIZ2	ASIZ1	ASIZ0
W		▨						
Reset	0	0	1	1	1	1	1	1

▨ = Unimplemented or Reserved

Figure 5-3. External Bus Interface Control Register 0 (EBICTL0)

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

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

This register controls input pin threshold level and determines the external address and data bus sizes in normal expanded mode. If not in use with the external bus interface, the related pins can be used for alternative functions.

External bus is available as programmed in normal expanded mode and always full-sized in emulation modes and special test mode; function not available in single-chip modes.

Table 5-3. EBICTL0 Field Descriptions

Field	Description
7 ITHRS	**Reduced Input Threshold** — This bit selects reduced input threshold on external data bus pins and specific control input signals which are in use with the external bus interface in order to adapt to external devices with a 3.3 V, 5 V tolerant I/O. The reduced input threshold level takes effect depending on ITHRS, the operating mode and the related enable signals of the EBI pin function as summarized in Table 5-4. 0 Input threshold is at standard level on all pins 1 Reduced input threshold level enabled on pins in use with the external bus interface
5 HDBE	**High Data Byte Enable** — This bit enables the higher half of the 16-bit data bus. If disabled, only the lower 8-bit data bus can be used with the external bus interface. In this case the unused data pins and the data select signals ($\overline{\text{UDS}}$ and $\overline{\text{LDS}}$) are free to be used for alternative functions. 0 DATA[15:8], $\overline{\text{UDS}}$, and $\overline{\text{LDS}}$ disabled 1 DATA[15:8], $\overline{\text{UDS}}$, and $\overline{\text{LDS}}$ enabled
4–0 ASIZ[4:0]	**External Address Bus Size** — These bits allow scalability of the external address bus. The programmed value corresponds to the number of available low-aligned address lines (refer to Table 5-5). All address lines ADDR[22:0] start up as outputs after reset in expanded modes. This needs to be taken into consideration when using alternative functions on relevant pins in applications which utilize a reduced external address bus.

Table 5-4. Input Threshold Levels on External Signals

ITHRS	External Signal	NS	SS	NX	ES	EX	ST
0	DATA[15:8] $\overline{\text{TAGHI}}$, $\overline{\text{TAGLO}}$	Standard	Standard	Standard	Reduced	Reduced	Standard
	DATA[7:0]						
	$\overline{\text{EWAIT}}$				Standard	Standard	
1	DATA[15:8] $\overline{\text{TAGHI}}$, $\overline{\text{TAGLO}}$	Standard	Standard	Reduced if HDBE = 1	Reduced	Reduced	Reduced
	DATA[7:0]			Reduced			
	$\overline{\text{EWAIT}}$			Reduced if EWAIT enabled[1]	Standard	Reduced if EWAIT enabled[1]	Standard

[1] EWAIT function is enabled if at least one $\overline{\text{CSx}}$ line is configured respectively in MMCCTL0. Refer to S12X_MMC section and Table 5-6.

Table 5-5. External Address Bus Size

ASIZ[4:0]	Available External Address Lines
00000	None
00001	$\overline{\text{UDS}}$
00010	ADDR1, $\overline{\text{UDS}}$

Table 5-5. External Address Bus Size

ASIZ[4:0]	Available External Address Lines
00011	ADDR[2:1], $\overline{\text{UDS}}$
:	:
10110	ADDR[21:1], $\overline{\text{UDS}}$
10111 : 11111	ADDR[22:1], $\overline{\text{UDS}}$

5.3.2.2 External Bus Interface Control Register 1 (EBICTL1)

Module Base +0x000F (PRR)

	7	6	5	4	3	2	1	0
R	0	EXSTR12	EXSTR11	EXSTR10	0	EXSTR02	EXSTR01	EXSTR00
W	▨				▨			
Reset	0	1	1	1	0	1	1	1

▨ = Unimplemented or Reserved

Figure 5-4. External Bus Interface Control Register 1 (EBICTL1)

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

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

This register allows programming of two independent values determining the amount of additional stretch cycles for external accesses (wait states).

With two bits in S12X_MMC register MMCCTL0 for every individual $\overline{\text{CSx}}$ line one of the two counter options or the $\overline{\text{EWAIT}}$ input is selected as stretch source. The chip select outputs can also be disabled to free up the pins for alternative functions (Table 5-6). Refer also to S12X_MMC section for register bit descriptions.

Table 5-6. Chip select function

CSxE1	CSxE0	Function
0	0	$\overline{\text{CSx}}$ disabled
0	1	$\overline{\text{CSx}}$ stretched with EXSTR0
1	0	$\overline{\text{CSx}}$ stretched with EXSTR1
1	1	$\overline{\text{CSx}}$ stretched with $\overline{\text{EWAIT}}$

If $\overline{\text{EWAIT}}$ input usage is selected in MMCCTL0 the minimum number of stretch cycles is 2 for accesses to the related address range.

If configured respectively, stretch cycles are added as programmed or dependent on $\overline{\text{EWAIT}}$ in normal expanded mode and emulation expanded mode; function not available in all other operating modes.

Table 5-7. EBICTL1 Field Descriptions

Field	Description
6–4 EXSTR1[2:0]	**External Access Stretch Option 1 Bits 2, 1, 0** — This three bit field determines the amount of additional clock stretch cycles on every access to the external address space as shown in Table 5-8.
2–0 EXSTR0[2:0]	**External Access Stretch Option 0 Bits 2, 1, 0** — This three bit field determines the amount of additional clock stretch cycles on every access to the external address space as shown in Table 5-8.

Table 5-8. External Access Stretch Bit Definition

EXSTRx[2:0]	Number of Stretch Cycles
000	1
001	2
010	3
011	4
100	5
101	6
110	7
111	8

5.4 Functional Description

This section describes the functions of the external bus interface. The availability of external signals and functions in relation to the operating mode is initially summarized and described in more detail in separate sub-sections.

5.4.1 Operating Modes and External Bus Properties

A summary of the external bus interface functions for each operating mode is shown in Table 5-9.

Table 5-9. Summary of Functions

Properties (if Enabled)	Single-Chip Modes		Expanded Modes			
	Normal Single-Chip	Special Single-Chip	Normal Expanded	Emulation Single-Chip	Emulation Expanded	Special Test
Timing Properties						
PRR access[1]	2 cycles read internal write internal	2 cycles read internal write internal	2 cycles read internal write internal	2 cycles read external write int & ext	2 cycles read external write int & ext	2 cycles read internal write internal
Internal access visible externally	—	—	—	1 cycle	1 cycle	1 cycle
External address access and unimplemented area access[2]	—	—	Max. of 2 to 9 programmed cycles or n cycles of ext. wait[3]	1 cycle	Max. of 2 to 9 programmed cycles or n cycles of ext. wait[3]	1 cycle

Table 5-9. Summary of Functions (continued)

Properties (if Enabled)	Single-Chip Modes		Expanded Modes			
	Normal Single-Chip	**Special Single-Chip**	**Normal Expanded**	**Emulation Single-Chip**	**Emulation Expanded**	**Special Test**
Flash area address access[4]	—	—	—	1 cycle	1 cycle	1 cycle
Signal Properties						
Bus signals	—	—	ADDR[22:1] DATA[15:0]	ADDR[22:20]/ ACC[2:0] ADDR[19:16]/ IQSTAT[3:0] ADDR[15:0]/ IVD[15:0] DATA[15:0]	ADDR[22:20]/ ACC[2:0] ADDR[19:16]/ IQSTAT[3:0] ADDR[15:0]/ IVD[15:0] DATA[15:0]	ADDR[22:0] DATA[15:0]
Data select signals (if 16-bit data bus)	—	—	$\overline{\text{UDS}}$ $\overline{\text{LDS}}$	ADDR0 $\overline{\text{LSTRB}}$	ADDR0 $\overline{\text{LSTRB}}$	ADDR0 $\overline{\text{LSTRB}}$
Data direction signals	—	—	$\overline{\text{RE}}$ $\overline{\text{WE}}$	R$\overline{\text{W}}$	R$\overline{\text{W}}$	R$\overline{\text{W}}$
Chip Selects	—	—	$\overline{\text{CS0}}$ $\overline{\text{CS1}}$ $\overline{\text{CS2}}$ $\overline{\text{CS3}}$	—	$\overline{\text{CS0}}$ $\overline{\text{CS1}}$ $\overline{\text{CS2}}$ $\overline{\text{CS3}}$	—
External wait feature	—	—	$\overline{\text{EWAIT}}$	—	$\overline{\text{EWAIT}}$	—
Reduced input threshold enabled on	—	—	Refer to Table 5-4	DATA[15:0] $\overline{\text{EWAIT}}$	DATA[15:0] $\overline{\text{EWAIT}}$	Refer to Table 5-4

[1] Incl. S12X_EBI registers

[2] Refer to S12X_MMC section.

[3] If EWAIT enabled for at least one $\overline{\text{CSx}}$ line (refer to S12X_MMC section), the minimum number of external bus cycles is 3.

[4] Available only if configured appropriately by ROMON and EROMON (refer to S12X_MMC section).

5.4.2 Internal Visibility

Internal visibility allows the observation of the internal CPU address and data bus as well as the determination of the access source and the CPU pipe (queue) status through the external bus interface.

Internal visibility is always enabled in emulation single chip mode and emulation expanded mode. Internal CPU accesses are made visible on the external bus interface except CPU execution of BDM firmware instructions.

Internal reads are made visible on ADDRx/IVDx (address and read data multiplexed, see Table 5-12 to Table 5-14), internal writes on ADDRx and DATAx (see Table 5-15 to Table 5-17). R$\overline{\text{W}}$ and $\overline{\text{LSTRB}}$ show the type of access. External read data are also visible on IVDx.

During ‘no access’ cycles R$\overline{\text{W}}$ is held in read position while $\overline{\text{LSTRB}}$ is undetermined.

All accesses which make use of the external bus interface are considered external accesses.

5.4.2.1 Access Source Signals (ACC)

The access source can be determined from the external bus control signals ACC[2:0] as shown in Table 5-10.

Table 5-10. Determining Access Source from Control Signals

ACC[2:0]	Access Description
000	Repetition of previous access cycle
001	CPU access
010	BDM external access
011	XGATE PRR access
100	No access[1]
101	CPU access error
110, 111	Reserved

[1] Denotes also CPU accesses to BDM firmware and BDM registers (IQSTATx are ‘XXXX’ and R$\overline{\text{W}}$ = 1 in these cases)

5.4.2.2 Instruction Queue Status Signals (IQSTAT)

The CPU instruction queue status (execution-start and data-movement information) is brought out as IQSTAT[3:0] signals. For decoding of the IQSTAT values, refer to the S12X_CPU section.

5.4.2.3 Internal Visibility Data (IVD)

Depending on the access size and alignment, either a word of read data is made visible on the address lines or only the related data byte will be presented in the ECLK low phase. For details refer to Table 5-11.

Invalid IVD are brought out in case of non-CPU read accesses.

Table 5-11. IVD Read Data Output

Access	IVD[15:8]	IVD[7:0]
Word read of data at an even and even+1 address	ivd(even)	ivd(even+1)
Word read of data at an odd and odd+1 internal RAM address (misaligned)	ivd(odd+1)	ivd(odd)
Byte read of data at an even address	ivd(even)	addr[7:0] (rep.)
Byte read of data at an odd address	addr[15:8] (rep.)	ivd(odd)

5.4.2.4 Emulation Modes Timing

A bus access lasts 1 ECLK cycle. In case of a stretched external access (emulation expanded mode), up to an infinite amount of ECLK cycles may be added. ADDRx values will only be shown in ECLK high phases, while ACCx, IQSTATx, and IVDx values will only be presented in ECLK low phases.

Based on this multiplex timing, ACCx are only shown in the current (first) access cycle. IQSTATx and (for read accesses) IVDx follow in the next cycle. If the access takes more than one bus cycle, ACCx display NULL (0x000) in the second and all following cycles of the access. IQSTATx display NULL (0x0000) from the third until one cycle after the access to indicate continuation.

The resulting timing pattern of the external bus signals is outlined in the following tables for read, write and interleaved read/write accesses. Three examples represent different access lengths of 1, 2, and n–1 bus cycles. Non-shaded bold entries denote all values related to Access #0.

The following terminology is used:

‘addr’ — value(ADDRx); small letters denote the logic values at the respective pins
‘x’ — Undefined output pin values
‘z’ — Tristate pins
‘?’ — Dependent on previous access (read or write); IVDx: ‘ivd’ or ‘x’; DATAx: ‘data’ or ‘z’

5.4.2.4.1 Read Access Timing

Table 5-12. Read Access (1 Cycle)

		Access #0				**Access #1**		
Bus cycle ->	...	**1**		**2**		**3**		...
ECLK phase	...	high	low	high	low	high	low	...
ADDR[22:20] / ACC[2:0]	...	**addr 0**	**acc 0**	addr 1	acc 1	addr 2	acc 2	...
ADDR[19:16] / IQSTAT[3:0]	...		iqstat -1		**iqstat 0**		iqstat 1	...
ADDR[15:0] / IVD[15:0]	...		?		**ivd 0**		ivd 1	...
DATA[15:0] (internal read)	...	?	**z**	**z**	z	z	z	...
DATA[15:0] (external read)	...	?	**z**	**data 0**	z	data 1	z	...
R$\overline{\text{W}}$	...	1	1	1	1	1	1	...

Table 5-13. Read Access (2 Cycles)

		Access #0				**Access #1**		
Bus cycle ->	...	**1**		**2**		**3**		...
ECLK phase	...	high	low	high	low	high	low	...
ADDR[22:20] / ACC[2:0]	...	**addr 0**	**acc 0**	**addr 0**	**000**	addr 1	acc 1	...
ADDR[19:16] / IQSTAT[3:0]	...		iqstat-1		**iqstat 0**		**0000**	...
ADDR[15:0] / IVD[15:0]	...		?		**x**		**ivd 0**	...
DATA[15:0] (internal read)	...	?	**z**	**z**	**z**	**z**	z	...
DATA[15:0] (external read)	...	?	**z**	**z**	**z**	**data 0**	z	...
R$\overline{\text{W}}$	...	1	1	1	1	1	1	...

Table 5-14. Read Access (n–1 Cycles)

		Access #0							**Access #1**		
Bus cycle ->	...	**1**		**2**		**3**		...	**n**		...
ECLK phase	...	high	low	high	low	high	low	...	high	low	...
ADDR[22:20] / ACC[2:0]	...	**addr 0**	**acc 0**	**addr 0**	**000**	**addr 0**	**000**	...	addr 1	acc 1	...
ADDR[19:16] / IQSTAT[3:0]	...		iqstat-1		**iqstat 0**		**0000**	...		**0000**	...
ADDR[15:0] / IVD[15:0]	...		?		**x**		**x**	...		**ivd 0**	...
DATA[15:0] (internal read)	...	?	**z**	**z**	**z**	**z**	**z**	...	**z**	z	...

Table 5-14. Read Access (n–1 Cycles)

DATA[15:0] (external read)	...	?	**z**	**z**	**z**	**z**	**z**	...	**data 0**	z	...
R$\overline{W}$	...	1	1	1	1	1	1	...	1	1	...

5.4.2.4.2 Write Access Timing

Table 5-15. Write Access (1 Cycle)

		Access #0		**Access #1**		**Access #2**		
Bus cycle ->	...	**1**		**2**		**3**		...
ECLK phase	...	high	low	high	low	high	low	...
ADDR[22:20] / ACC[2:0]	...	**addr 0**	**acc 0**	addr 1	acc 1	addr 2	acc 2	...
ADDR[19:16] / IQSTAT[3:0]	...		iqstat -1		**iqstat 0**		iqstat 1	...
ADDR[15:0] / IVD[15:0]	...		?		**x**		x	...
DATA[15:0] (write)	...	?	**data 0**		data 1		data 2	...
R$\overline{W}$	...	0	0	1	1	1	1	...

Table 5-16. Write Access (2 Cycles)

		Access #0				**Access #1**		
Bus cycle ->	...	**1**		**2**		**3**		...
ECLK phase	...	high	low	high	low	high	low	...
ADDR[22:20] / ACC[2:0]	...	**addr 0**	**acc 0**	**addr 0**	**000**	addr 1	acc 1	...
ADDR[19:16] / IQSTAT[3:0]	...		iqstat-1		**iqstat 0**		**0000**	...
ADDR[15:0] / IVD[15:0]	...		?		**x**		**x**	...
DATA[15:0] (write)	...	?	**data 0**				x	...
R$\overline{W}$	...	0	0	0	0	1	1	...

Table 5-17. Write Access (n–1 Cycles)

		Access #0							**Access #1**		
Bus cycle ->	...	**1**		**2**		**3**		...	**n**		...
ECLK phase	...	high	low	high	low	high	low	...	high	low	...
ADDR[22:20] / ACC[2:0]	...	**addr 0**	**acc 0**	**addr 0**	**000**	**addr 0**	**000**	**...**	addr 1	acc 1	...
ADDR[19:16] / IQSTAT[3:0]	...		iqstat-1		**iqstat 0**		**0000**	**...**		**0000**	...
ADDR[15:0] / IVD[15:0]	...		?		**x**		**x**	**...**		**x**	...
DATA[15:0] (write)	...	?	**data 0**							x	...
R$\overline{W}$	...	0	0	0	0	0	0	...	1	1	...

5.4.2.4.3 Read-Write-Read Access Timing

Table 5-18. Interleaved Read-Write-Read Accesses (1 Cycle)

		Access #0	**Access #1**	**Access #2**	
Bus cycle ->	...	**1**	**2**	**3**	...

Table 5-18. Interleaved Read-Write-Read Accesses (1 Cycle) (continued)

ECLK phase	...	high	low	high	low	high	low	...
ADDR[22:20] / ACC[2:0]	...	**addr 0**	**acc 0**	addr 1	acc 1	addr 2	acc 2	...
ADDR[19:16] / IQSTAT[3:0]	...		iqstat -1		**iqstat 0**		iqstat 1	...
ADDR[15:0] / IVD[15:0]	...		?		**ivd 0**		x	...
DATA[15:0] (internal read)	...	?	z	z	(write) data 1		z	...
DATA[15:0] (external read)	...	?	z	**data 0**	(write) data 1		z	...
R$\overline{\text{W}}$	...	1	1	0	0	1	1	...

5.4.3 Accesses to Port Replacement Registers

All read and write accesses to PRR addresses take two bus clock cycles independent of the operating mode. If writing to these addresses in emulation modes, the access is directed to both, the internal register and the external resource while reads will be treated external.

The XEBI control registers also belong to this category.

5.4.4 Stretched External Bus Accesses

In order to allow fast internal bus cycles to coexist in a system with slower external resources, the XEBI supports stretched external bus accesses (wait states) for each external address range related to one of the 4 chip select lines individually.

This feature is available in normal expanded mode and emulation expanded mode for accesses to all external addresses except emulation memory and PRR. In these cases the fixed access times are 1 or 2 cycles, respectively.

Stretched accesses are controlled by:

1. EXSTR1[2:0] and EXSTR0[2:0] bits in the EBICTL1 register configuring a fixed amount of stretch cycles individually for each $\overline{\text{CSx}}$ line in MMCCTL0
2. Activation of the external wait feature for each $\overline{\text{CSx}}$ line MMCCTL0 register
3. Assertion of the external $\overline{\text{EWAIT}}$ signal when at least one $\overline{\text{CSx}}$ line is configured for EWAIT

The EXSTRx[2:0] control bits can be programmed for generation of a fixed number of 1 to 8 stretch cycles. If the external wait feature is enabled, the minimum number of additional stretch cycles is 2. An arbitrary amount of stretch cycles can be added using the $\overline{\text{EWAIT}}$ input.

$\overline{\text{EWAIT}}$ needs to be asserted at least for a minimal specified time window within an external access cycle for the internal logic to detect it and add a cycle (refer to electrical characteristics). Holding it for additional cycles will cause the external bus access to be stretched accordingly.

Write accesses are stretched by holding the initiator in its current state for additional cycles as programmed and controlled by external wait after the data have been driven out on the external bus. This results in an extension of time the bus signals and the related control signals are valid externally.

Read data are not captured by the system in normal expanded mode until the specified setup time before the $\overline{\text{RE}}$ rising edge.

Read data are not captured in emulation expanded mode until the specified setup time before the falling edge of ECLK.

In emulation expanded mode, accesses to the internal flash or the emulation memory (determined by EROMON and ROMON bits; see S12X_MMC section for details) always take 1 cycle and stretching is not supported. In case the internal flash is taken out of the map in user applications, accesses are stretched as programmed and controlled by external wait.

