Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	HCS12X
Core Size	16-Bit
Speed	80MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	59
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12xdt256maa

Chapter 27
512 Kbyte Flash Module (S12XFTX512K4V2)

5.3.2.6 ATD Control Register 5 (ATDCTL5)

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

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

= Unimplemented or Reserved

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

Read: Anytime

Write: Anytime

Table 5-13. ATDCTL5 Field Descriptions

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

BGT Branch if Greater than Zero BGT

Operation

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

Branch instruction to compare signed numbers.

Branch if RS1 > RS2:

```
SUB     R0,RS1,RS2
BGE     REL9
```

CCR Effects

N	Z	V	C
—	—	—	—

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

Code and CPU Cycles

Source Form	Address Mode	Machine Code								Cycles
BGT REL9	REL9	0	0	1	1	1	0	0	REL9	PP/P

CMPL — Compare Immediate 8 bit Constant (Low Byte) — CMPL

Operation

RS.L – IMM8 ⇒ NONE, only condition code flags get updated

Subtracts the 8 bit constant IMM8 contained in the instruction code from the low byte of the source register RS.L using binary subtraction and updates the condition code register accordingly.

Remark: There is no equivalent operation using triadic addressing. Comparing the values of two registers can be performed by using the subtract instruction with R0 as destination register.

CCR Effects

N	Z	V	C
Δ	Δ	Δ	Δ

N: Set if bit 7 of the result is set; cleared otherwise.

Z: Set if the 8 bit result is $00; cleared otherwise.

V: Set if a two´s complement overflow resulted from the 8 bit operation; cleared otherwise.
RS[7] & $\overline{\text{IMM8[7]}}$ & $\overline{\text{result[7]}}$ | $\overline{\text{RS[7]}}$ & IMM8[7] & result[7]

C: Set if there is a carry from the Bit 7 to Bit 8 of the result; cleared otherwise.
$\overline{\text{RS[7]}}$ & IMM8[7] | $\overline{\text{RS[7]}}$ & result[7] | IMM8[7] & result[7]

Code and CPU Cycles

Source Form	Address Mode	Machine Code							Cycles
CMPL RS, #IMM8	IMM8	1	1	0	1	0	RS	IMM8	P

ORL — Logical OR Immediate 8 bit Constant (Low Byte) — ORL

Operation

RD.L | IMM8 ⇒ RD.L

Performs a bit wise logical OR between the low byte of register RD and an immediate 8 bit constant and stores the result in the destination register RD.L. The high byte of RD is not affected.

CCR Effects

N	Z	V	C
Δ	Δ	0	—

N: Set if bit 7 of the result is set; cleared otherwise.
Z: Set if the 8 bit result is $00; cleared otherwise.
V: 0; cleared.
C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code							Cycles
ORL RD, #IMM8	IMM8	1	0	1	0	0	RD	IMM8	P

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
TSCR2	R	TOI	0	0	0	TCRE	PR2	PR1	PR0
	W								
TFLG1	R	C7F	C6F	C5F	C4F	C3F	C2F	C1F	C0F
	W								
TFLG2	R	TOF	0	0	0	0	0	0	0
	W								
TC0 (High)	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
TC0 (Low)	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
TC1 (High)	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
TC1 (Low)	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
TC2 (High)	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
TC2 (Low)	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
TC3 (High)	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
TC3 (Low)	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
TC4 (High)	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
TC4 (Low)	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
TC5 (High)	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
TC5 (Low)	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								

= Unimplemented or Reserved

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

Table 7-30. ICSYS Field Descriptions (continued)

Field	Description
0 LATQ	**Input Control Latch or Queue Mode Enable** — The BUFEN control bit should be set in order to enable the IC and pulse accumulators holding registers. Otherwise LATQ latching modes are disabled. Write one into ICLAT bit in MCCTL, when LATQ and BUFEN are set will produce latching of input capture and pulse accumulators registers into their holding registers. 0 Queue mode of Input Capture is enabled. The main timer value is memorized in the IC register by a valid input pin transition. With a new occurrence of a capture, the value of the IC register will be transferred to its holding register and the IC register memorizes the new timer value. 1 Latch mode is enabled. Latching function occurs when modulus down-counter reaches zero or a zero is written into the count register MCCNT (see Section 7.4.1.1.2, “Buffered IC Channels”). With a latching event the contents of IC registers and 8-bit pulse accumulators are transferred to their holding registers. 8-bit pulse accumulators are cleared.

7.3.2.25 Precision Timer Prescaler Select Register (PTPSR)

	7	6	5	4	3	2	1	0
R W	PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0
Reset	0	0	0	0	0	0	0	0

Figure 7-47. Precision Timer Prescaler Select Register (PTPSR)

Read: Anytime

Write: Anytime

All bits reset to zero.

Table 7-31. PTPSR Field Descriptions

Field	Description
7:0 PTPS[7:0]	**Precision Timer Prescaler Select Bits** — These eight bits specify the division rate of the main Timer prescaler. These are effective only when the PRNT bit of TSCR1 is set to 1. Table 7-32 shows some selection examples in this case. The newly selected prescale factor will not take effect until the next synchronized edge where all prescale counter stages equal zero.