5.4.5 Data Select and Data Direction Signals

The S12X_EBI supports byte and word accesses at any valid external address. The big endian system of the MCU is extended to the external bus; however, word accesses are restricted to even aligned addresses. The only exception is the visibility of misaligned word accesses to addresses in the internal RAM as this module exclusively supports these kind of accesses in a single cycle.

With the above restriction, a fixed relationship is implied between the address parity and the dedicated bus halves where the data are accessed: DATA[15:8] is related to even addresses and DATA[7:0] is related to odd addresses.

In expanded modes the data access type is externally determined by a set of control signals, i.e., data select and data direction signals, as described below. The data select signals are not available if using the external bus interface with an 8-bit data bus.

5.4.5.1 Normal Expanded Mode

In normal expanded mode, the external signals $\overline{\text{RE}}$, $\overline{\text{WE}}$, $\overline{\text{UDS}}$, $\overline{\text{LDS}}$ indicate the access type (read/write), data size and alignment of an external bus access (Table 5-19).

Table 5-19. Access in Normal Expanded Mode

Access	$\overline{\text{RE}}$	$\overline{\text{WE}}$	$\overline{\text{UDS}}$	$\overline{\text{LDS}}$	DATA[15:8]		DATA[7:0]	
					I/O	data(addr)	I/O	data(addr)
Word write of data on DATA[15:0] at an even and even+1 address	1	0	0	0	Out	data(even)	Out	data(odd)
Byte write of data on DATA[7:0] at an odd address	1	0	1	0	In	x	Out	data(odd)
Byte write of data on DATA[15:8] at an even address	1	0	0	1	Out	data(even)	In	x
Word read of data on DATA[15:0] at an even and even+1 address	0	1	0	0	In	data(even)	In	data(odd)
Byte read of data on DATA[7:0] at an odd address	0	1	1	0	In	x	In	data(odd)
Byte read of data on DATA[15:8] at an even address	0	1	0	1	In	data(even)	In	x
Indicates No Access	1	1	1	1	In	x	In	x
Unimplemented	1	1	1	0	In	x	In	x
	1	1	0	1	In	x	In	x

5.4.5.2 Emulation Modes and Special Test Mode

In emulation modes and special test mode, the external signals $\overline{\text{LSTRB}}$, R$\overline{\text{W}}$, and ADDR0 indicate the access type (read/write), data size and alignment of an external bus access. Misaligned accesses to the internal RAM and misaligned XGATE PRR accesses in emulation modes are the only type of access that are able to produce $\overline{\text{LSTRB}}$ = ADDR0 = 1. This is summarized in Table 5-20.

Table 5-20. Access in Emulation Modes and Special Test Mode

Access	R$\overline{\text{W}}$	$\overline{\text{LSTRB}}$	ADDR0	DATA[15:8]		DATA[7:0]	
				I/O	data(addr)	I/O	data(addr)
Word write of data on DATA[15:0] at an even and even+1 address	0	0	0	Out	data(even)	Out	data(odd)
Byte write of data on DATA[7:0] at an odd address	0	0	1	In	x	Out	data(odd)
Byte write of data on DATA[15:8] at an even address	0	1	0	Out	data(odd)	In	x
Word write at an odd and odd+1 internal RAM address (misaligned — only in emulation modes)	0	1	1	Out	data(odd+1)	Out	data(odd)
Word read of data on DATA[15:0] at an even and even+1 address	1	0	0	In	data(even)	In	data(even+1)
Byte read of data on DATA[7:0] at an odd address	1	0	1	In	x	In	data(odd)
Byte read of data on DATA[15:8] at an even address	1	1	0	In	data(even)	In	x
Word read at an odd and odd+1 internal RAM address (misaligned - only in emulation modes)	1	1	1	In	data(odd+1)	In	data(odd)

5.4.6 Low-Power Options

The XEBI does not support any user-controlled options for reducing power consumption.

5.4.6.1 Run Mode

The XEBI does not support any options for reducing power in run mode.

Power consumption is reduced in single-chip modes due to the absence of the external bus interface. Operation in expanded modes results in a higher power consumption, however any unnecessary toggling of external bus signals is reduced to the lowest indispensable activity by holding the previous states between external accesses.

5.4.6.2 Wait Mode

The XEBI does not support any options for reducing power in wait mode.

5.4.6.3 Stop Mode

The XEBI will cease to function in stop mode.

5.5 Initialization/Application Information

This section describes the external bus interface usage and timing. Typical customer operating modes are normal expanded mode and emulation modes, specifically to be used in emulator applications. Taking the availability of the external wait feature into account the use cases are divided into four scenarios:

- Normal expanded mode
  - External wait feature disabled
  - External wait feature enabled
- Emulation modes
  - Emulation single-chip mode (without wait states)
  - Emulation expanded mode (with optional access stretching)

Normal single-chip mode and special single-chip mode do not have an external bus. Special test mode is used for factory test only. Therefore, these modes are omitted here.

All timing diagrams referred to throughout this section are available in the Electrical Characteristics appendix of the SoC section.

5.5.1 Normal Expanded Mode

This mode allows interfacing to external memories or peripherals which are available in the commercial market. In these applications the normal bus operation requires a minimum of 1 cycle stretch for each external access.

5.5.1.1 Example 1a: External Wait Feature Disabled

The first example of bus timing of an external read and write access with the external wait feature disabled is shown in

- Figure ‘Example 1a: Normal Expanded Mode — Read Followed by Write’

The associated supply voltage dependent timing are numbers given in

- Table ‘Example 1a: Normal Expanded Mode Timing V_{DD5} = 5.0 V (EWAIT disabled)’
- Table ‘Example 1a: Normal Expanded Mode Timing V_{DD5} = 3.0 V (EWAIT disabled)’

Systems designed this way rely on the internal programmable access stretching. These systems have predictable external memory access times. The additional stretch time can be programmed up to 8 cycles to provide longer access times.

5.5.1.2 Example 1b: External Wait Feature Enabled

The external wait operation is shown in this example. It can be used to exceed the amount of stretch cycles over the programmed number in EXSTR[2:0]. The feature must be enabled by configuring at least one $\overline{\text{CSx}}$ line for EWAIT.

If the $\overline{\text{EWAIT}}$ signal is not asserted, the number of stretch cycles is forced to a minimum of 2 cycles. If $\overline{\text{EWAIT}}$ is asserted within the predefined time window during the access it will be strobed active and another stretch cycle is added. If strobed inactive, the next cycle will be the last cycle before the access is finished. $\overline{\text{EWAIT}}$ can be held asserted as long as desired to stretch the access.

An access with 1 cycle stretch by $\overline{\text{EWAIT}}$ assertion is shown in

- Figure ‘Example 1b: Normal Expanded Mode — Stretched Read Access’
- Figure ‘Example 1b: Normal Expanded Mode — Stretched Write Access’

The associated timing numbers for both operations are given in

- Table ‘Example 1b: Normal Expanded Mode Timing V_{DD5} = 5.0 V (EWAIT enabled)’
- Table ‘Example 1b: Normal Expanded Mode Timing V_{DD5} = 3.0 V (EWAIT enabled)’

It is recommended to use the free-running clock (ECLK) at the fastest rate (bus clock rate) to synchronize the $\overline{\text{EWAIT}}$ input signal.

5.5.2 Emulation Modes

In emulation mode applications, the development systems use a custom PRU device to rebuild the single-chip or expanded bus functions which are lost due to the use of the external bus with an emulator.

Accesses to a set of registers controlling the related ports in normal modes (refer to SoC section) are directed to the external bus in emulation modes which are substituted by PRR as part of the PRU. Accesses to these registers take a constant time of 2 cycles.

Depending on the setting of ROMON and EROMON (refer to S12X_MMC section), the program code can be executed from internal memory or an optional external emulation memory (EMULMEM). No wait state operation (stretching) of the external bus access is done in emulation modes when accessing internal memory or emulation memory addresses.

In both modes observation of the internal operation is supported through the external bus (internal visibility).

5.5.2.1 Example 2a: Emulation Single-Chip Mode

This mode is used for emulation systems in which the target application is operating in normal single-chip mode.

Figure 5-5 shows the PRU connection with the available external bus signals in an emulator application.

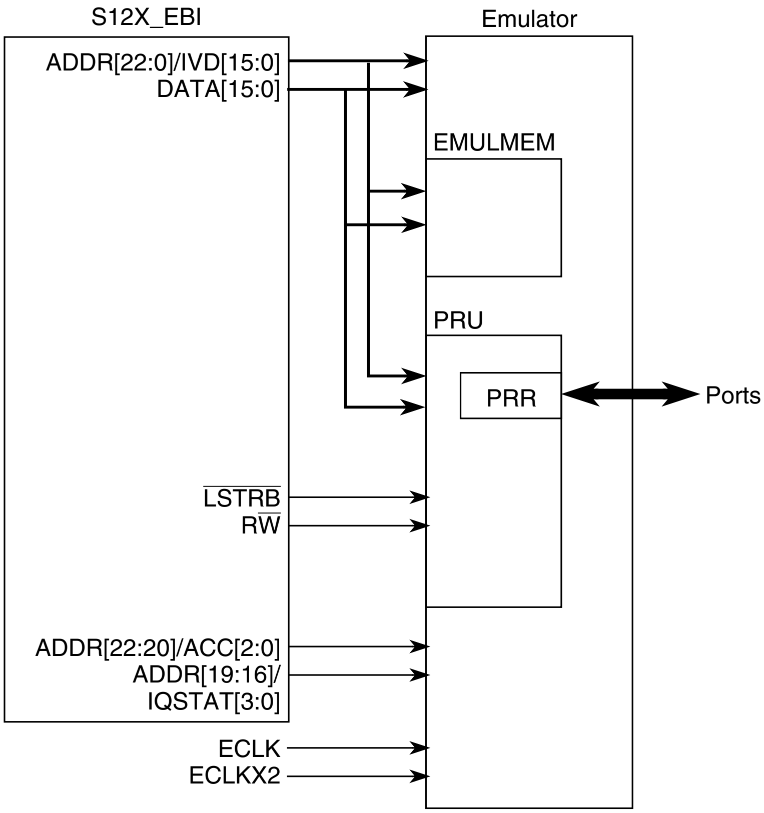


Figure 5-5. Application in Emulation Single-Chip Mode

The timing diagram for this operation is shown in:

- Figure ‘Example 2a: Emulation Single-Chip Mode — Read Followed by Write’

The associated timing numbers are given in:

- Table ‘Example 2a: Emulation Single-Chip Mode Timing (EWAIT disabled)’

Timing considerations:

- Signals muxed with address lines ADDRx, i.e., IVDx, IQSTATx and ACCx, have the same timing.
- $\overline{\text{LSTRB}}$ has the same timing as R$\overline{\text{W}}$.

- ECLKX2 rising edges have the same timing as ECLK edges.
- The timing for accesses to PRU registers, which take 2 cycles to complete, is the same as the timing for an external non-PRR access with 1 cycle of stretch as shown in example 2b.

5.5.2.2 Example 2b: Emulation Expanded Mode

This mode is used for emulation systems in which the target application is operating in normal expanded mode.

If the external bus is used with a PRU, the external device rebuilds the data select and data direction signals $\overline{\text{UDS}}$, $\overline{\text{LDS}}$, $\overline{\text{RE}}$, and $\overline{\text{WE}}$ from the ADDR0, $\overline{\text{LSTRB}}$, and R$\overline{\text{W}}$ signals.

Figure 5-6 shows the PRU connection with the available external bus signals in an emulator application.

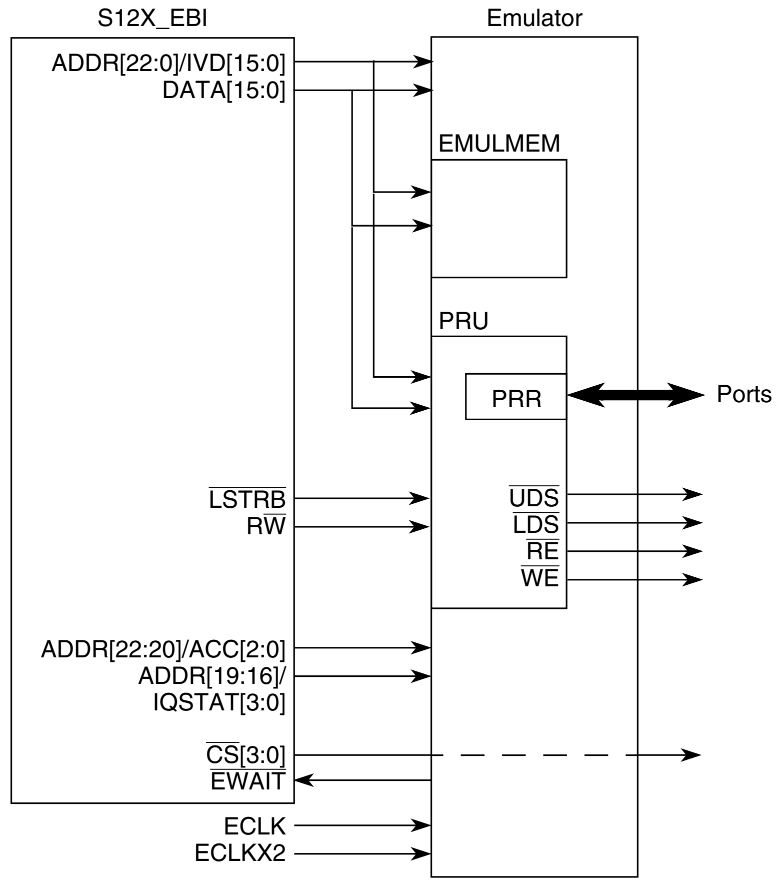


Figure 5-6. Application in Emulation Expanded Mode

The timings of accesses with 1 stretch cycle are shown in

- Figure ‘Example 2b: Emulation Expanded Mode — Read with 1 Stretch Cycle’
- Figure ‘Example 2b: Emulation Expanded Mode — Write with 1 Stretch Cycle’

The associated timing numbers are given in

- Table ‘Example 2b: Emulation Expanded Mode Timing $V_{DD5} = 5.0$ V (EWAIT disabled)’ (this also includes examples for alternative settings of 2 and 3 additional stretch cycles)

Timing considerations:

- If no stretch cycle is added, the timing is the same as in Emulation Single-Chip Mode.

Chapter 6
Interrupt (S12XINTV2)

Table 6-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V02.00	01 Jul 2005	6.1.2/6-262	Initial V2 release, added new features: - XGATE threads can be interrupted. - SYS instruction vector. - Access violation interrupt vectors.
V02.04	11 Jan 2007	6.3.2.2/6-267 6.3.2.4/6-268	- Added Notes for devices without XGATE module.
V02.05	20 Mar 2007	6.4.6/6-274	- Fixed priority definition for software exceptions.
V02.07	13 Dec 2011	6.5.3.1/6-276	- Re-worded for difference of Wake-up feature between STOP and WAIT modes.

6.1 Introduction

The XINT module decodes the priority of all system exception requests and provides the applicable vector for processing the exception to either the CPU or the XGATE module. The XINT module supports:

- I bit and X bit maskable interrupt requests
- One non-maskable unimplemented op-code trap
- One non-maskable software interrupt (SWI) or background debug mode request
- One non-maskable system call interrupt (SYS)
- Three non-maskable access violation interrupts
- One spurious interrupt vector request
- Three system reset vector requests

Each of the I bit maskable interrupt requests can be assigned to one of seven priority levels supporting a flexible priority scheme. For interrupt requests that are configured to be handled by the CPU, the priority scheme can be used to implement nested interrupt capability where interrupts from a lower level are automatically blocked if a higher level interrupt is being processed. Interrupt requests configured to be handled by the XGATE module can be nested one level deep.

NOTE

The HPRIO register and functionality of the original S12 interrupt module is no longer supported. It is superseded by the 7-level interrupt request priority scheme.

6.1.1 Glossary

The following terms and abbreviations are used in the document.

Table 6-2. Terminology

Term	Meaning
CCR	Condition Code Register (in the S12X CPU)
DMA	Direct Memory Access
INT	Interrupt
IPL	Interrupt Processing Level
ISR	Interrupt Service Routine
MCU	Micro-Controller Unit
XGATE	refers to the XGATE co-processor; XGATE is an optional feature
$\overline{\text{IRQ}}$	refers to the interrupt request associated with the $\overline{\text{IRQ}}$ pin
$\overline{\text{XIRQ}}$	refers to the interrupt request associated with the $\overline{\text{XIRQ}}$ pin

6.1.2 Features

- Interrupt vector base register (IVBR)
- One spurious interrupt vector (at address vector base[1] + 0x0010).
- One non-maskable system call interrupt vector request (at address vector base + 0x0012).
- Three non-maskable access violation interrupt vector requests (at address vector base + 0x0014–0x0018).
- 2–109 I bit maskable interrupt vector requests (at addresses vector base + 0x001A–0x00F2).
- Each I bit maskable interrupt request has a configurable priority level and can be configured to be handled by either the CPU or the XGATE module[2].
- I bit maskable interrupts can be nested, depending on their priority levels.
- One X bit maskable interrupt vector request (at address vector base + 0x00F4).
- One non-maskable software interrupt request (SWI) or background debug mode vector request (at address vector base + 0x00F6).
- One non-maskable unimplemented op-code trap (TRAP) vector (at address vector base + 0x00F8).
- Three system reset vectors (at addresses 0xFFFA–0xFFFE).
- Determines the highest priority XGATE and interrupt vector requests, drives the vector to the XGATE module or to the bus on CPU request, respectively.
- Wakes up the system from stop or wait mode when an appropriate interrupt request occurs or whenever $\overline{\text{XIRQ}}$ is asserted, even if X interrupt is masked.
- XGATE can wake up and execute code, even with the CPU remaining in stop or wait mode.

[1] The vector base is a 16-bit address which is accumulated from the contents of the interrupt vector base register (IVBR, used as upper byte) and 0x00 (used as lower byte).

[2] The $\overline{\text{IRQ}}$ interrupt can only be handled by the CPU

6.1.3 Modes of Operation

- Run mode

  This is the basic mode of operation.
- Wait mode

  In wait mode, the XINT module is frozen. It is however capable of either waking up the CPU if an interrupt occurs or waking up the XGATE if an XGATE request occurs. Please refer to Section 6.5.3, “Wake Up from Stop or Wait Mode” for details.
- Stop Mode

  In stop mode, the XINT module is frozen. It is however capable of either waking up the CPU if an interrupt occurs or waking up the XGATE if an XGATE request occurs. Please refer to Section 6.5.3, “Wake Up from Stop or Wait Mode” for details.
- Freeze mode (BDM active)

  In freeze mode (BDM active), the interrupt vector base register is overridden internally. Please refer to Section 6.3.2.1, “Interrupt Vector Base Register (IVBR)” for details.

6.1.4 Block Diagram

Figure 6-1 shows a block diagram of the XINT module.

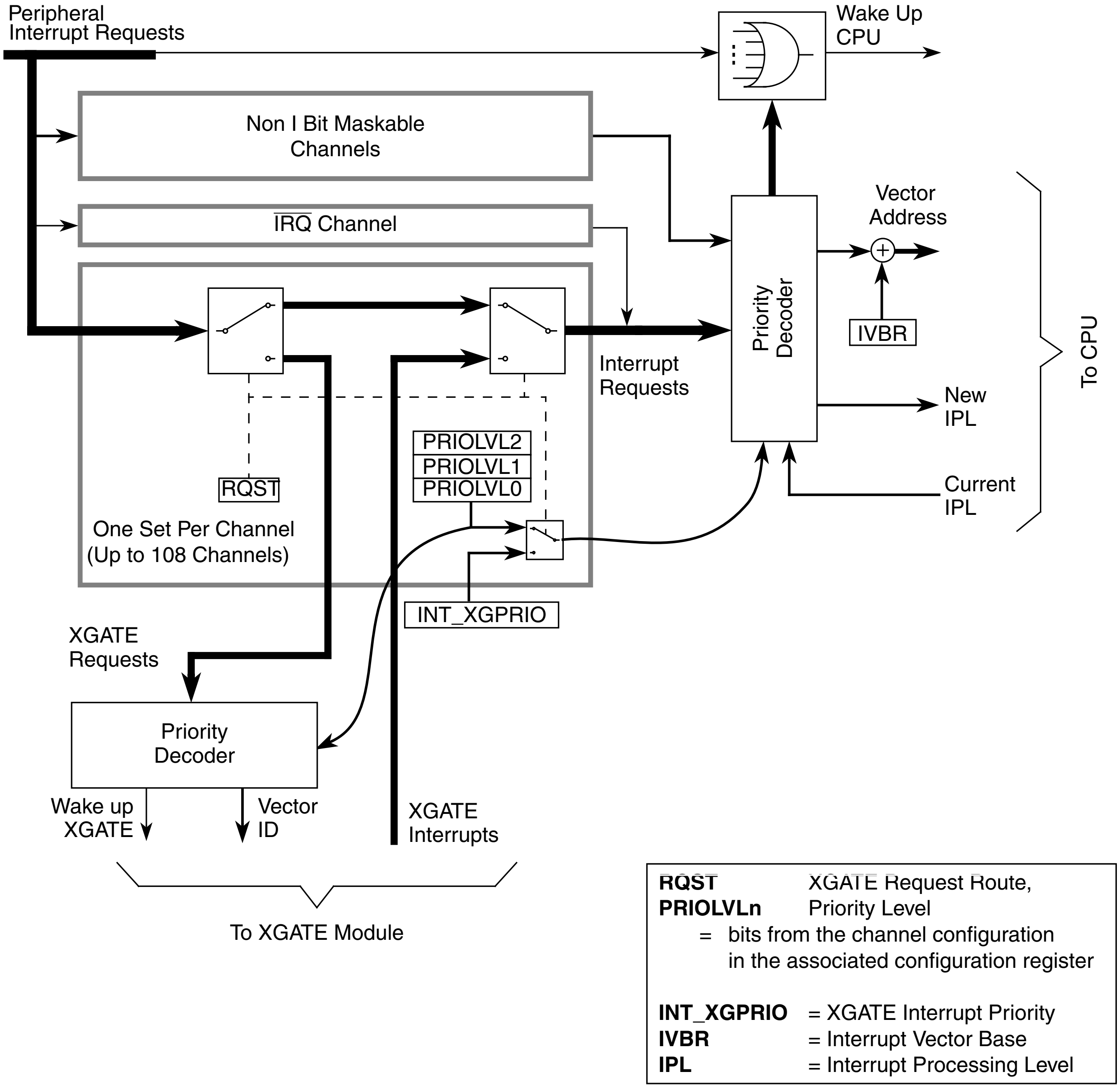


Figure 6-1. XINT Block Diagram

6.2 External Signal Description

The XINT module has no external signals.

6.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the XINT module.

6.3.1 Module Memory Map

Table 6-3 gives an overview over all XINT module registers.

Table 6-3. XINT Memory Map

Address	Use	Access
0x0120	RESERVED	—
0x0121	Interrupt Vector Base Register (IVBR)	R/W
0x0122–0x0125	RESERVED	—
0x0126	XGATE Interrupt Priority Configuration Register (INT_XGPRIO)	R/W
0x0127	Interrupt Request Configuration Address Register (INT_CFADDR)	R/W
0x0128	Interrupt Request Configuration Data Register 0 (INT_CFDATA0)	R/W
0x0129	Interrupt Request Configuration Data Register 1 (INT_CFDATA1)	R/W
0x012A	Interrupt Request Configuration Data Register 2 (INT_CFDATA2	R/W
0x012B	Interrupt Request Configuration Data Register 3 (INT_CFDATA3)	R/W
0x012C	Interrupt Request Configuration Data Register 4 (INT_CFDATA4)	R/W
0x012D	Interrupt Request Configuration Data Register 5 (INT_CFDATA5)	R/W
0x012E	Interrupt Request Configuration Data Register 6 (INT_CFDATA6)	R/W
0x012F	Interrupt Request Configuration Data Register 7 (INT_CFDATA7)	R/W

6.3.2 Register Descriptions

This section describes in address order all the XINT module registers and their individual bits.

Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0121	IVBR	R	IVB_ADDR[7:0]7							
		W								
0x0126	INT_XGPRIO	R	0	0	0	0	0	XILVL[2:0]		
		W	▨	▨	▨	▨	▨			
0x0127	INT_CFADDR	R	INT_CFADDR[7:4]				0	0	0	0
		W					▨	▨	▨	▨
0x0128	INT_CFDATA0	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W		▨	▨	▨	▨			
0x0129	INT_CFDATA1	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W		▨	▨	▨	▨			
0x012A	INT_CFDATA2	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W		▨	▨	▨	▨			
0x012B	INT_CFDATA3	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W		▨	▨	▨	▨			
0x012C	INT_CFDATA4	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W		▨	▨	▨	▨			
0x012D	INT_CFDATA5	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W		▨	▨	▨	▨			
0x012E	INT_CFDATA6	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W		▨	▨	▨	▨			
0x012F	INT_CFDATA7	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W		▨	▨	▨	▨			

▨ = Unimplemented or Reserved

Figure 6-2. XINT Register Summary

6.3.2.1 Interrupt Vector Base Register (IVBR)

Address: 0x0121

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

Figure 6-3. Interrupt Vector Base Register (IVBR)

Read: Anytime

Write: Anytime

Table 6-4. IVBR Field Descriptions

Field	Description
7–0 IVB_ADDR[7:0]	**Interrupt Vector Base Address Bits** — These bits represent the upper byte of all vector addresses. Out of reset these bits are set to 0xFF (i.e., vectors are located at 0xFF10–0xFFFE) to ensure compatibility to previous S12 microcontrollers. **Note:** A system reset will initialize the interrupt vector base register with “0xFF” before it is used to determine the reset vector address. Therefore, changing the IVBR has no effect on the location of the three reset vectors (0xFFFA–0xFFFE). **Note:** If the BDM is active (i.e., the CPU is in the process of executing BDM firmware code), the contents of IVBR are ignored and the upper byte of the vector address is fixed as “0xFF”.

6.3.2.2 XGATE Interrupt Priority Configuration Register (INT_XGPRIO)

Address: 0x0126

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	XILVL[2:0]		
W	▨	▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1

▨ = Unimplemented or Reserved

Figure 6-4. XGATE Interrupt Priority Configuration Register (INT_XGPRIO)

Read: Anytime

Write: Anytime

Table 6-5. INT_XGPRIO Field Descriptions

Field	Description
2–0 XILVL[2:0]	**XGATE Interrupt Priority Level** — The XILVL[2:0] bits configure the shared interrupt level of the XGATE interrupts coming from the XGATE module. Out of reset the priority is set to the lowest active level (“1”). **Note:** If the XGATE module is not available on the device, write accesses to this register are ignored and read accesses to this register will return all 0.

Table 6-6. XGATE Interrupt Priority Levels

Priority	XILVL2	XILVL1	XILVL0	Meaning
	0	0	0	Interrupt request is disabled
low	0	0	1	Priority level 1
	0	1	0	Priority level 2
	0	1	1	Priority level 3
	1	0	0	Priority level 4
	1	0	1	Priority level 5
	1	1	0	Priority level 6
high	1	1	1	Priority level 7

6.3.2.3 Interrupt Request Configuration Address Register (INT_CFADDR)

Address: 0x0127

	7	6	5	4	3	2	1	0
R	INT_CFADDR[7:4]				0	0	0	0
W					▨	▨	▨	▨
Reset	0	0	0	1	0	0	0	0

▨ = Unimplemented or Reserved

Figure 6-5. Interrupt Configuration Address Register (INT_CFADDR)

Read: Anytime

Write: Anytime

Table 6-7. INT_CFADDR Field Descriptions

Field	Description
7–4 INT_CFADDR[7:4]	**Interrupt Request Configuration Data Register Select Bits** — These bits determine which of the 128 configuration data registers are accessible in the 8 register window at INT_CFDATA0–7. The hexadecimal value written to this register corresponds to the upper nibble of the lower byte of the address of the interrupt vector, i.e., writing 0xE0 to this register selects the configuration data register block for the 8 interrupt vector requests starting with vector at address (vector base + 0x00E0) to be accessible as INT_CFDATA0–7. **Note:** Writing all 0s selects non-existing configuration registers. In this case write accesses to INT_CFDATA0–7 will be ignored and read accesses will return all 0.

6.3.2.4 Interrupt Request Configuration Data Registers (INT_CFDATA0–7)

The eight register window visible at addresses INT_CFDATA0–7 contains the configuration data for the block of eight interrupt requests (out of 128) selected by the interrupt configuration address register (INT_CFADDR) in ascending order. INT_CFDATA0 represents the interrupt configuration data register of the vector with the lowest address in this block, while INT_CFDATA7 represents the interrupt configuration data register of the vector with the highest address, respectively.

Address: 0x0128

	7	6	5	4	3	2	1	0
R	RQST	0	0	0	0	PRIOLVL[2:0]		
W		▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1[1]

▨ = Unimplemented or Reserved

Figure 6-6. Interrupt Request Configuration Data Register 0 (INT_CFDATA0)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x0129

	7	6	5	4	3	2	1	0
R	RQST	0	0	0	0	PRIOLVL[2:0]		
W		▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1[1]

▨ = Unimplemented or Reserved

Figure 6-7. Interrupt Request Configuration Data Register 1 (INT_CFDATA1)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012A

	7	6	5	4	3	2	1	0
R	RQST	0	0	0	0	PRIOLVL[2:0]		
W		▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1[1]

▨ = Unimplemented or Reserved

Figure 6-8. Interrupt Request Configuration Data Register 2 (INT_CFDATA2)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012B

	7	6	5	4	3	2	1	0
R	RQST	0	0	0	0	PRIOLVL[2:0]		
W		▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1[1]

▨ = Unimplemented or Reserved

Figure 6-9. Interrupt Request Configuration Data Register 3 (INT_CFDATA3)

[1] Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012C

	7	6	5	4	3	2	1	0
R	RQST	0	0	0	0	PRIOLVL[2:0]		
W		▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1[1]

▨ = Unimplemented or Reserved

Figure 6-10. Interrupt Request Configuration Data Register 4 (INT_CFDATA4)

1. Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012D

	7	6	5	4	3	2	1	0
R	RQST	0	0	0	0	PRIOLVL[2:0]		
W		▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1[1]

▨ = Unimplemented or Reserved

Figure 6-11. Interrupt Request Configuration Data Register 5 (INT_CFDATA5)

1. Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012E

	7	6	5	4	3	2	1	0
R	RQST	0	0	0	0	PRIOLVL[2:0]		
W		▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1[1]

▨ = Unimplemented or Reserved

Figure 6-12. Interrupt Request Configuration Data Register 6 (INT_CFDATA6)

1. Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012F

	7	6	5	4	3	2	1	0
R	RQST	0	0	0	0	PRIOLVL[2:0]		
W		▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	1[1]

▨ = Unimplemented or Reserved

Figure 6-13. Interrupt Request Configuration Data Register 7 (INT_CFDATA7)

1. Please refer to the notes following the PRIOLVL[2:0] description below.

Read: Anytime

Write: Anytime

Table 6-8. INT_CFDATA0–7 Field Descriptions

Field	Description
7 RQST	**XGATE Request Enable** — This bit determines if the associated interrupt request is handled by the CPU or by the XGATE module. 0 Interrupt request is handled by the CPU 1 Interrupt request is handled by the XGATE module **Note:** The $\overline{\text{IRQ}}$ interrupt cannot be handled by the XGATE module. For this reason, the configuration register for vector (vector base + 0x00F2) = $\overline{\text{IRQ}}$ vector address) does not contain a RQST bit. Writing a 1 to the location of the RQST bit in this register will be ignored and a read access will return 0. **Note:** If the XGATE module is not available on the device, writing a 1 to the location of the RQST bit in this register will be ignored and a read access will return 0.
2–0 PRIOLVL[2:0]	**Interrupt Request Priority Level Bits** — The PRIOLVL[2:0] bits configure the interrupt request priority level of the associated interrupt request. Out of reset all interrupt requests are enabled at the lowest active level (“1”) to provide backwards compatibility with previous S12 interrupt controllers. Please also refer to Table 6-9 for available interrupt request priority levels. **Note:** Write accesses to configuration data registers of unused interrupt channels will be ignored and read accesses will return all 0. For information about what interrupt channels are used in a specific MCU, please refer to the Device Reference Manual of that MCU. **Note:** When vectors (vector base + 0x00F0–0x00FE) are selected by writing 0xF0 to INT_CFADDR, writes to INT_CFDATA2–7 (0x00F4–0x00FE) will be ignored and read accesses will return all 0s. The corresponding vectors do not have configuration data registers associated with them. **Note:** When vectors (vector base + 0x0010–0x001E) are selected by writing 0x10 to INT_CFADDR, writes to INT_CFDATA1–INT_CFDATA4 (0x0012–0x0018) will be ignored and read accesses will return all 0s. The corresponding vectors do not have configuration data registers associated with them. **Note:** Write accesses to the configuration register for the spurious interrupt vector request (vector base + 0x0010) will be ignored and read accesses will return 0x07 (request is handled by the CPU, PRIOLVL = 7).

Table 6-9. Interrupt Priority Levels

Priority	PRIOLVL2	PRIOLVL1	PRIOLVL0	Meaning
	0	0	0	Interrupt request is disabled
low	0	0	1	Priority level 1
	0	1	0	Priority level 2
	0	1	1	Priority level 3
	1	0	0	Priority level 4
	1	0	1	Priority level 5
	1	1	0	Priority level 6
high	1	1	1	Priority level 7

6.4 Functional Description

The XINT module processes all exception requests to be serviced by the CPU module. These exceptions include interrupt vector requests and reset vector requests. Each of these exception types and their overall priority level is discussed in the subsections below.

6.4.1 S12X Exception Requests

The CPU handles both reset requests and interrupt requests. The XINT module contains registers to configure the priority level of each I bit maskable interrupt request which can be used to implement an interrupt priority scheme. This also includes the possibility to nest interrupt requests. A priority decoder is used to evaluate the priority of a pending interrupt request.

6.4.2 Interrupt Prioritization

After system reset all interrupt requests with a vector address lower than or equal to (vector base + 0x00F2) are enabled, are set up to be handled by the CPU and have a pre-configured priority level of 1. Exceptions to this rule are the non-maskable interrupt requests and the spurious interrupt vector request at (vector base + 0x0010) which cannot be disabled, are always handled by the CPU and have a fixed priority levels. A priority level of 0 effectively disables the associated I bit maskable interrupt request.

If more than one interrupt request is configured to the same interrupt priority level the interrupt request with the higher vector address wins the prioritization.

The following conditions must be met for an I bit maskable interrupt request to be processed.

1. The local interrupt enabled bit in the peripheral module must be set.
2. The setup in the configuration register associated with the interrupt request channel must meet the following conditions:
   a) The XGATE request enable bit must be 0 to have the CPU handle the interrupt request.
   b) The priority level must be set to non zero.
   c) The priority level must be greater than the current interrupt processing level in the condition code register (CCR) of the CPU (PRIOLVL[2:0] > IPL[2:0]).
3. The I bit in the condition code register (CCR) of the CPU must be cleared.
4. There is no access violation interrupt request pending.
5. There is no SYS, SWI, BDM, TRAP, or $\overline{\text{XIRQ}}$ request pending.

NOTE

All non I bit maskable interrupt requests always have higher priority than I bit maskable interrupt requests. If an I bit maskable interrupt request is interrupted by a non I bit maskable interrupt request, the currently active interrupt processing level (IPL) remains unaffected. It is possible to nest non I bit maskable interrupt requests, e.g., by nesting SWI or TRAP calls.

6.4.2.1 Interrupt Priority Stack

The current interrupt processing level (IPL) is stored in the condition code register (CCR) of the CPU. This way the current IPL is automatically pushed to the stack by the standard interrupt stacking procedure. The new IPL is copied to the CCR from the priority level of the highest priority active interrupt request channel which is configured to be handled by the CPU. The copying takes place when the interrupt vector is fetched. The previous IPL is automatically restored by executing the RTI instruction.

6.4.3 XGATE Requests

If the XGATE module is implemented on the device, the XINT module is also used to process all exception requests to be serviced by the XGATE module. The overall priority level of those exceptions is discussed in the subsections below.

6.4.3.1 XGATE Request Prioritization

An interrupt request channel is configured to be handled by the XGATE module, if the RQST bit of the associated configuration register is set to 1 (please refer to Section 6.3.2.4, "Interrupt Request Configuration Data Registers (INT_CFDATA0–7)"). The priority level configuration (PRIOLVL) for this channel becomes the XGATE priority which will be used to determine the highest priority XGATE request to be serviced next by the XGATE module. Additionally, XGATE interrupts may be raised by the XGATE module by setting one or more of the XGATE channel interrupt flags (by using the SIF instruction). This will result in an CPU interrupt with vector address vector base + (2 * channel ID number), where the channel ID number corresponds to the highest set channel interrupt flag, if the XGIE and channel RQST bits are set.

The shared interrupt priority for the XGATE interrupt requests is taken from the XGATE interrupt priority configuration register (please refer to Section 6.3.2.2, "XGATE Interrupt Priority Configuration Register (INT_XGPRIO)"). If more than one XGATE interrupt request channel becomes active at the same time, the channel with the highest vector address wins the prioritization.

6.4.4 Priority Decoders

The XINT module contains priority decoders to determine the priority for all interrupt requests pending for the respective target.

There are two priority decoders, one for each interrupt request target, CPU or XGATE. The function of both priority decoders is basically the same with one exception: the priority decoder for the XGATE module does not take the current XGATE thread processing level into account. Instead, XGATE requests are handed to the XGATE module including a 1-bit priority identifier. The XGATE module uses this additional information to decide if the new request can interrupt a currently running thread. The 1-bit priority identifier corresponds to the most significant bit of the priority level configuration of the requesting channel. This means that XGATE requests with priority levels 4, 5, 6 or 7 can interrupt running XGATE threads with priority levels 1, 2 and 3.

A CPU interrupt vector is not supplied until the CPU requests it. Therefore, it is possible that a higher priority interrupt request could override the original exception which caused the CPU to request the vector. In this case, the CPU will receive the highest priority vector and the system will process this exception instead of the original request.

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

NOTE

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

6.4.5 Reset Exception Requests

The XINT module supports three system reset exception request types (for details please refer to the Clock and Reset Generator module (CRG)):

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

6.4.6 Exception Priority

The priority (from highest to lowest) and address of all exception vectors issued by the XINT module upon request by the CPU is shown in Table 6-10. Generally, all non-maskable interrupts have higher priorities than maskable interrupts. Please note that between the three software interrupts (Unimplemented op-code trap request, SWI/BGND request, SYS request) there is no real priority defined because they cannot occur simultaneously (the S12XCPU executes one instruction at a time).

Table 6-10. Exception Vector Map and Priority

Vector Address[1]	Source
0xFFFE	Pin reset, power-on reset, low-voltage reset, illegal address reset
0xFFFC	Clock monitor reset
0xFFFA	COP watchdog reset
(Vector base + 0x00F8)	Unimplemented op-code trap
(Vector base + 0x00F6)	Software interrupt instruction (SWI) or BDM vector request
(Vector base + 0x0012)	System call interrupt instruction (SYS)
(Vector base + 0x0018)	(reserved for future use)
(Vector base + 0x0016)	XGATE Access violation interrupt request[2]
(Vector base + 0x0014)	CPU Access violation interrupt request[3]
(Vector base + 0x00F4)	$\overline{\text{XIRQ}}$ interrupt request
(Vector base + 0x00F2)	$\overline{\text{IRQ}}$ interrupt request
(Vector base + 0x00F0–0x001A)	Device specific I bit maskable interrupt sources (priority determined by the associated configuration registers, in descending order)
(Vector base + 0x0010)	Spurious interrupt

[1] 16 bits vector address based

[2] only implemented if device features both a Memory Protection Unit (MPU) and an XGATE co-processor

[3] only implemented if device features a Memory Protection Unit (MPU)

6.5 Initialization/Application Information

6.5.1 Initialization

After system reset, software should:

- Initialize the interrupt vector base register if the interrupt vector table is not located at the default location (0xFF10–0xFFF9).
- Initialize the interrupt processing level configuration data registers (INT_CFADDR, INT_CFDATA0–7) for all interrupt vector requests with the desired priority levels and the request target (CPU or XGATE module). It might be a good idea to disable unused interrupt requests.
- If the XGATE module is used, setup the XGATE interrupt priority register (INT_XGPRIO) and configure the XGATE module (please refer the XGATE Block Guide for details).
- Enable I maskable interrupts by clearing the I bit in the CCR.
- Enable the X maskable interrupt by clearing the X bit in the CCR (if required).

6.5.2 Interrupt Nesting

The interrupt request priority level scheme makes it possible to implement priority based interrupt request nesting for the I bit maskable interrupt requests handled by the CPU.

- I bit maskable interrupt requests can be interrupted by an interrupt request with a higher priority, so that there can be up to seven nested I bit maskable interrupt requests at a time (refer to Figure 6-14 for an example using up to three nested interrupt requests).

I bit maskable interrupt requests cannot be interrupted by other I bit maskable interrupt requests per default. In order to make an interrupt service routine (ISR) interruptible, the ISR must explicitly clear the I bit in the CCR (CLI). After clearing the I bit, I bit maskable interrupt requests with higher priority can interrupt the current ISR.

An ISR of an interruptible I bit maskable interrupt request could basically look like this:

- Service interrupt, e.g., clear interrupt flags, copy data, etc.
- Clear I bit in the CCR by executing the instruction CLI (thus allowing interrupt requests with higher priority)
- Process data
- Return from interrupt by executing the instruction RTI

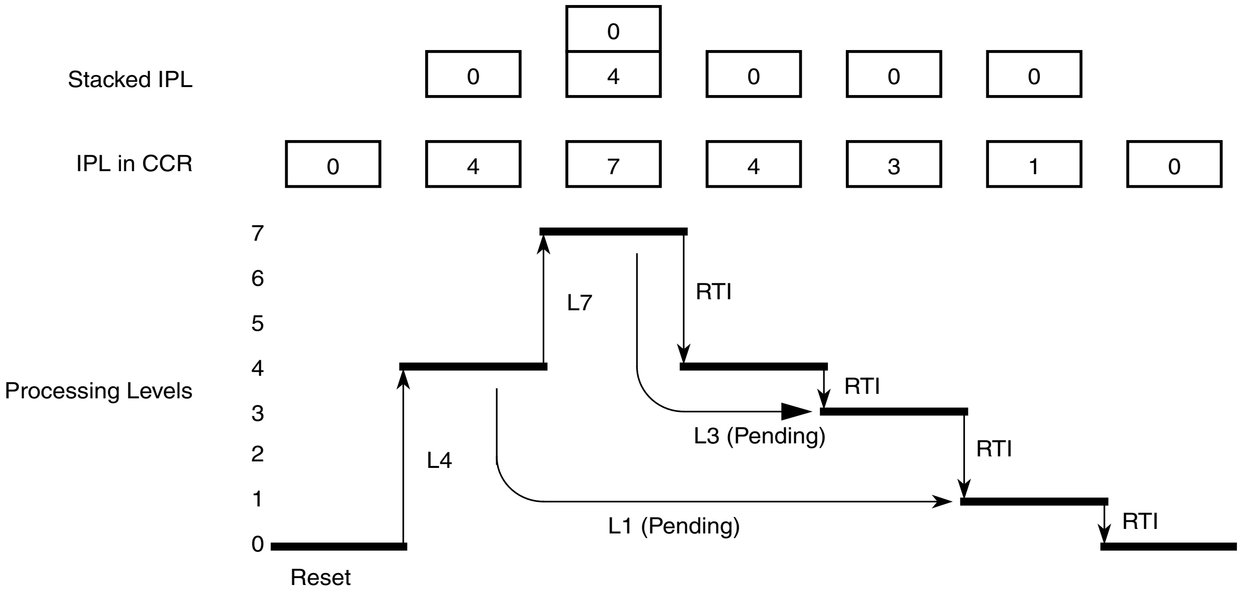


Figure 6-14. Interrupt Processing Example

6.5.3 Wake Up from Stop or Wait Mode

6.5.3.1 CPU Wake Up from Stop or Wait Mode

Only I bit maskable interrupt requests which are configured to be handled by the CPU are capable of waking the MCU from wait mode.

Since bus and core clocks are disabled in stop mode, only interrupt requests that can be generated without these clocks can wake the MCU from stop mode. These are listed in the device overview interrupt vector table. Only I bit maskable interrupt requests which are configured to be handled by the CPU are capable of waking the MCU from stop mode.

To determine whether an I bit maskable interrupt is qualified to wake up the CPU or not, the same settings as in normal run mode are applied during stop or wait mode:

- If the I bit in the CCR is set, all I bit maskable interrupts are masked from waking up the MCU.
- An I bit maskable interrupt is ignored if it is configured to a priority level below or equal to the current IPL in CCR.
- I bit maskable interrupt requests which are configured to be handled by the XGATE module are not capable of waking up the CPU.

The X bit maskable interrupt request can wake up the MCU from stop or wait mode at anytime, even if the X bit in CCR is set. If the X bit maskable interrupt request is used to wake-up the MCU with the X bit in the CCR set, the associated ISR is not called. The CPU then resumes program execution with the instruction following the WAI or STOP instruction. This features works following the same rules like any interrupt request, i.e. care must be taken that the X interrupt request used for wake-up remains active at least until the system begins execution of the instruction following the WAI or STOP instruction; otherwise, wake-up may not occur.

6.5.3.2 XGATE Wake Up from Stop or Wait Mode

Interrupt request channels which are configured to be handled by the XGATE module are capable of waking up the XGATE module. Interrupt request channels handled by the XGATE module do not affect the state of the CPU.

Chapter 7
Background Debug Module (S12XBDMV2)

Table 7-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V02.00	07 Mar 2006		- First version of S12XBDMV2
V02.01	14 May 2008		- Introduced standardized Revision History Table
V02.02	12 Sep 2012		- Minor formatting corrections

7.1 Introduction

This section describes the functionality of the background debug module (BDM) sub-block of the HCS12X core platform.

The background debug module (BDM) sub-block is a single-wire, background debug system implemented in on-chip hardware for minimal CPU intervention. All interfacing with the BDM is done via the BKGD pin.

The BDM has enhanced capability for maintaining synchronization between the target and host while allowing more flexibility in clock rates. This includes a sync signal to determine the communication rate and a handshake signal to indicate when an operation is complete. The system is backwards compatible to the BDM of the S12 family with the following exceptions:

- TAGGO command no longer supported by BDM
- External instruction tagging feature now part of DBG module
- BDM register map and register content extended/modified
- Global page access functionality
- Enabled but not active out of reset in emulation modes (if modes available)
- CLKSW bit set out of reset in emulation modes (if modes available).
- Family ID readable from firmware ROM at global address 0x7FFF0F (value for HCS12X devices is 0xC1)

7.1.1 Features

The BDM includes these distinctive features:

- Single-wire communication with host development system
- Enhanced capability for allowing more flexibility in clock rates
- SYNC command to determine communication rate
- GO_UNTIL command

- Hardware handshake protocol to increase the performance of the serial communication
- Active out of reset in special single chip mode
- Nine hardware commands using free cycles, if available, for minimal CPU intervention
- Hardware commands not requiring active BDM
- 14 firmware commands execute from the standard BDM firmware lookup table
- Software control of BDM operation during wait mode
- Software selectable clocks
- Global page access functionality
- Enabled but not active out of reset in emulation modes (if modes available)
- CLKSW bit set out of reset in emulation modes (if modes available).
- When secured, hardware commands are allowed to access the register space in special single chip mode, if the non-volatile memory erase test fail.
- Family ID readable from firmware ROM at global address 0x7FFF0F (value for HCS12X devices is 0xC1)
- BDM hardware commands are operational until system stop mode is entered (all bus masters are in stop mode)

7.1.2 Modes of Operation

BDM is available in all operating modes but must be enabled before firmware commands are executed. Some systems may have a control bit that allows suspending thefunction during background debug mode.

7.1.2.1 Regular Run Modes

All of these operations refer to the part in run mode and not being secured. The BDM does not provide controls to conserve power during run mode.

- Normal modes

  General operation of the BDM is available and operates the same in all normal modes.
- Special single chip mode

  In special single chip mode, background operation is enabled and active out of reset. This allows programming a system with blank memory.
- Emulation modes (if modes available)

  In emulation mode, background operation is enabled but not active out of reset. This allows debugging and programming a system in this mode more easily.

7.1.2.2 Secure Mode Operation

If the device is in secure mode, the operation of the BDM is reduced to a small subset of its regular run mode operation. Secure operation prevents BDM and CPU accesses to non-volatile memory (Flash and/or EEPROM) other than allowing erasure. For more information please see Section 7.4.1, “Security”.

7.1.2.3 Low-Power Modes

The BDM can be used until all bus masters (e.g., CPU or XGATE or others depending on which masters are available on the SOC) are in stop mode. When CPU is in a low power mode (wait or stop mode) all BDM firmware commands as well as the hardware BACKGROUND command can not be used respectively are ignored. In this case the CPU can not enter BDM active mode, and only hardware read and write commands are available. Also the CPU can not enter a low power mode during BDM active mode.

If all bus masters are in stop mode, the BDM clocks are stopped as well. When BDM clocks are disabled and one of the bus masters exits from stop mode the BDM clocks will restart and BDM will have a soft reset (clearing the instruction register, any command in progress and disable the ACK function). The BDM is now ready to receive a new command.

7.1.3 Block Diagram

A block diagram of the BDM is shown in Figure 7-1.

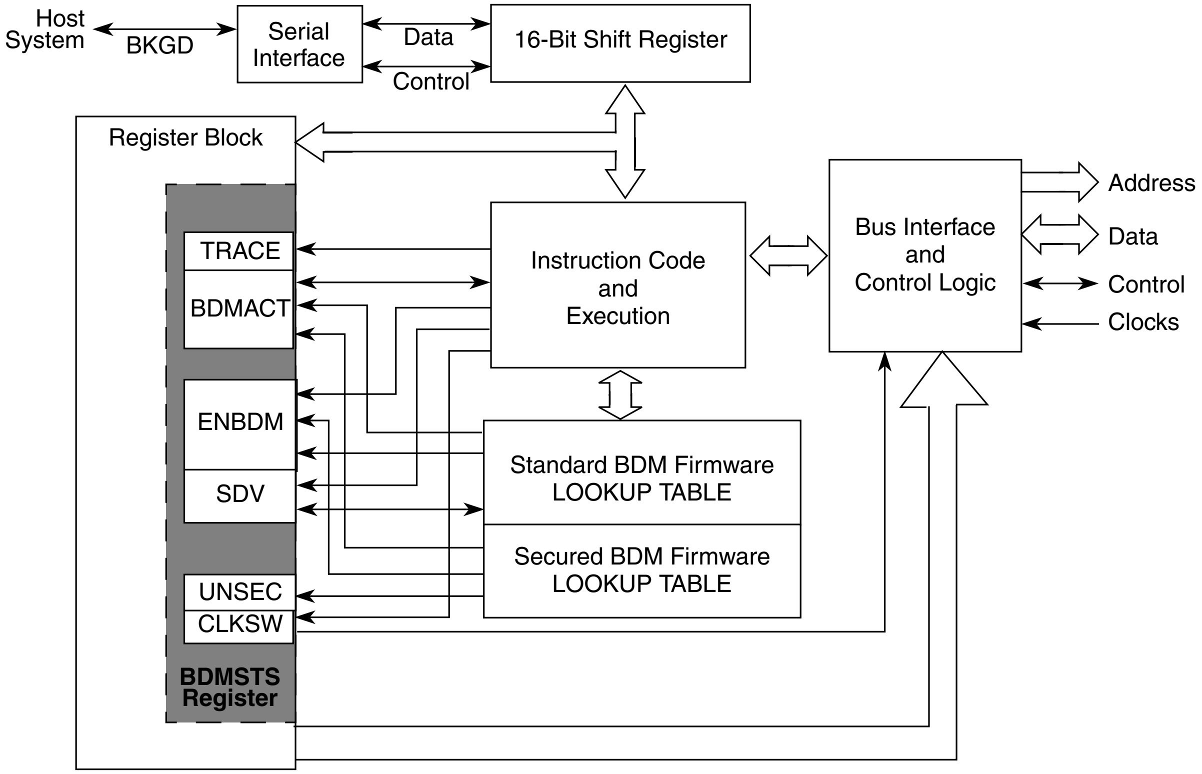


Figure 7-1. BDM Block Diagram

7.2 External Signal Description

A single-wire interface pin called the background debug interface (BKGD) pin is used to communicate with the BDM system. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the background debug mode.

7.3 Memory Map and Register Definition

7.3.1 Module Memory Map

Table 7-2 shows the BDM memory map when BDM is active.

Table 7-2. BDM Memory Map

Global Address	Module	Size (Bytes)
0x7FFF00–0x7FFF0B	BDM registers	12
0x7FFF0C–0x7FFF0E	BDM firmware ROM	3
0x7FFF0F	Family ID (part of BDM firmware ROM)	1
0x7FFF10–0x7FFFFF	BDM firmware ROM	240

7.3.2 Register Descriptions

A summary of the registers associated with the BDM is shown in Figure 7-2. Registers are accessed by host-driven communications to the BDM hardware using READ_BD and WRITE_BD commands.

Global Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x7FFF00	Reserved	R	X	X	X	X	X	X	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x7FFF01	BDMSTS	R	ENBDM	BDMACT	0	SDV	TRACE	CLKSW	UNSEC	0
		W		▨	▨	■	■		▨	▨
0x7FFF02	Reserved	R	X	X	X	X	X	X	X	X
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x7FFF03	Reserved	R	X	X	X	X	X	X	X	X
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x7FFF04	Reserved	R	X	X	X	X	X	X	X	X
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x7FFF05	Reserved	R	X	X	X	X	X	X	X	X
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x7FFF06	BDMCCRL	R	CCR7	CCR6	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0
		W								

▨ = Unimplemented, Reserved

■ = Implemented (do not alter)

X = Indeterminate

0 = Always read zero

Figure 7-2. BDM Register Summary

Global Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x7FFF07	BDMCCRH	R	0	0	0	0	0	CCR10	CCR9	CCR8
		W	▨	▨	▨	▨	▨			
0x7FFF08	BDMGPR	R	BGAE	BGP6	BGP5	BGP4	BGP3	BGP2	BGP1	BGP0
		W								
0x7FFF09	Reserved	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x7FFF0A	Reserved	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x7FFF0B	Reserved	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨

▨ = Unimplemented, Reserved   ■ = Implemented (do not alter)

X = Indeterminate   0 = Always read zero

Figure 7-2. BDM Register Summary (continued)

7.3.2.1 BDM Status Register (BDMSTS)

Register Global Address 0x7FFF01

	7	6	5	4	3	2	1	0
R	ENBDM	BDMACT	0	SDV	TRACE	CLKSW	UNSEC	0
W		▨	▨		■		▨	▨
Reset								
Special Single-Chip Mode	0[1]	1	0	0	0	0	0[3]	0
Emulation Modes (if modes available)	1	0	0	0	0	1[2]	0	0
All Other Modes	0	0	0	0	0	0	0	0

▨ = Unimplemented, Reserved   ■ = Implemented (do not alter)

0 = Always read zero

1. ENBDM is read as 1 by a debugging environment in special single chip mode when the device is not secured or secured but fully erased (non-volatile memory). This is because the ENBDM bit is set by the standard firmware before a BDM command can be fully transmitted and executed.
2. CLKSW is read as 1 by a debugging environment in emulation modes when the device is not secured and read as 0 when secured if emulation modes available.
3. UNSEC is read as 1 by a debugging environment in special single chip mode when the device is secured and fully erased, else it is 0 and can only be read if not secure (see also bit description).

Figure 7-3. BDM Status Register (BDMSTS)

Read: All modes through BDM operation when not secured

Write: All modes through BDM operation when not secured, but subject to the following:

- ENBDM should only be set via a BDM hardware command if the BDM firmware commands are needed. (This does not apply in special single chip and emulation modes).
- BDMACT can only be set by BDM hardware upon entry into BDM. It can only be cleared by the standard BDM firmware lookup table upon exit from BDM active mode.
- CLKSW can only be written via BDM hardware WRITE_BD commands.
- All other bits, while writable via BDM hardware or standard BDM firmware write commands, should only be altered by the BDM hardware or standard firmware lookup table as part of BDM command execution.

Table 7-3. BDMSTS Field Descriptions

Field	Description
7 ENBDM	**Enable BDM** — This bit controls whether the BDM is enabled or disabled. When enabled, BDM can be made active to allow firmware commands to be executed. When disabled, BDM cannot be made active but BDM hardware commands are still allowed. 0 BDM disabled 1 BDM enabled **Note:** ENBDM is set by the firmware out of reset in special single chip mode. In emulation modes (if modes available) the ENBDM bit is set by BDM hardware out of reset. In special single chip mode with the device secured, this bit will not be set by the firmware until after the non-volatile memory erase verify tests are complete. In emulation modes (if modes available) with the device secured, the BDM operations are blocked.
6 BDMACT	**BDM Active Status** — This bit becomes set upon entering BDM. The standard BDM firmware lookup table is then enabled and put into the memory map. BDMACT is cleared by a carefully timed store instruction in the standard BDM firmware as part of the exit sequence to return to user code and remove the BDM memory from the map. 0 BDM not active 1 BDM active
4 SDV	**Shift Data Valid** — This bit is set and cleared by the BDM hardware. It is set after data has been transmitted as part of a firmware or hardware read command or after data has been received as part of a firmware or hardware write command. It is cleared when the next BDM command has been received or BDM is exited. SDV is used by the standard BDM firmware to control program flow execution. 0 Data phase of command not complete 1 Data phase of command is complete
3 TRACE	**TRACE1 BDM Firmware Command is Being Executed** — This bit gets set when a BDM TRACE1 firmware command is first recognized. It will stay set until BDM firmware is exited by one of the following BDM commands: GO or GO_UNTIL. 0 TRACE1 command is not being executed 1 TRACE1 command is being executed

Table 7-3. BDMSTS Field Descriptions (continued)

Field	Description
2 CLKSW	**Clock Switch** — The CLKSW bit controls which clock the BDM operates with. It is only writable from a hardware BDM command. A minimum delay of 150 cycles at the clock speed that is active during the data portion of the command send to change the clock source should occur before the next command can be send. The delay should be obtained no matter which bit is modified to effectively change the clock source (either PLLSEL bit or CLKSW bit). This guarantees that the start of the next BDM command uses the new clock for timing subsequent BDM communications. Table 7-4 shows the resulting BDM clock source based on the CLKSW and the PLLSEL (PLL select in the CRG module, the bit is part of the CLKSEL register) bits. **Note:** The BDM alternate clock source can only be selected when CLKSW = 0 and PLLSEL = 1. The BDM serial interface is now fully synchronized to the alternate clock source, when enabled. This eliminates frequency restriction on the alternate clock which was required on previous versions. Refer to the device specification to determine which clock connects to the alternate clock source input. **Note:** If the acknowledge function is turned on, changing the CLKSW bit will cause the ACK to be at the new rate for the write command which changes it. **Note:** In emulation modes (if modes available), the CLKSW bit will be set out of RESET.
1 UNSEC	**Unsecure** — If the device is secured this bit is only writable in special single chip mode from the BDM secure firmware. It is in a zero state as secure mode is entered so that the secure BDM firmware lookup table is enabled and put into the memory map overlapping the standard BDM firmware lookup table. The secure BDM firmware lookup table verifies that the non-volatile memories (e.g. on-chip EEPROM and/or Flash EEPROM) are erased. This being the case, the UNSEC bit is set and the BDM program jumps to the start of the standard BDM firmware lookup table and the secure BDM firmware lookup table is turned off. If the erase test fails, the UNSEC bit will not be asserted. 0 System is in a secured mode. 1 System is in a unsecured mode. **Note:** When UNSEC is set, security is off and the user can change the state of the secure bits in the on-chip Flash EEPROM. Note that if the user does not change the state of the bits to “unsecured” mode, the system will be secured again when it is next taken out of reset.After reset this bit has no meaning or effect when the security byte in the Flash EEPROM is configured for unsecure mode.

Table 7-4. BDM Clock Sources

PLLSEL	CLKSW	BDMCLK
0	0	Bus clock dependent on oscillator
0	1	Bus clock dependent on oscillator
1	0	Alternate clock (refer to the device specification to determine the alternate clock source)
1	1	Bus clock dependent on the PLL

7.3.2.2 BDM CCR LOW Holding Register (BDMCCRL)

Register Global Address 0x7FFF06

	7	6	5	4	3	2	1	0
R W	CCR7	CCR6	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0
Reset								
Special Single-Chip Mode	1	1	0	0	1	0	0	0
All Other Modes	0	0	0	0	0	0	0	0

Figure 7-4. BDM CCR LOW Holding Register (BDMCCRL)

Read: All modes through BDM operation when not secured

Write: All modes through BDM operation when not secured

NOTE

When BDM is made active, the CPU stores the content of its CCR_L register in the BDMCCRL register. However, out of special single-chip reset, the BDMCCRL is set to 0xD8 and not 0xD0 which is the reset value of the CCR_L register in this CPU mode. Out of reset in all other modes the BDMCCRL register is read zero.

When entering background debug mode, the BDM CCR LOW holding register is used to save the low byte of the condition code register of the user's program. It is also used for temporary storage in the standard BDM firmware mode. The BDM CCR LOW holding register can be written to modify the CCR value.

7.3.2.3 BDM CCR HIGH Holding Register (BDMCCRH)

Register Global Address 0x7FFF07

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	CCR10	CCR9	CCR8
W	▨	▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 7-5. BDM CCR HIGH Holding Register (BDMCCRH)

Read: All modes through BDM operation when not secured

Write: All modes through BDM operation when not secured

When entering background debug mode, the BDM CCR HIGH holding register is used to save the high byte of the condition code register of the user's program. The BDM CCR HIGH holding register can be written to modify the CCR value.

7.3.2.4 BDM Global Page Index Register (BDMGPR)

Register Global Address 0x7FFF08

	7	6	5	4	3	2	1	0
R W	BGAE	BGP6	BGP5	BGP4	BGP3	BGP2	BGP1	BGP0
Reset	0	0	0	0	0	0	0	0

Figure 7-6. BDM Global Page Register (BDMGPR)

Read: All modes through BDM operation when not secured

Write: All modes through BDM operation when not secured

Table 7-5. BDMGPR Field Descriptions

Field	Description
7 BGAE	**BDM Global Page Access Enable Bit** — BGAE enables global page access for BDM hardware and firmware read/write instructions The BDM hardware commands used to access the BDM registers (READ_BD_ and WRITE_BD_) can not be used for global accesses even if the BGAE bit is set. 0 BDM Global Access disabled 1 BDM Global Access enabled
6–0 BGP[6:0]	**BDM Global Page Index Bits 6–0** — These bits define the extended address bits from 22 to 16. For more detailed information regarding the global page window scheme, please refer to the S12X_MMC Block Guide.

7.3.3 Family ID Assignment

The family ID is a 8-bit value located in the firmware ROM (at global address: 0x7FFF0F). The read-only value is a unique family ID which is 0xC1 for S12X devices.

7.4 Functional Description

The BDM receives and executes commands from a host via a single wire serial interface. There are two types of BDM commands: hardware and firmware commands.

Hardware commands are used to read and write target system memory locations and to enter active background debug mode, see Section 7.4.3, “BDM Hardware Commands”. Target system memory includes all memory that is accessible by the CPU.

Firmware commands are used to read and write CPU resources and to exit from active background debug mode, see Section 7.4.4, “Standard BDM Firmware Commands”. The CPU resources referred to are the accumulator (D), X index register (X), Y index register (Y), stack pointer (SP), and program counter (PC).

Hardware commands can be executed at any time and in any mode excluding a few exceptions as highlighted (see Section 7.4.3, “BDM Hardware Commands”) and in secure mode (see Section 7.4.1, “Security”). Firmware commands can only be executed when the system is not secure and is in active background debug mode (BDM).

7.4.1 Security

If the user resets into special single chip mode with the system secured, a secured mode BDM firmware lookup table is brought into the map overlapping a portion of the standard BDM firmware lookup table. The secure BDM firmware verifies that the on-chip non-volatile memory (e.g. EEPROM and Flash EEPROM) is erased. This being the case, the UNSEC and ENBDM bit will get set. The BDM program jumps to the start of the standard BDM firmware and the secured mode BDM firmware is turned off and all BDM commands are allowed. If the non-volatile memory does not verify as erased, the BDM firmware sets the ENBDM bit, without asserting UNSEC, and the firmware enters a loop. This causes the BDM hardware commands to become enabled, but does not enable the firmware commands. This allows the BDM hardware to be used to erase the non-volatile memory.

BDM operation is not possible in any other mode than special single chip mode when the device is secured. The device can be unsecured via BDM serial interface in special single chip mode only. More information regarding security is provided in the security section of the device documentation.

7.4.2 Enabling and Activating BDM

The system must be in active BDM to execute standard BDM firmware commands. BDM can be activated only after being enabled. BDM is enabled by setting the ENBDM bit in the BDM status (BDMSTS) register. The ENBDM bit is set by writing to the BDM status (BDMSTS) register, via the single-wire interface, using a hardware command such as WRITE_BD_BYTE.

After being enabled, BDM is activated by one of the following[1]:

- Hardware BACKGROUND command
- CPU BGND instruction
- External instruction tagging mechanism[2]
- Breakpoint force or tag mechanism[2]

When BDM is activated, the CPU finishes executing the current instruction and then begins executing the firmware in the standard BDM firmware lookup table. When BDM is activated by a breakpoint, the type of breakpoint used determines if BDM becomes active before or after execution of the next instruction.

NOTE

> If an attempt is made to activate BDM before being enabled, the CPU resumes normal instruction execution after a brief delay. If BDM is not enabled, any hardware BACKGROUND commands issued are ignored by the BDM and the CPU is not delayed.

In active BDM, the BDM registers and standard BDM firmware lookup table are mapped to addresses 0x7FFF00 to 0x7FFFFF. BDM registers are mapped to addresses 0x7FFF00 to 0x7FFF0B. The BDM uses these registers which are readable anytime by the BDM. However, these registers are not readable by user programs.

[1] BDM is enabled and active immediately out of special single-chip reset.

[2] This method is provided by the S12X_DBG module.

7.4.3 BDM Hardware Commands

Hardware commands are used to read and write target system memory locations and to enter active background debug mode. Target system memory includes all memory that is accessible by the CPU on the SOC which can be on-chip RAM, non-volatile memory (e.g. EEPROM, Flash EEPROM), I/O and control registers, and all external memory.

Hardware commands are executed with minimal or no CPU intervention and do not require the system to be in active BDM for execution, although, they can still be executed in this mode. When executing a hardware command, the BDM sub-block waits for a free bus cycle so that the background access does not disturb the running application program. If a free cycle is not found within 128 clock cycles, the CPU is momentarily frozen so that the BDM can steal a cycle. When the BDM finds a free cycle, the operation does not intrude on normal CPU operation provided that it can be completed in a single cycle. However, if an operation requires multiple cycles the CPU is frozen until the operation is complete, even though the BDM found a free cycle.

The BDM hardware commands are listed in Table 7-6.

The READ_BD and WRITE_BD commands allow access to the BDM register locations. These locations are not normally in the system memory map but share addresses with the application in memory. To distinguish between physical memory locations that share the same address, BDM memory resources are enabled just for the READ_BD and WRITE_BD access cycle. This allows the BDM to access BDM locations unobtrusively, even if the addresses conflict with the application memory map.

Table 7-6. Hardware Commands

Command	Opcode (hex)	Data	Description
BACKGROUND	90	None	Enter background mode if firmware is enabled. If enabled, an ACK will be issued when the part enters active background mode.
ACK_ENABLE	D5	None	Enable Handshake. Issues an ACK pulse after the command is executed.
ACK_DISABLE	D6	None	Disable Handshake. This command does not issue an ACK pulse.
READ_BD_BYTE	E4	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table in map. Odd address data on low byte; even address data on high byte.
READ_BD_WORD	EC	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table in map. Must be aligned access.
READ_BYTE	E0	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table out of map. Odd address data on low byte; even address data on high byte.
READ_WORD	E8	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table out of map. Must be aligned access.
WRITE_BD_BYTE	C4	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table in map. Odd address data on low byte; even address data on high byte.
WRITE_BD_WORD	CC	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table in map. Must be aligned access.
WRITE_BYTE	C0	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table out of map. Odd address data on low byte; even address data on high byte.

Table 7-6. Hardware Commands (continued)

Command	Opcode (hex)	Data	Description
WRITE_WORD	C8	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table out of map. Must be aligned access.

NOTE:
If enabled, ACK will occur when data is ready for transmission for all BDM READ commands and will occur after the write is complete for all BDM WRITE commands.

7.4.4 Standard BDM Firmware Commands

Firmware commands are used to access and manipulate CPU resources. The system must be in active BDM to execute standard BDM firmware commands, see Section 7.4.2, “Enabling and Activating BDM”. Normal instruction execution is suspended while the CPU executes the firmware located in the standard BDM firmware lookup table. The hardware command BACKGROUND is the usual way to activate BDM.

As the system enters active BDM, the standard BDM firmware lookup table and BDM registers become visible in the on-chip memory map at 0x7FFF00–0x7FFFFF, and the CPU begins executing the standard BDM firmware. The standard BDM firmware watches for serial commands and executes them as they are received.

The firmware commands are shown in Table 7-7.

Table 7-7. Firmware Commands

Command[1]	Opcode (hex)	Data	Description
READ_NEXT[2]	62	16-bit data out	Increment X index register by 2 (X = X + 2), then read word X points to.
READ_PC	63	16-bit data out	Read program counter.
READ_D	64	16-bit data out	Read D accumulator.
READ_X	65	16-bit data out	Read X index register.
READ_Y	66	16-bit data out	Read Y index register.
READ_SP	67	16-bit data out	Read stack pointer.
WRITE_NEXT	42	16-bit data in	Increment X index register by 2 (X = X + 2), then write word to location pointed to by X.
WRITE_PC	43	16-bit data in	Write program counter.
WRITE_D	44	16-bit data in	Write D accumulator.
WRITE_X	45	16-bit data in	Write X index register.
WRITE_Y	46	16-bit data in	Write Y index register.
WRITE_SP	47	16-bit data in	Write stack pointer.
GO	08	none	Go to user program. If enabled, ACK will occur when leaving active background mode.
GO_UNTIL[3]	0C	none	Go to user program. If enabled, ACK will occur upon returning to active background mode.
TRACE1	10	none	Execute one user instruction then return to active BDM. If enabled, ACK will occur upon returning to active background mode.
TAGGO -> GO	18	none	(Previous enable tagging and go to user program.) This command will be deprecated and should not be used anymore. Opcode will be executed as a GO command.

[1] If enabled, ACK will occur when data is ready for transmission for all BDM READ commands and will occur after the write is complete for all BDM WRITE commands.

[2] When the firmware command READ_NEXT or WRITE_NEXT is used to access the BDM address space the BDM resources are accessed rather than user code. Writing BDM firmware is not possible.

[3] System stop disables the ACK function and ignored commands will not have an ACK-pulse (e.g., CPU in stop or wait mode). The GO_UNTIL command will not get an Acknowledge if CPU executes the wait or stop instruction before the “UNTIL” condition (BDM active again) is reached (see Section 7.4.7, “Serial Interface Hardware Handshake Protocol” last Note).

7.4.5 BDM Command Structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.

8-bit reads return 16-bits of data, of which, only one byte will contain valid data. If reading an even address, the valid data will appear in the MSB. If reading an odd address, the valid data will appear in the LSB.

16-bit misaligned reads and writes are generally not allowed. If attempted by BDM hardware command, the BDM will ignore the least significant bit of the address and will assume an even address from the remaining bits.

For devices with external bus:

The following cycle count information is only valid when the external wait function is not used (see wait bit of EBI sub-block). During an external wait the BDM can not steal a cycle. Hence be careful with the external wait function if the BDM serial interface is much faster than the bus, because of the BDM soft-reset after time-out (see Section 7.4.11, “Serial Communication Time Out”).

For hardware data read commands, the external host must wait at least 150 bus clock cycles after sending the address before attempting to obtain the read data. This is to be certain that valid data is available in the BDM shift register, ready to be shifted out. For hardware write commands, the external host must wait 150 bus clock cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed. The 150 bus clock cycle delay in both cases includes the maximum 128 cycle delay that can be incurred as the BDM waits for a free cycle before stealing a cycle.

For firmware read commands, the external host should wait at least 48 bus clock cycles after sending the command opcode and before attempting to obtain the read data. This includes the potential of extra cycles when the access is external and stretched (+1 to maximum +7 cycles) or to registers of the PRU (port replacement unit) in emulation modes (if modes available). The 48 cycle wait allows enough time for the requested data to be made available in the BDM shift register, ready to be shifted out.

NOTE

This timing has increased from previous BDM modules due to the new capability in which the BDM serial interface can potentially run faster than the bus. On previous BDM modules this extra time could be hidden within the serial time.

For firmware write commands, the external host must wait 36 bus clock cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed.

The external host should wait at least for 76 bus clock cycles after a TRACE1 or GO command before starting any new serial command. This is to allow the CPU to exit gracefully from the standard BDM firmware lookup table and resume execution of the user code. Disturbing the BDM shift register prematurely may adversely affect the exit from the standard BDM firmware lookup table.

NOTE

If the bus rate of the target processor is unknown or could be changing or the external wait function is used, it is recommended that the ACK (acknowledge function) is used to indicate when an operation is complete. When using ACK, the delay times are automated.

Figure 7-7 represents the BDM command structure. The command blocks illustrate a series of eight bit times starting with a falling edge. The bar across the top of the blocks indicates that the BKGD line idles in the high state. The time for an 8-bit command is 8 × 16 target clock cycles.[1]

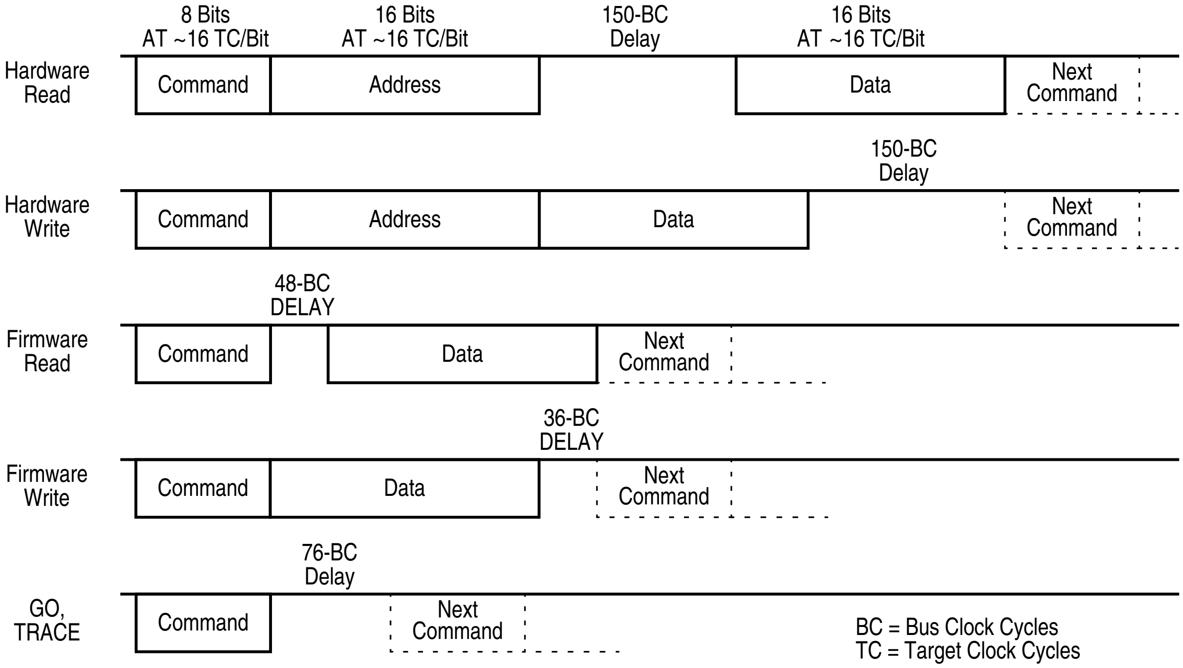


Figure 7-7. BDM Command Structure

7.4.6 BDM Serial Interface

The BDM communicates with external devices serially via the BKGD pin. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the BDM.

The BDM serial interface is timed using the clock selected by the CLKSW bit in the status register see Section 7.3.2.1, “BDM Status Register (BDMSTS)”. This clock will be referred to as the target clock in the following explanation.

The BDM serial interface uses a clocking scheme in which the external host generates a falling edge on the BKGD pin to indicate the start of each bit time. This falling edge is sent for every bit whether data is transmitted or received. Data is transferred most significant bit (MSB) first at 16 target clock cycles per bit. The interface times out if 512 clock cycles occur between falling edges from the host.

The BKGD pin is a pseudo open-drain pin and has an weak on-chip active pull-up that is enabled at all times. It is assumed that there is an external pull-up and that drivers connected to BKGD do not typically drive the high level. Since R-C rise time could be unacceptably long, the target system and host provide brief driven-high (speedup) pulses to drive BKGD to a logic 1. The source of this speedup pulse is the host for transmit cases and the target for receive cases.

The timing for host-to-target is shown in Figure 7-8 and that of target-to-host in Figure 7-9 and Figure 7-10. All four cases begin when the host drives the BKGD pin low to generate a falling edge. Since the host and target are operating from separate clocks, it can take the target system up to one full clock

[1] Target clock cycles are cycles measured using the target MCU’s serial clock rate. See Section 7.4.6, “BDM Serial Interface” and Section 7.3.2.1, “BDM Status Register (BDMSTS)” for information on how serial clock rate is selected.

cycle to recognize this edge. The target measures delays from this perceived start of the bit time while the host measures delays from the point it actually drove BKGD low to start the bit up to one target clock cycle earlier. Synchronization between the host and target is established in this manner at the start of every bit time.

Figure 7-8 shows an external host transmitting a logic 1 and transmitting a logic 0 to the BKGD pin of a target system. The host is asynchronous to the target, so there is up to a one clock-cycle delay from the host-generated falling edge to where the target recognizes this edge as the beginning of the bit time. Ten target clock cycles later, the target senses the bit level on the BKGD pin. Internal glitch detect logic requires the pin be driven high no later that eight target clock cycles after the falling edge for a logic 1 transmission.

Since the host drives the high speedup pulses in these two cases, the rising edges look like digitally driven signals.

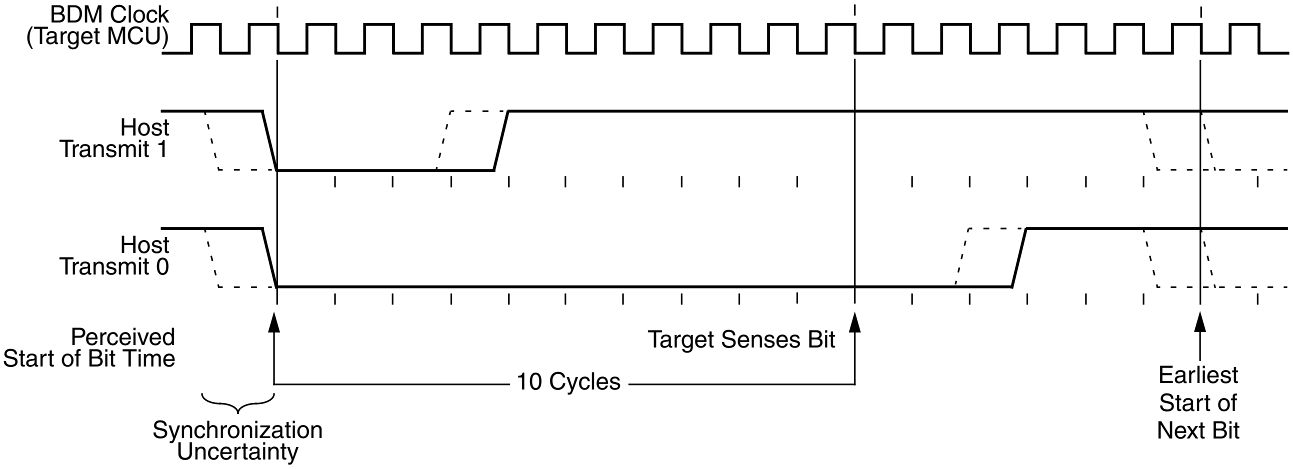


Figure 7-8. BDM Host-to-Target Serial Bit Timing

The receive cases are more complicated. Figure 7-9 shows the host receiving a logic 1 from the target system. Since the host is asynchronous to the target, there is up to one clock-cycle delay from the host-generated falling edge on BKGD to the perceived start of the bit time in the target. The host holds the BKGD pin low long enough for the target to recognize it (at least two target clock cycles). The host must release the low drive before the target drives a brief high speedup pulse seven target clock cycles after the perceived start of the bit time. The host should sample the bit level about 10 target clock cycles after it started the bit time.

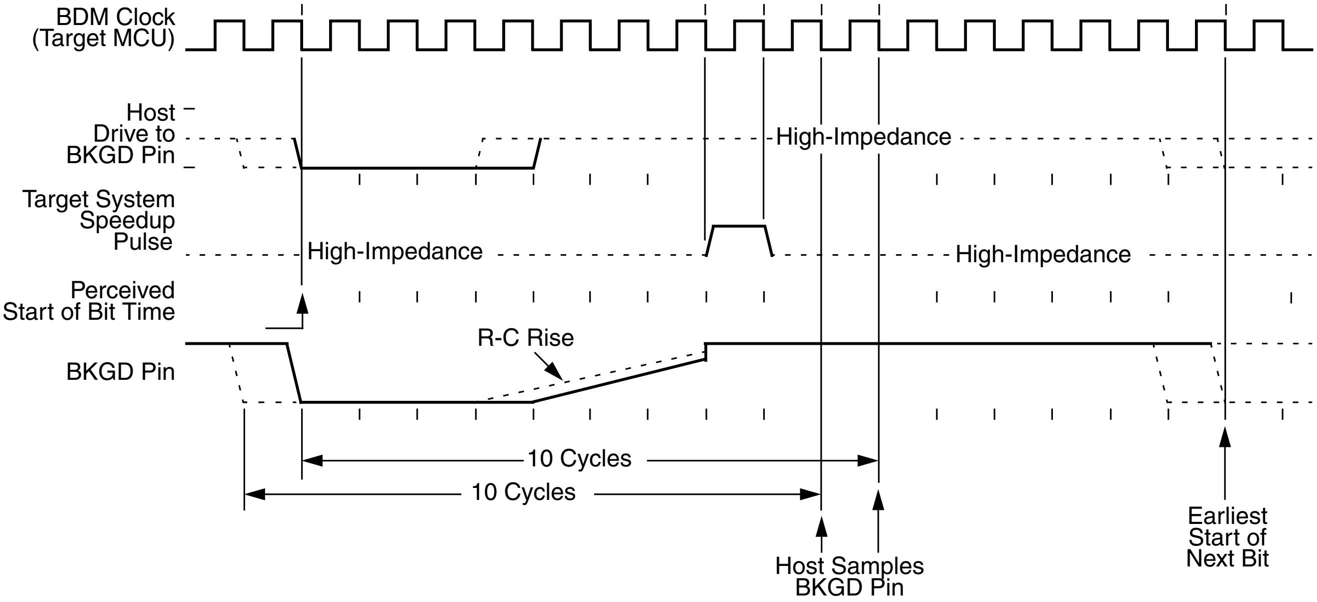


Figure 7-9. BDM Target-to-Host Serial Bit Timing (Logic 1)

Figure 7-10 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.

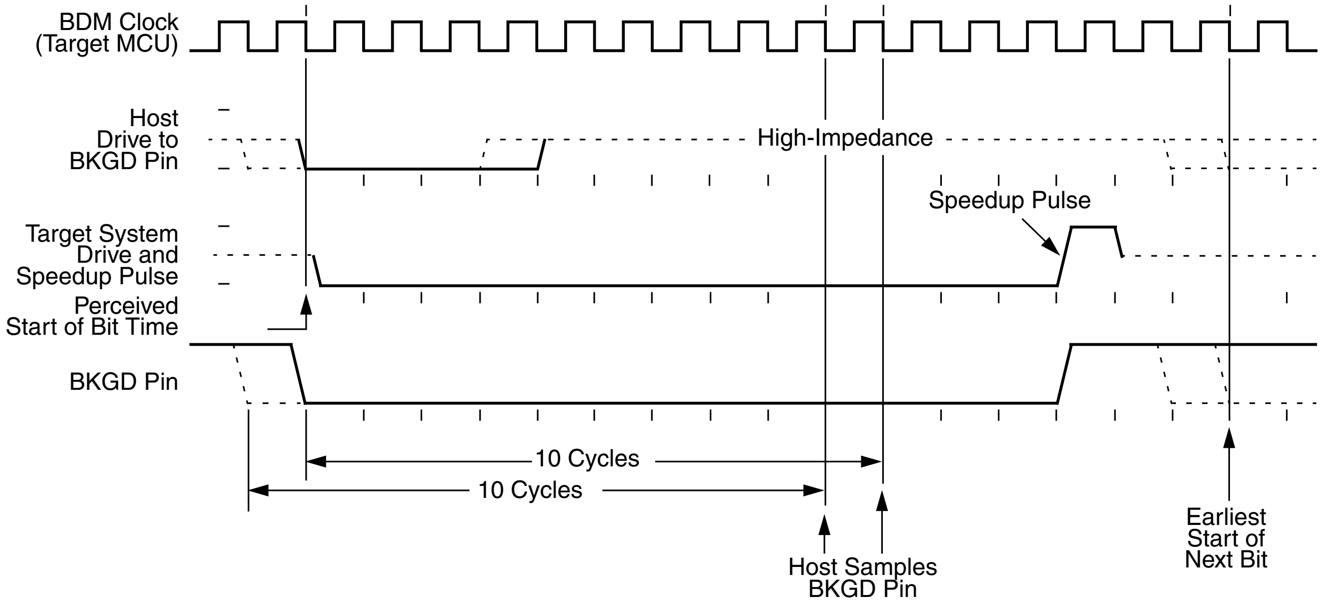


Figure 7-10. BDM Target-to-Host Serial Bit Timing (Logic 0)

7.4.7 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 7-11). 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 compared to the serial communication rate. This protocol allows a great flexibility for the POD designers, since it does not rely on any accurate time measurement or short response time to any event in the serial communication.

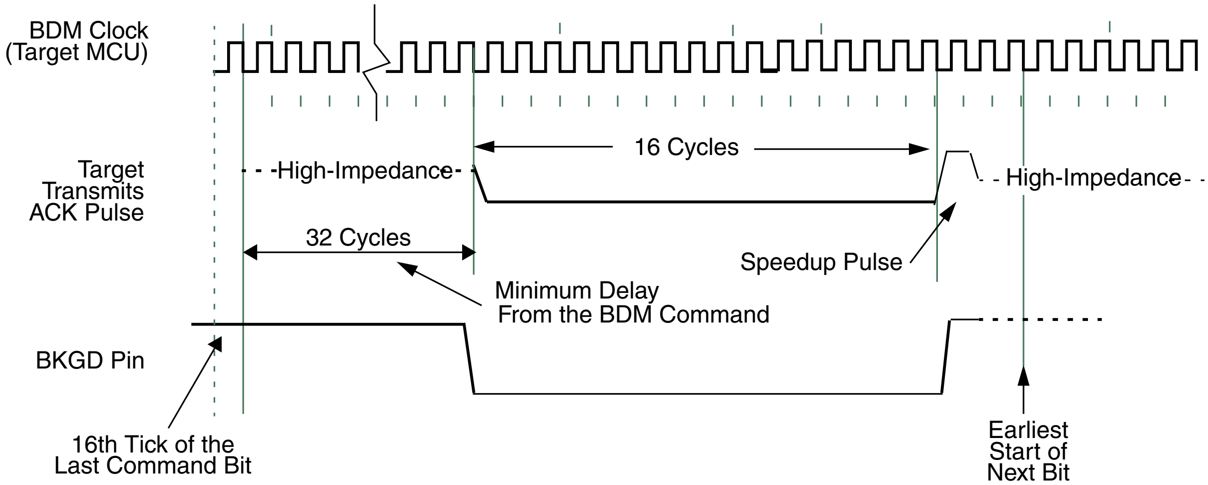


Figure 7-11. Target Acknowledge Pulse (ACK)

NOTE

> If the ACK pulse was issued by the target, the host assumes the previous command was executed. If the CPU enters wait or stop prior to executing a hardware command, the ACK pulse will not be issued meaning that the BDM command was not executed. After entering wait or stop mode, the BDM command is no longer pending.

Figure 7-12 shows the ACK handshake protocol in a command level timing diagram. The READ_BYTE instruction is used as an example. First, the 8-bit instruction opcode is sent by the host, followed by the address of the memory location to be read. The target BDM decodes the instruction. A bus cycle is grabbed (free or stolen) by the BDM and it executes the READ_BYTE operation. Having retrieved the data, the BDM issues an ACK pulse to the host controller, indicating that the addressed byte is ready to be retrieved. After detecting the ACK pulse, the host initiates the byte retrieval process. Note that data is sent in the form of a word and the host needs to determine which is the appropriate byte based on whether the address was odd or even.

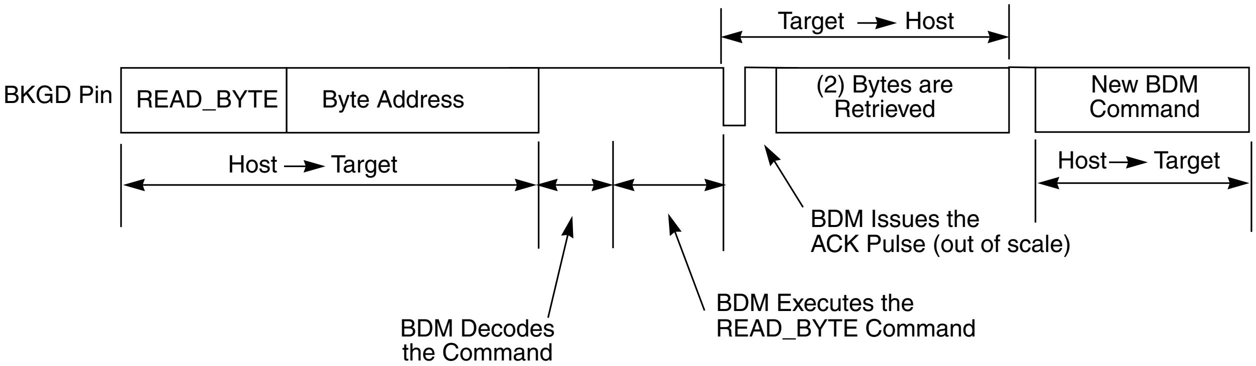


Figure 7-12. Handshake Protocol at Command Level

Differently from the normal bit transfer (where the host initiates the transmission), the serial interface ACK handshake pulse is initiated by the target MCU by issuing a negative edge in the BKGD pin. The hardware handshake protocol in Figure 7-11 specifies the timing when the BKGD pin is being driven, so the host should follow this timing constraint in order to avoid the risk of an electrical conflict in the BKGD pin.

NOTE

The only place the BKGD pin can have an electrical conflict is when one side is driving low and the other side is issuing a speedup pulse (high). Other "highs" are pulled rather than driven. However, at low rates the time of the speedup pulse can become lengthy and so the potential conflict time becomes longer as well.

The ACK handshake protocol does not support nested ACK pulses. If a BDM command is not acknowledge by an ACK pulse, the host needs to abort the pending command first in order to be able to issue a new BDM command. When the CPU enters wait or stop while the host issues a hardware command (e.g., WRITE_BYTE), the target discards the incoming command due to the wait or stop being detected. Therefore, the command is not acknowledged by the target, which means that the ACK pulse will not be issued in this case. After a certain time the host (not aware of stop or wait) should decide to abort any possible pending ACK pulse in order to be sure a new command can be issued. Therefore, the protocol provides a mechanism in which a command, and its corresponding ACK, can be aborted.

NOTE

The ACK pulse does not provide a time out. This means for the GO_UNTIL command that it can not be distinguished if a stop or wait has been executed (command discarded and ACK not issued) or if the "UNTIL" condition (BDM active) is just not reached yet. Hence in any case where the ACK pulse of a command is not issued the possible pending command should be aborted before issuing a new command. See the handshake abort procedure described in Section 7.4.8, "Hardware Handshake Abort Procedure".

7.4.8 Hardware Handshake Abort Procedure

The abort procedure is based on the SYNC command. In order to abort a command, which had not issued the corresponding ACK pulse, the host controller should generate a low pulse in the BKGD pin by driving it low for at least 128 serial clock cycles and then driving it high for one serial clock cycle, providing a speedup pulse. By detecting this long low pulse in the BKGD pin, the target executes the SYNC protocol, see Section 7.4.9, "SYNC — Request Timed Reference Pulse", and assumes that the pending command and therefore the related ACK pulse, are being aborted. Therefore, after the SYNC protocol has been completed the host is free to issue new BDM commands. For Firmware READ or WRITE commands it can not be guaranteed that the pending command is aborted when issuing a SYNC before the corresponding ACK pulse. There is a short latency time from the time the READ or WRITE access begins until it is finished and the corresponding ACK pulse is issued. The latency time depends on the firmware READ or WRITE command that is issued and if the serial interface is running on a different clock rate than the bus. When the SYNC command starts during this latency time the READ or WRITE command will not be aborted, but the corresponding ACK pulse will be aborted. A pending GO, TRACE1 or

GO_UNTIL command can not be aborted. Only the corresponding ACK pulse can be aborted by the SYNC command.

Although it is not recommended, the host could abort a pending BDM command by issuing a low pulse in the BKGD pin shorter than 128 serial clock cycles, which will not be interpreted as the SYNC command. The ACK is actually aborted when a negative edge is perceived by the target in the BKGD pin. The short abort pulse should have at least 4 clock cycles keeping the BKGD pin low, in order to allow the negative edge to be detected by the target. In this case, the target will not execute the SYNC protocol but the pending command will be aborted along with the ACK pulse. The potential problem with this abort procedure is when there is a conflict between the ACK pulse and the short abort pulse. In this case, the target may not perceive the abort pulse. The worst case is when the pending command is a read command (i.e., READ_BYTE). If the abort pulse is not perceived by the target the host will attempt to send a new command after the abort pulse was issued, while the target expects the host to retrieve the accessed memory byte. In this case, host and target will run out of synchronism. However, if the command to be aborted is not a read command the short abort pulse could be used. After a command is aborted the target assumes the next negative edge, after the abort pulse, is the first bit of a new BDM command.

NOTE

The details about the short abort pulse are being provided only as a reference for the reader to better understand the BDM internal behavior. It is not recommended that this procedure be used in a real application.

Since the host knows the target serial clock frequency, the SYNC command (used to abort a command) does not need to consider the lower possible target frequency. In this case, the host could issue a SYNC very close to the 128 serial clock cycles length. Providing a small overhead on the pulse length in order to assure the SYNC pulse will not be misinterpreted by the target. See Section 7.4.9, “SYNC — Request Timed Reference Pulse”.

Figure 7-13 shows a SYNC command being issued after a READ_BYTE, which aborts the READ_BYTE command. Note that, after the command is aborted a new command could be issued by the host computer.

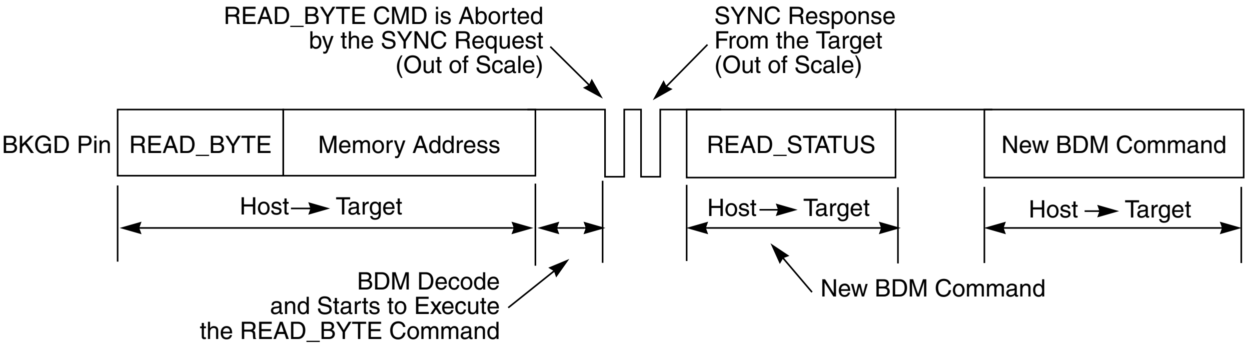


Figure 7-13. ACK Abort Procedure at the Command Level

NOTE

Figure 7-13 does not represent the signals in a true timing scale

Figure 7-14 shows a conflict between the ACK pulse and the SYNC request pulse. This conflict could occur if a POD device is connected to the target BKGD pin and the target is already in debug active mode.

Consider that the target CPU is executing a pending BDM command at the exact moment the POD is being connected to the BKGD pin. In this case, an ACK pulse is issued along with the SYNC command. In this case, there is an electrical conflict between the ACK speedup pulse and the SYNC pulse. Since this is not a probable situation, the protocol does not prevent this conflict from happening.

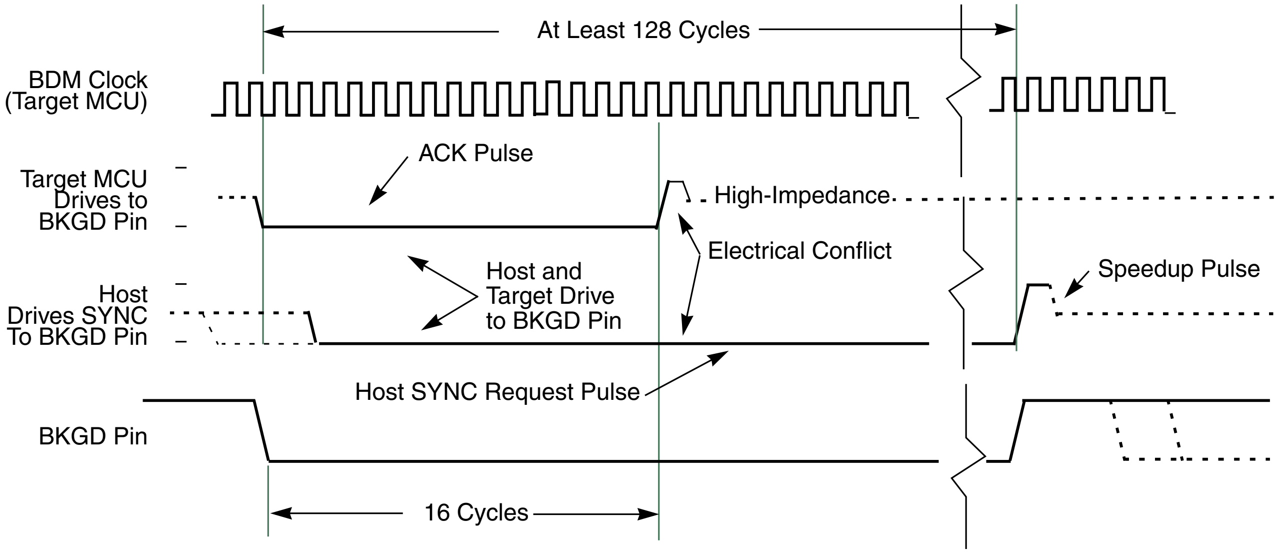


Figure 7-14. ACK Pulse and SYNC Request Conflict

NOTE

This information is being provided so that the MCU integrator will be aware that such a conflict could eventually occur.

The hardware handshake protocol is enabled by the ACK_ENABLE and disabled by the ACK_DISABLE BDM commands. This provides backwards compatibility with the existing POD devices which are not able to execute the hardware handshake protocol. It also allows for new POD devices, that support the hardware handshake protocol, to freely communicate with the target device. If desired, without the need for waiting for the ACK pulse.

The commands are described as follows:

- ACK_ENABLE — enables the hardware handshake protocol. The target will issue the ACK pulse when a CPU command is executed by the CPU. The ACK_ENABLE command itself also has the ACK pulse as a response.
- ACK_DISABLE — disables the ACK pulse protocol. In this case, the host needs to use the worst case delay time at the appropriate places in the protocol.

The default state of the BDM after reset is hardware handshake protocol disabled.

All the read commands will ACK (if enabled) when the data bus cycle has completed and the data is then ready for reading out by the BKGD serial pin. All the write commands will ACK (if enabled) after the data has been received by the BDM through the BKGD serial pin and when the data bus cycle is complete. See Section 7.4.3, “BDM Hardware Commands” and Section 7.4.4, “Standard BDM Firmware Commands” for more information on the BDM commands.

The ACK_ENABLE sends an ACK pulse when the command has been completed. This feature could be used by the host to evaluate if the target supports the hardware handshake protocol. If an ACK pulse is issued in response to this command, the host knows that the target supports the hardware handshake protocol. If the target does not support the hardware handshake protocol the ACK pulse is not issued. In this case, the ACK_ENABLE command is ignored by the target since it is not recognized as a valid command.

The BACKGROUND command will issue an ACK pulse when the CPU changes from normal to background mode. The ACK pulse related to this command could be aborted using the SYNC command.

The GO command will issue an ACK pulse when the CPU exits from background mode. The ACK pulse related to this command could be aborted using the SYNC command.

The GO_UNTIL command is equivalent to a GO command with exception that the ACK pulse, in this case, is issued when the CPU enters into background mode. This command is an alternative to the GO command and should be used when the host wants to trace if a breakpoint match occurs and causes the CPU to enter active background mode. Note that the ACK is issued whenever the CPU enters BDM, which could be caused by a breakpoint match or by a BGND instruction being executed. The ACK pulse related to this command could be aborted using the SYNC command.

The TRACE1 command has the related ACK pulse issued when the CPU enters background active mode after one instruction of the application program is executed. The ACK pulse related to this command could be aborted using the SYNC command.

7.4.9 SYNC — Request Timed Reference Pulse

The SYNC command is unlike other BDM commands because the host does not necessarily know the correct communication speed to use for BDM communications until after it has analyzed the response to the SYNC command. To issue a SYNC command, the host should perform the following steps:

1. Drive the BKGD pin low for at least 128 cycles at the lowest possible BDM serial communication frequency (the lowest serial communication frequency is determined by the crystal oscillator or the clock chosen by CLKSW.)
2. Drive BKGD high for a brief speedup pulse to get a fast rise time (this speedup pulse is typically one cycle of the host clock.)
3. Remove all drive to the BKGD pin so it reverts to high impedance.
4. Listen to the BKGD pin for the sync response pulse.

Upon detecting the SYNC request from the host, the target performs the following steps:

1. Discards any incomplete command received or bit retrieved.
2. Waits for BKGD to return to a logic one.
3. Delays 16 cycles to allow the host to stop driving the high speedup pulse.
4. Drives BKGD low for 128 cycles at the current BDM serial communication frequency.
5. Drives a one-cycle high speedup pulse to force a fast rise time on BKGD.
6. Removes all drive to the BKGD pin so it reverts to high impedance.

The host measures the low time of this 128 cycle SYNC response pulse and determines the correct speed for subsequent BDM communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

As soon as the SYNC request is detected by the target, any partially received command or bit retrieved is discarded. This is referred to as a soft-reset, equivalent to a time-out in the serial communication. After the SYNC response, the target will consider the next negative edge (issued by the host) as the start of a new BDM command or the start of new SYNC request.

Another use of the SYNC command pulse is to abort a pending ACK pulse. The behavior is exactly the same as in a regular SYNC command. Note that one of the possible causes for a command to not be acknowledged by the target is a host-target synchronization problem. In this case, the command may not have been understood by the target and so an ACK response pulse will not be issued.

7.4.10 Instruction Tracing

When a TRACE1 command is issued to the BDM in active BDM, the CPU exits the standard BDM firmware and executes a single instruction in the user code. Once this has occurred, the CPU is forced to return to the standard BDM firmware and the BDM is active and ready to receive a new command. If the TRACE1 command is issued again, the next user instruction will be executed. This facilitates stepping or tracing through the user code one instruction at a time.

If an interrupt is pending when a TRACE1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. Once back in standard BDM firmware execution, the program counter points to the first instruction in the interrupt service routine.

Be aware when tracing through the user code that the execution of the user code is done step by step but all peripherals are free running. Hence possible timing relations between CPU code execution and occurrence of events of other peripherals no longer exist.

Do not trace the CPU instruction BGND used for soft breakpoints. Tracing the BGND instruction will result in a return address pointing to BDM firmware address space.

When tracing through user code which contains stop or wait instructions the following will happen when the stop or wait instruction is traced:

The CPU enters stop or wait mode and the TRACE1 command can not be finished before leaving the low power mode. This is the case because BDM active mode can not be entered after CPU executed the stop instruction. However all BDM hardware commands except the BACKGROUND command are operational after tracing a stop or wait instruction and still being in stop or wait mode. If system stop mode is entered (all bus masters are in stop mode) no BDM command is operational.

As soon as stop or wait mode is exited the CPU enters BDM active mode and the saved PC value points to the entry of the corresponding interrupt service routine.

In case the handshake feature is enabled the corresponding ACK pulse of the TRACE1 command will be discarded when tracing a stop or wait instruction. Hence there is no ACK pulse when BDM active mode is entered as part of the TRACE1 command after CPU exited from stop or wait mode. All valid commands sent during CPU being in stop or wait mode or after CPU exited from stop or wait mode will have an ACK pulse. The handshake feature becomes disabled only when system stop mode has been reached. Hence after a system stop mode the handshake feature must be enabled again by sending the ACK_ENABLE command.

7.4.11 Serial Communication Time Out

The host initiates a host-to-target serial transmission by generating a falling edge on the BKGD pin. If BKGD is kept low for more than 128 target clock cycles, the target understands that a SYNC command was issued. In this case, the target will keep waiting for a rising edge on BKGD in order to answer the SYNC request pulse. If the rising edge is not detected, the target will keep waiting forever without any time-out limit.

Consider now the case where the host returns BKGD to logic one before 128 cycles. This is interpreted as a valid bit transmission, and not as a SYNC request. The target will keep waiting for another falling edge marking the start of a new bit. If, however, a new falling edge is not detected by the target within 512 clock cycles since the last falling edge, a time-out occurs and the current command is discarded without affecting memory or the operating mode of the MCU. This is referred to as a soft-reset.

If a read command is issued but the data is not retrieved within 512 serial clock cycles, a soft-reset will occur causing the command to be disregarded. The data is not available for retrieval after the time-out has occurred. This is the expected behavior if the handshake protocol is not enabled. However, consider the behavior where the BDM is running in a frequency much greater than the CPU frequency. In this case, the command could time out before the data is ready to be retrieved. In order to allow the data to be retrieved even with a large clock frequency mismatch (between BDM and CPU) when the hardware handshake protocol is enabled, the time out between a read command and the data retrieval is disabled. Therefore, the host could wait for more then 512 serial clock cycles and still be able to retrieve the data from an issued read command. However, once the handshake pulse (ACK pulse) is issued, the time-out feature is re-activated, meaning that the target will time out after 512 clock cycles. Therefore, the host needs to retrieve the data within a 512 serial clock cycles time frame after the ACK pulse had been issued. After that period, the read command is discarded and the data is no longer available for retrieval. Any negative edge in the BKGD pin after the time-out period is considered to be a new command or a SYNC request.

Note that whenever a partially issued command, or partially retrieved data, has occurred the time out in the serial communication is active. This means that if a time frame higher than 512 serial clock cycles is observed between two consecutive negative edges and the command being issued or data being retrieved is not complete, a soft-reset will occur causing the partially received command or data retrieved to be disregarded. The next negative edge in the BKGD pin, after a soft-reset has occurred, is considered by the target as the start of a new BDM command, or the start of a SYNC request pulse.

Chapter 8
S12X Debug (S12XDBGV3) Module

Table 8-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V03.20	14 Sep 2007	8.3.2.7/8-317	- Clarified reserved State Sequencer encodings.
V03.21	23 Oct 2007	8.4.2.2/8-329 8.4.2.4/8-330	- Added single databyte comparison limitation information - Added statement about interrupt vector fetches whilst tagging.
V03.22	12 Nov 2007	8.4.5.2/8-334 8.4.5.5/8-341	- Removed LOOP1 tracing restriction NOTE. - Added pin reset effect NOTE.
V03.23	13 Nov 2007	General	- Text readability improved, typo removed.
V03.24	04 Jan 2008	8.4.5.3/8-336	- Corrected bit name.
V03.25	14 May 2008	General	- Updated Revision History Table format. Corrected other paragraph formats.
V03.26	12 Sep 2012	General	- Added missing full stops. Removed redundant quotation marks.

8.1 Introduction

The S12XDBG module provides an on-chip trace buffer with flexible triggering capability to allow non-intrusive debug of application software. The S12XDBG module is optimized for the S12X 16-bit architecture and allows debugging of CPU12Xand XGATE module operations.

Typically the S12XDBG module is used in conjunction with the S12XBDM module, whereby the user configures the S12XDBG module for a debugging session over the BDM interface. Once configured the S12XDBG module is armed and the device leaves BDM Mode returning control to the user program, which is then monitored by the S12XDBG module. Alternatively the S12XDBG module can be configured over a serial interface using SWI routines.

8.1.1 Glossary

Table 8-2. Glossary Of Terms

Term	Definition
COF	Change Of Flow. Change in the program flow due to a conditional branch, indexed jump or interrupt
BDM	Background Debug Mode
DUG	Device User Guide, describing the features of the device into which the DBG is integrated
WORD	16-bit data entity

Table 8-2. Glossary Of Terms (continued)

Term	Definition
Data Line	64-bit data entity
CPU	CPU12X module
Tag	Tags can be attached to XGATE or CPU opcodes as they enter the instruction pipe. If the tagged opcode reaches the execution stage a tag hit occurs.

8.1.2 Overview

The comparators monitor the bus activity of the CPU12X and XGATE. When a match occurs the control logic can trigger the state sequencer to a new state. On a transition to the Final State, bus tracing is triggered and/or a breakpoint can be generated.

Independent of comparator matches a transition to Final State with associated tracing and breakpoint can be triggered by the external $\overline{\text{TAGHI}}$ and $\overline{\text{TAGLO}}$ signals, or by an XGATE module S/W breakpoint request or by writing to the TRIG control bit.

The trace buffer is visible through a 2-byte window in the register address map and can be read out using standard 16-bit word reads. Tracing is disabled when the MCU system is secured.

8.1.3 Features

- Four comparators (A, B, C, and D)
  - Comparators A and C compare the full address bus and full 16-bit data bus
  - Comparators A and C feature a data bus mask register
  - Comparators B and D compare the full address bus only
  - Each comparator can be configured to monitor CPU12X or XGATE buses
  - Each comparator features selection of read or write access cycles
  - Comparators B and D allow selection of byte or word access cycles
  - Comparisons can be used as triggers for the state sequencer
- Three comparator modes
  - Simple address/data comparator match mode
  - Inside address range mode, Addmin ≤ Address ≤ Addmax
  - Outside address range match mode, Address < Addmin or Address > Addmax
- Two types of triggers
  - Tagged — This triggers just before a specific instruction begins execution
  - Force — This triggers on the first instruction boundary after a match occurs.
- The following types of breakpoints
  - CPU12X breakpoint entering BDM on breakpoint (BDM)
  - CPU12X breakpoint executing SWI on breakpoint (SWI)
  - XGATE breakpoint
- External CPU12X instruction tagging trigger independent of comparators

- XGATE S/W breakpoint request trigger independent of comparators
- TRIG Immediate software trigger independent of comparators
- Four trace modes
  - Normal: change of flow (COF) PC information is stored (see Section 8.4.5.2.1) for change of flow definition.
  - Loop1: same as Normal but inhibits consecutive duplicate source address entries
  - Detail: address and data for all cycles except free cycles and opcode fetches are stored
  - Pure PC: All program counter addresses are stored.
- 4-stage state sequencer for trace buffer control
  - Tracing session trigger linked to Final State of state sequencer
  - Begin, End, and Mid alignment of tracing to trigger

8.1.4 Modes of Operation

The S12XDBG module can be used in all MCU functional modes.

During BDM hardware accesses and whilst the BDM module is active, CPU12X monitoring is disabled. Thus breakpoints, comparators, and CPU12X bus tracing are disabled but XGATE bus monitoring accessing the S12XDBG registers, including comparator registers, is still possible. While in active BDM or during hardware BDM accesses, XGATE activity can still be compared, traced and can be used to generate a breakpoint to the XGATE module. When the CPU12X enters active BDM Mode through a BACKGROUND command, with the S12XDBG module armed, the S12XDBG remains armed.

The S12XDBG module tracing is disabled if the MCU is secure. However, breakpoints can still be generated if the MCU is secure.

Table 8-3. Mode Dependent Restriction Summary

BDM Enable	BDM Active	MCU Secure	Comparator Matches Enabled	Breakpoints Possible	Tagging Possible	Tracing Possible
x	x	1	Yes	Yes	Yes	No
0	0	0	Yes	Only SWI	Yes	Yes
0	1	0	Active BDM not possible when not enabled			
1	0	0	Yes	Yes	Yes	Yes
1	1	0	XGATE only	XGATE only	XGATE only	XGATE only

8.1.5 Block Diagram

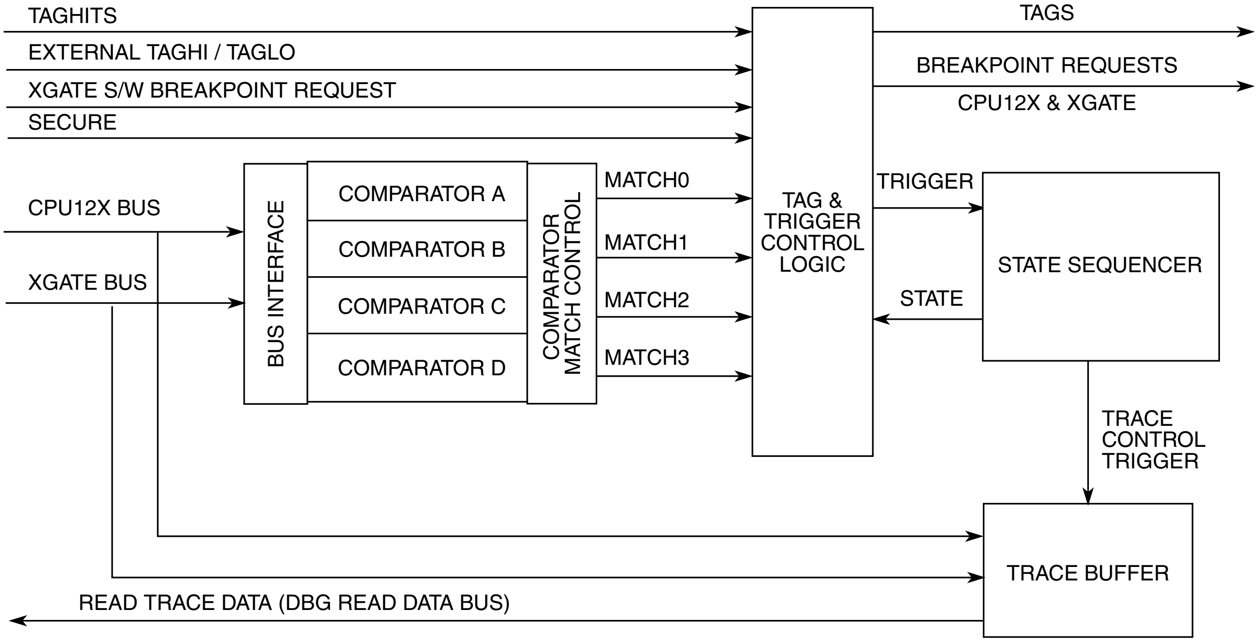


Figure 8-1. Debug Module Block Diagram

8.2 External Signal Description

The S12XDBG sub-module features two external tag input signals. See Device User Guide (DUG) for the mapping of these signals to device pins. These tag pins may be used for the external tagging in emulation modes only.

Table 8-4. External System Pins Associated With S12XDBG

Pin Name	Pin Functions	Description
$\overline{\text{TAGHI}}$ (See DUG)	TAGHI	When instruction tagging is on, tags the high half of the instruction word being read into the instruction queue.
$\overline{\text{TAGLO}}$ (See DUG)	TAGLO	When instruction tagging is on, tags the low half of the instruction word being read into the instruction queue.
$\overline{\text{TAGLO}}$ (See DUG)	Unconditional Tagging Enable	In emulation modes, a low assertion on this pin in the 7th or 8th cycle after the end of reset enables the Unconditional Tagging function.

8.3 Memory Map and Registers

8.3.1 Module Memory Map

A summary of the registers associated with the S12XDBG sub-block is shown in Table 8-2. Detailed descriptions of the registers and bits are given in the subsections that follow.

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0020	DBGC1	R	ARM	0	XGSBPE	BDM	DBGBRK		COMRV	
		W		TRIG						
0x0021	DBGSR	R	TBF	EXTF	0	0	0	SSF2	SSF1	SSF0
		W								
0x0022	DBGTCR	R	TSOURCE		TRANGE		TRCMOD		TALIGN	
		W								
0x0023	DBGC2	R	0	0	0	0	CDCM		ABCM	
		W								
0x0024	DBGTBH	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		W								
0x0025	DBGTBL	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		W								
0x0026	DBGCNT	R	0	CNT						
		W								
0x0027	DBGSCRX	R	0	0	0	0	SC3	SC2	SC1	SC0
		W								
0x0027	DBGMFR	R	0	0	0	0	MC3	MC2	MC1	MC0
		W								
0x0028[1]	DBGXCTL (COMPA/C)	R	0	NDB	TAG	BRK	RW	RWE	SRC	COMPE
		W								
0x0028[2]	DBGXCTL (COMPB/D)	R	SZE	SZ	TAG	BRK	RW	RWE	SRC	COMPE
		W								
0x0029	DBGXAH	R	0	Bit 22	21	20	19	18	17	Bit 16
		W								
0x002A	DBGXAM	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x002B	DBGXAL	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x002C	DBGXDH	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x002D	DBGXDL	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x002E	DBGXDHM	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x002F	DBGXDLM	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								

[1] This represents the contents if the Comparator A or C control register is blended into this address.

[2] This represents the contents if the Comparator B or D control register is blended into this address

Figure 8-2. Quick Reference to S12XDBG Registers

8.3.2 Register Descriptions

This section consists of the S12XDBG control and trace buffer register descriptions in address order. Each comparator has a bank of registers that are visible through an 8-byte window between 0x0028 and 0x002F in the S12XDBG module register address map. When ARM is set in DBGC1, the only bits in the S12XDBG module registers that can be written are ARM, TRIG, and COMRV[1:0].

8.3.2.1 Debug Control Register 1 (DBGC1)

Address: 0x0020

	7	6	5	4	3	2	1	0
R	ARM	0	XGSBPE	BDM	DBGBRK		COMRV	
W		TRIG						
Reset	0	0	0	0	0	0	0	0

Figure 8-3. Debug Control Register (DBGC1)

Read: Anytime

Write: Bits 7, 1, 0 anytime
Bit 6 can be written anytime but always reads back as 0.
Bits 5:2 anytime S12XDBG is not armed.

NOTE

If a write access to DBGC1 with the ARM bit position set occurs simultaneously to a hardware disarm from an internal trigger event, then the ARM bit is cleared due to the hardware disarm.

NOTE

When disarming the S12XDBG by clearing ARM with software, the contents of bits[5:2] are not affected by the write, since up until the write operation, ARM = 1 preventing these bits from being written. These bits must be cleared using a second write if required.

Table 8-5. DBGC1 Field Descriptions

Field	Description
7 ARM	**Arm Bit** — The ARM bit controls whether the S12XDBG module is armed. This bit can be set and cleared by user software and is automatically cleared on completion of a tracing session, or if a breakpoint is generated with tracing not enabled. On setting this bit the state sequencer enters State1. 0 Debugger disarmed 1 Debugger armed
6 TRIG	**Immediate Trigger Request Bit** — This bit when written to 1 requests an immediate trigger independent of comparator or external tag signal status. When tracing is complete a forced breakpoint may be generated depending upon DBGBRK and BDM bit settings. This bit always reads back a 0. Writing a 0 to this bit has no effect. If TSOURCE are clear no tracing is carried out. If tracing has already commenced using BEGIN- or MID trigger alignment, it continues until the end of the tracing session as defined by the TALIGN bit settings, thus TRIG has no affect. In secure mode tracing is disabled and writing to this bit has no effect. 0 Do not trigger until the state sequencer enters the Final State. 1 Trigger immediately .

Table 8-5. DBGC1 Field Descriptions (continued)

Field	Description
5 XGSBPE	**XGATE S/W Breakpoint Enable** — The XGSBPE bit controls whether an XGATE S/W breakpoint request is passed to the CPU12X. The XGATE S/W breakpoint request is handled by the S12XDBG module, which can request an CPU12X breakpoint depending on the state of this bit. 0 XGATE S/W breakpoint request is disabled 1 XGATE S/W breakpoint request is enabled
4 BDM	**Background Debug Mode Enable** — This bit determines if an S12X breakpoint causes the system to enter Background Debug Mode (BDM) or initiate a Software Interrupt (SWI). If this bit is set but the BDM is not enabled by the ENBDM bit in the BDM module, then breakpoints default to SWI. 0 Breakpoint to Software Interrupt if BDM inactive. Otherwise no breakpoint. 1 Breakpoint to BDM, if BDM enabled. Otherwise breakpoint to SWI
3–2 DBGBRK	**S12XDBG Breakpoint Enable Bits** — The DBGBRK bits control whether the debugger will request a breakpoint to either CPU12X or XGATE or both upon reaching the state sequencer Final State. If tracing is enabled, the breakpoint is generated on completion of the tracing session. If tracing is not enabled, the breakpoint is generated immediately. Please refer to Section 8.4.7 for further details. XGATE software breakpoints are independent of the DBGBRK bits. XGATE software breakpoints force a breakpoint to the CPU12X independent of the DBGBRK bit field configuration. See Table 8-6.
1–0 COMRV	**Comparator Register Visibility Bits** — These bits determine which bank of comparator register is visible in the 8-byte window of the S12XDBG module address map, located between 0x0028 to 0x002F. Furthermore these bits determine which register is visible at the address 0x0027. See Table 8-7.

Table 8-6. DBGBRK Encoding

DBGBRK	Resource Halted by Breakpoint
00	No breakpoint generated
01	XGATE breakpoint generated
10	CPU12X breakpoint generated
11	Breakpoints generated for CPU12X and XGATE

Table 8-7. COMRV Encoding

COMRV	Visible Comparator	Visible Register at 0x0027
00	Comparator A	DBGSCR1
01	Comparator B	DBGSCR2
10	Comparator C	DBGSCR3
11	Comparator D	DBGMFR

8.3.2.2 Debug Status Register (DBGSR)

Address: 0x0021

	7	6	5	4	3	2	1	0
R	TBF	EXTF	0	0	0	SSF2	SSF1	SSF0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	—	0	0	0	0	0	0	0
POR	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 8-4. Debug Status Register (DBGSR)

Read: Anytime

Write: Never

Table 8-8. DBGSR Field Descriptions

Field	Description
7 TBF	**Trace Buffer Full** — The TBF bit indicates that the trace buffer has stored 64 or more lines of data since it was last armed. If this bit is set, then all 64 lines will be valid data, regardless of the value of DBGCNT bits CNT[6:0]. The TBF bit is cleared when ARM in DBGC1 is written to a one. The TBF is cleared by the power on reset initialization. Other system generated resets have no affect on this bit.
6 EXTF	**External Tag Hit Flag** — The EXTF bit indicates if a tag hit condition from an external $\overline{\text{TAGHI}}$/$\overline{\text{TAGLO}}$ tag was met since arming. This bit is cleared when ARM in DBGC1 is written to a one. 0 External tag hit has not occurred 1 External tag hit has occurred
2–0 SSF[2:0]	**State Sequencer Flag Bits** — The SSF bits indicate in which state the State Sequencer is currently in. During a debug session on each transition to a new state these bits are updated. If the debug session is ended by software clearing the ARM bit, then these bits retain their value to reflect the last state of the state sequencer before disarming. If a debug session is ended by an internal trigger, then the state sequencer returns to state0 and these bits are cleared to indicate that state0 was entered during the session. On arming the module the state sequencer enters state1 and these bits are forced to SSF[2:0] = 001. See Table 8-9**.**

Table 8-9. SSF[2:0] — State Sequence Flag Bit Encoding

SSF[2:0]	Current State
000	State0 (disarmed)
001	State1
010	State2
011	State3
100	Final State
101,110,111	Reserved

8.3.2.3 Debug Trace Control Register (DBGTCR)

Address: 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 8-5. Debug Trace Control Register (DBGTCR)

Read: Anytime

Write: Bits 7:6 only when S12XDBG is neither secure nor armed.
Bits 5:0 anytime the module is disarmed.

Table 8-10. 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 8-11.
5–4 TRANGE	**Trace Range Bits** — The TRANGE bits allow filtering of trace information from a selected address range when tracing from the CPU12X 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. If the corresponding SRC bit is set the comparator is mapped to the XGATE buses, the TRANGE bits have no effect on the valid address range, memory accesses within the whole memory map are traced. See Table 8-12.
3–2 TRCMOD	**Trace Mode Bits** — See Section 8.4.5.2 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 8-13.
1–0 TALIGN	**Trigger Align Bits** — These bits control whether the trigger is aligned to the beginning, end or the middle of a tracing session. See Table 8-14.

Table 8-11. TSOURCE — Trace Source Bit Encoding

TSOURCE	Tracing Source
00	No tracing requested
01	CPU12X
10[1]	XGATE
11[1,2]	Both CPU12X 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 8-12. TRANGE Trace Range Encoding

TRANGE	Tracing Range
00	Trace from all addresses (No filter)
01	Trace only in address range from $00000 to Comparator D
10	Trace only in address range from Comparator C to $7FFFFF
11	Trace only in range from Comparator C to Comparator D

Table 8-13. TRCMOD Trace Mode Bit Encoding

TRCMOD	Description
00	Normal
01	Loop1
10	Detail
11	Pure PC

Table 8-14. TALIGN Trace Alignment Encoding

TALIGN	Description
00	Trigger at end of stored data
01	Trigger before storing data
10	Trace buffer entries before and after trigger
11	Reserved

8.3.2.4 Debug Control Register2 (DBGC2)

Address: 0x0023

	7	6	5	4	3	2	1	0
R	0	0	0	0	CDCM		ABCM	
W	▨	▨	▨	▨				
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 8-6. Debug Control Register2 (DBGC2)

Read: Anytime

Write: Anytime the module is disarmed.

This register configures the comparators for range matching.

Table 8-15. DBGC2 Field Descriptions

Field	Description
3–2 CDCM[1:0]	**C and D Comparator Match Control** — These bits determine the C and D comparator match mapping as described in Table 8-16.
1–0 ABCM[1:0]	**A and B Comparator Match Control** — These bits determine the A and B comparator match mapping as described in Table 8-17.

Table 8-16. CDCM Encoding

CDCM	Description
00	Match2 mapped to comparator C match....... Match3 mapped to comparator D match.
01	Match2 mapped to comparator C/D inside range....... Match3 disabled.
10	Match2 mapped to comparator C/D outside range....... Match3 disabled.
11	Reserved[1]

[1] Currently defaults to Match2 mapped to comparator C : Match3 mapped to comparator D

Table 8-17. ABCM Encoding

ABCM	Description
00	Match0 mapped to comparator A match....... Match1 mapped to comparator B match.
01	Match 0 mapped to comparator A/B inside range....... Match1 disabled.
10	Match 0 mapped to comparator A/B outside range....... Match1 disabled.
11	Reserved[1]

[1] Currently defaults to Match0 mapped to comparator A : Match1 mapped to comparator B

8.3.2.5 Debug Trace Buffer Register (DBGTBH:DBGTBL)

Address: 0x0024, 0x0025

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
POR	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Other Resets	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Figure 8-7. Debug Trace Buffer Register (DBGTB)

Read: Only when unlocked AND not secured AND not armed AND with a TSOURCE bit set.

Write: Aligned word writes when disarmed unlock the trace buffer for reading but do not affect trace buffer contents.

Table 8-18. DBGTB Field Descriptions

Field	Description
15–0 Bit[15:0]	**Trace Buffer Data Bits** — The Trace Buffer Register is a window through which the 64-bit wide data lines of the Trace Buffer may be read 16 bits at a time. Each valid read of DBGTB increments an internal trace buffer pointer which points to the next address to be read. When the ARM bit is written to 1 the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by writing to DBGTB with an aligned word write when the module is disarmed. The DBGTB register can be read only as an aligned word, any byte reads or misaligned access of these registers will return 0 and will not cause the trace buffer pointer to increment to the next trace buffer address. The same is true for word reads while the debugger is armed. The POR state is undefined Other resets do not affect the trace buffer contents. .

8.3.2.6 Debug Count Register (DBGCNT)

Address: 0x0026

	7	6	5	4	3	2	1	0
R	0	CNT						
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	—	—	—	—	—	—	—
POR	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 8-8. Debug Count Register (DBGCNT)

Read: Anytime

Write: Never

Table 8-19. DBGCNT Field Descriptions

Field	Description
6–0 CNT[6:0]	**Count Value** — The CNT bits [6:0] indicate the number of valid data 64-bit data lines stored in the Trace Buffer. Table 8-20 shows the correlation between the CNT bits and the number of valid data lines in the Trace Buffer. When the CNT rolls over to zero, the TBF bit in DBGSR is set and incrementing of CNT will continue in end-trigger or mid-trigger mode. The DBGCNT register is cleared when ARM in DBGC1 is written to a one. The DBGCNT register is cleared by power-on-reset initialization but is not cleared by other system resets. Thus should a reset occur during a debug session, the DBGCNT register still indicates after the reset, the number of valid trace buffer entries stored before the reset occurred. The DBGCNT register is not decremented when reading from the trace buffer.

Table 8-20. CNT Decoding Table

TBF (DBGSR)	CNT[6:0]	Description
0	0000000	No data valid
0	0000001	32 bits of one line valid[1]
0	0000010 0000100 0000110 .. 1111100	1 line valid 2 lines valid 3 lines valid .. 62 lines valid
0	1111110	63 lines valid
1	0000000	64 lines valid; if using Begin trigger alignment, ARM bit will be cleared and the tracing session ends.
1	0000010 1111110	64 lines valid, oldest data has been overwritten by most recent data

[1] This applies to Normal/Loop1/PurePC Modes when tracing from either CPU12X or XGATE only.

8.3.2.7 Debug State Control Registers

There is a dedicated control register for each of the state sequencer states 1 to 3 that determines if transitions from that state are allowed, depending upon comparator matches or tag hits, and defines the next state for the state sequencer following a match. The three debug state control registers are located at the same address in the register address map (0x0027). Each register can be accessed using the COMRV bits in DBGC1 to blend in the required register. The COMRV = 11 value blends in the match flag register (DBGMFR).

Table 8-21. State Control Register Access Encoding

COMRV	Visible State Control Register
00	DBGSCR1
01	DBGSCR2
10	DBGSCR3
11	DBGMFR

8.3.2.7.1 Debug State Control Register 1 (DBGSCR1)

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 8-9. Debug State Control Register 1 (DBGSCR1)

Read: If COMRV[1:0] = 00

Write: If COMRV[1:0] = 00 and S12XDBG is not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 00. The state control register 1 selects the targeted next state whilst in State1. The matches refer to the match channels of the comparator match control logic as depicted in Figure 8-1 and described in Section 8.3.2.8.1. Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 8-22. DBGSCR1 Field Descriptions

Field	Description
3–0 SC[3:0]	These bits select the targeted next state whilst in State1, based upon the match event.

Table 8-23. State1 Sequencer Next State Selection

SC[3:0]	Description
0000	Any match triggers to state2
0001	Any match triggers to state3
0010	Any match triggers to Final State

Table 8-23. State1 Sequencer Next State Selection (continued)

SC[3:0]	Description
0011	Match2 triggers to State2....... Other matches have no effect
0100	Match2 triggers to State3....... Other matches have no effect
0101	Match2 triggers to Final State....... Other matches have no effect
0110	Match0 triggers to State2....... 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 State2....... 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 State2....... Match3 triggers to State3....... Other matches have no effect
1011	Match3 triggers to State3....... Match1 triggers to Final State....... Other matches have no effect
1100	Match3 has no effect....... All other matches (M0,M1,M2) trigger to State2
1101	Reserved. (No match triggers state sequencer transition)
1110	Reserved. (No match triggers state sequencer transition)
1111	Reserved. (No match triggers state sequencer transition)

The trigger priorities described in Table 8-42 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.

8.3.2.7.2 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 8-10. Debug State Control Register 2 (DBGSCR2)

Read: If COMRV[1:0] = 01

Write: If COMRV[1:0] = 01 and S12XDBG is 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 8-1 and described in Section 8.3.2.8.1. Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 8-24. 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 8-25. State2 —Sequencer Next State Selection

SC[3:0]	Description
0000	Any match triggers to state1

Table 8-25. State2 —Sequencer Next State Selection (continued)

SC[3:0]	Description
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. (No match triggers state sequencer transition)
1111	Reserved. (No match triggers state sequencer transition)

The trigger priorities described in Table 8-42 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.

8.3.2.7.3 Debug State Control Register 3 (DBGSCR3)

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 8-11. Debug State Control Register 3 (DBGSCR3)

Read: If COMRV[1:0] = 10

Write: If COMRV[1:0] = 10 and S12XDBG is not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 10. The state control register three selects the targeted next state whilst in State3. The matches refer to the match channels of the comparator match control logic as depicted in Figure 8-1 and described in Section 8.3.2.8.1. Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 8-26. DBGSCR3 Field Descriptions

Field	Description
3–0 SC[3:0]	These bits select the targeted next state whilst in State3, based upon the match event.

Table 8-27. State3 — Sequencer Next State Selection

SC[3:0]	Description
0000	Any match triggers to state1
0001	Any match triggers to state2
0010	Any match triggers to Final State
0011	Match0 triggers to State1....... Other matches have no effect
0100	Match0 triggers to State2....... Other matches have no effect
0101	Match0 triggers to Final State.......Match1 triggers to State1...Other matches have no effect
0110	Match1 triggers to State1....... Other matches have no effect
0111	Match1 triggers to State2....... Other matches have no effect
1000	Match1 triggers to Final State....... Other matches have no effect
1001	Match2 triggers to State2....... Match0 triggers to Final State....... Other matches have no effect
1010	Match1 triggers to State1....... Match3 triggers to State2....... Other matches have no effect
1011	Match3 triggers to State2....... Match1 triggers to Final State....... Other matches have no effect
1100	Match2 triggers to Final State....... Other matches have no effect
1101	Match3 triggers to Final State....... Other matches have no effect
1110	Reserved. (No match triggers state sequencer transition)
1111	Reserved. (No match triggers state sequencer transition)

The trigger priorities described in Table 8-42 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.

8.3.2.7.4 Debug Match Flag Register (DBGMFR)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	MC3	MC2	MC1	MC0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 8-12. Debug Match Flag Register (DBGMFR)

Read: If COMRV[1:0] = 11

Write: Never

DBGMFR is visible at 0x0027 only with COMRV[1:0] = 11. It features four flag bits each mapped directly to a channel. Should a match occur on the channel during the debug session, then the corresponding flag is set and remains set until the next time the module is armed by writing to the ARM bit. Thus the contents are retained after a debug session for evaluation purposes. These flags cannot be cleared by software, they are cleared only when arming the module. A set flag does not inhibit the setting of other flags. Once a flag is set, further triggers on the same channel have no affect.

8.3.2.8 Comparator Register Descriptions

Each comparator has a bank of registers that are visible through an 8-byte window in the S12XDBG module register address map. Comparators A and C consist of 8 register bytes (3 address bus compare registers, two data bus compare registers, two data bus mask registers and a control register).

Comparators B and D consist of four register bytes (three address bus compare registers and a control register).

Each set of comparator registers is accessible in the same 8-byte window of the register address map and can be accessed using the COMRV bits in the DBGC1 register. If the Comparators B or D are accessed through the 8-byte window, then only the address and control bytes are visible, the 4 bytes associated with data bus and data bus masking read as zero and cannot be written. Furthermore the control registers for comparators B and D differ from those of comparators A and C.

Table 8-28. Comparator Register Layout

0x0028	CONTROL	Read/Write	Comparators A,B,C,D
0x0029	ADDRESS HIGH	Read/Write	Comparators A,B,C,D
0x002A	ADDRESS MEDIUM	Read/Write	Comparators A,B,C,D
0x002B	ADDRESS LOW	Read/Write	Comparators A,B,C,D
0x002C	DATA HIGH COMPARATOR	Read/Write	Comparator A and C only
0x002D	DATA LOW COMPARATOR	Read/Write	Comparator A and C only
0x002E	DATA HIGH MASK	Read/Write	Comparator A and C only
0x002F	DATA LOW MASK	Read/Write	Comparator A and C only

8.3.2.8.1 Debug Comparator Control Register (DBGXCTL)

The contents of this register bits 7 and 6 differ depending upon which comparator registers are visible in the 8-byte window of the DBG module register address map.

Address: 0x0028

	7	6	5	4	3	2	1	0
R	0	NDB	TAG	BRK	RW	RWE	SRC	COMPE
W	▨							
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 8-13. Debug Comparator Control Register (Comparators A and C)

Address: 0x0028

	7	6	5	4	3	2	1	0
R W	SZE	SZ	TAG	BRK	RW	RWE	SRC	COMPE
Reset	0	0	0	0	0	0	0	0

Figure 8-14. Debug Comparator Control Register (Comparators B and D)

Read: Anytime. See Table 8-29 for visible register encoding.

Write: If DBG not armed. See Table 8-29 for visible register encoding.

The DBGC1_COMRV bits determine which comparator control, address, data and datamask registers are visible in the 8-byte window from 0x0028 to 0x002F as shown in Section Table 8-29.

Table 8-29. Comparator Address Register Visibility

COMRV	Visible Comparator
00	DBGACTL, DBGAAH ,DBGAAM, DBGAAL, DBGADH, DBGADL, DBGADHM, DBGADLM
01	DBGBCTL, DBGBAH, DBGBAM, DBGBAL
10	DBGCCTL, DBGCAH, DBGCAM, DBGCAL, DBGCDH, DBGCDL, DBGCDHM, DBGCDLM
11	DBGDCTL, DBGDAH, DBGDAM, DBGDAL

Table 8-30. DBGXCTL Field Descriptions

Field	Description
7 SZE (Comparators B and D)	**Size Comparator Enable Bit** — The SZE bit controls whether access size comparison is enabled for the associated comparator. This bit is ignored if the TAG bit in the same register is set. 0 Word/Byte access size is not used in comparison 1 Word/Byte access size is used in comparison
6 NDB (Comparators A and C	**Not Data Bus** — The NDB bit controls whether the match occurs when the data bus matches the comparator register value or when the data bus differs from the register value. Furthermore data bus bits can be individually masked using the comparator data mask registers. This bit is only available for comparators A and C. This bit is ignored if the TAG bit in the same register is set. This bit position has an SZ functionality for comparators B and D. 0 Match on data bus equivalence to comparator register contents 1 Match on data bus difference to comparator register contents
6 SZ (Comparators B and D)	**Size Comparator Value Bit** — The SZ bit selects either word or byte access size in comparison for the associated comparator. This bit is ignored if the SZE bit is cleared or if the TAG bit in the same register is set. This bit position has NDB functionality for comparators A and C 0 Word access size will be compared 1 Byte access size will be compared
5 TAG	**Tag Select** — This bit controls whether the comparator match will cause a trigger or tag the opcode at the matched address. Tagged opcodes trigger only if they reach the execution stage of the instruction queue. 0 Trigger immediately on match 1 On match, tag the opcode. If the opcode is about to be executed a trigger is generated
4 BRK	**Break** — This bit controls whether a channel match terminates a debug session immediately, independent of state sequencer state. To generate an immediate breakpoint the module breakpoints must be enabled using DBGBRK. 0 The debug session termination is dependent upon the state sequencer and trigger conditions. 1 A match on this channel terminates the debug session immediately; breakpoints if active are generated, tracing, if active, is terminated and the module disarmed.
3 RW	**Read/Write Comparator Value Bit** — The RW bit controls whether read or write is used in compare for the associated comparator . The RW bit is not used if RWE = 0. 0 Write cycle will be matched 1 Read cycle will be matched
2 RWE	**Read/Write Enable Bit** — The RWE bit controls whether read or write comparison is enabled for the associated comparator. This bit is not used for tagged operations. 0 Read/Write is not used in comparison 1 Read/Write is used in comparison

Table 8-30. DBGXCTL Field Descriptions (continued)

Field	Description
1 SRC	Determines mapping of comparator to CPU12X or XGATE 0 The comparator is mapped to CPU12X buses 1 The comparator is mapped to XGATE address and data buses
0 COMPE	Determines if comparator is enabled 0 The comparator is not enabled 1 The comparator is enabled for state sequence triggers or tag generation

Table 8-31 shows the effect for RWE and RW on the comparison conditions. These bits are not useful for tagged operations since the trigger occurs based on the tagged opcode reaching the execution stage of the instruction queue. Thus these bits are ignored if tagged triggering is selected.

Table 8-31. Read or Write Comparison Logic Table

RWE Bit	RW Bit	RW Signal	Comment
0	x	0	RW not used in comparison
0	x	1	RW not used in comparison
1	0	0	Write
1	0	1	No match
1	1	0	No match
1	1	1	Read

8.3.2.8.2 Debug Comparator Address High Register (DBGXAH)

Address: 0x0029

	7	6	5	4	3	2	1	0
R	0	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W	▨							
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Figure 8-15. Debug Comparator Address High Register (DBGXAH)

Read: Anytime. See Table 8-29 for visible register encoding.

Write: If DBG not armed. See Table 8-29 for visible register encoding.

Table 8-32. DBGXAH Field Descriptions

Field	Description
6–0 Bit[22:16]	**Comparator Address High Compare Bits** — The Comparator address high compare bits control whether the selected comparator will compare the address bus bits [22:16] to a logic one or logic zero. This register byte is ignored for XGATE compares. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

8.3.2.8.3 Debug Comparator Address Mid Register (DBGXAM)

Address: 0x002A

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

Figure 8-16. Debug Comparator Address Mid Register (DBGXAM)

Read: Anytime. See Table 8-29 for visible register encoding.

Write: If DBG not armed. See Table 8-29 for visible register encoding.

Table 8-33. DBGXAM Field Descriptions

Field	Description
7–0 Bit[15:8]	**Comparator Address Mid Compare Bits**— The Comparator address mid compare bits control whether the selected comparator will compare the address bus bits [15:8] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

8.3.2.8.4 Debug Comparator Address Low Register (DBGXAL)

Address: 0x002B

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

Figure 8-17. Debug Comparator Address Low Register (DBGXAL)

Read: Anytime. See Table 8-29 for visible register encoding.

Write: If DBG not armed. See Table 8-29 for visible register encoding.

Table 8-34. DBGXAL Field Descriptions

Field	Description
7–0 Bits[7:0]	**Comparator Address Low Compare Bits** — The Comparator address low compare bits control whether the selected comparator will compare the address bus bits [7:0] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one