Table 7-32. Precision Timer Prescaler Selection Examples when PRNT = 1

PTPS7	PTPS6	PTPS5	PTPS4	PTPS3	PTPS2	PTPS1	PTPS0	Prescale Factor
0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	2
0	0	0	0	0	0	1	0	3
0	0	0	0	0	0	1	1	4
0	0	0	0	0	1	0	0	5
0	0	0	0	0	1	0	1	6
0	0	0	0	0	1	1	0	7

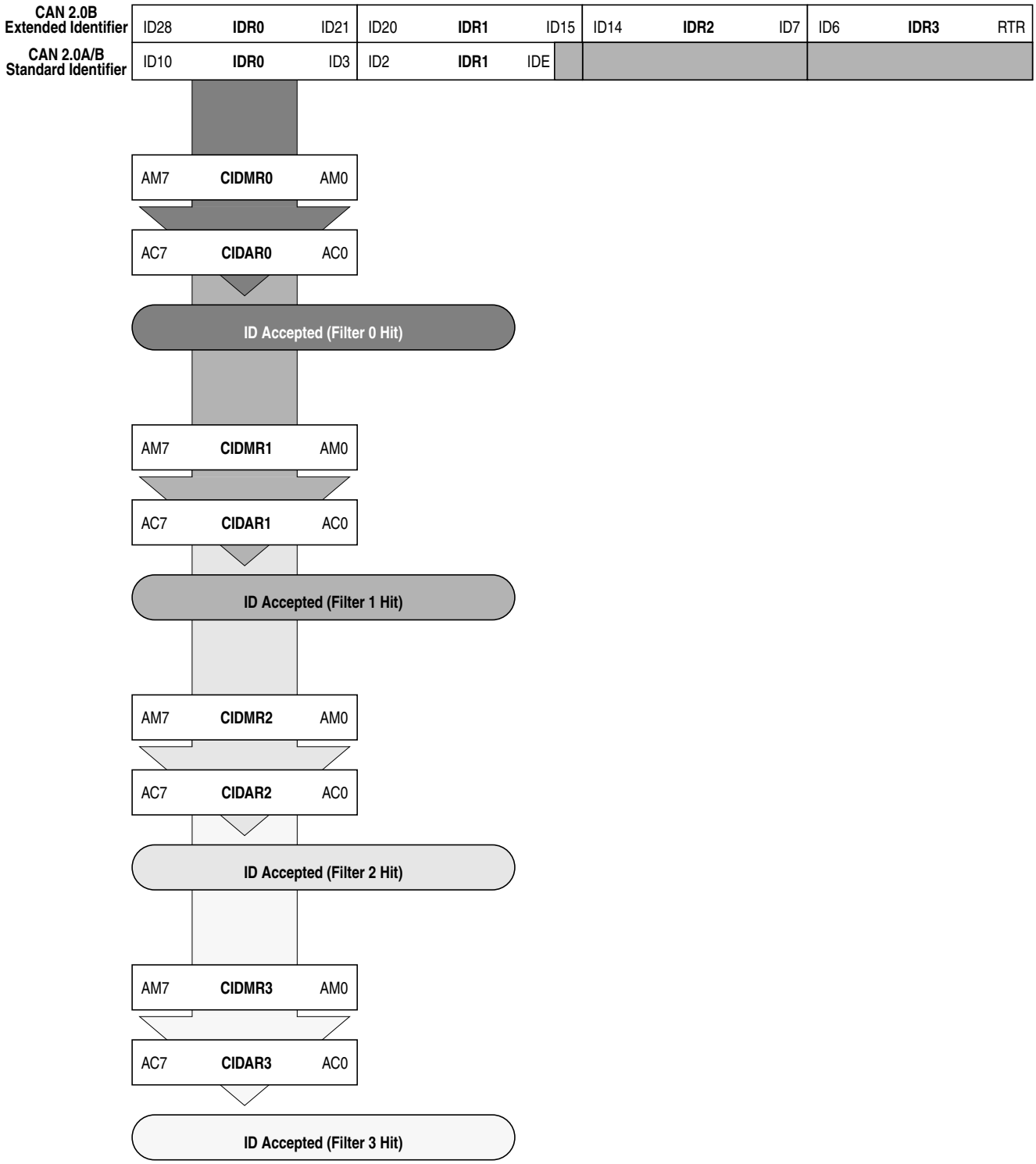

Figure 10-42. 8-bit Maskable Identifier Acceptance Filters

11.4 Functional Description

This section provides a complete functional description of the SCI block, detailing the operation of the design from the end user perspective in a number of subsections.

Figure 11-14 shows the structure of the SCI module. The SCI allows full duplex, asynchronous, serial communication between the CPU and remote devices, including other CPUs. The SCI transmitter and receiver operate independently, although they use the same baud rate generator. The CPU monitors the status of the SCI, writes the data to be transmitted, and processes received data.

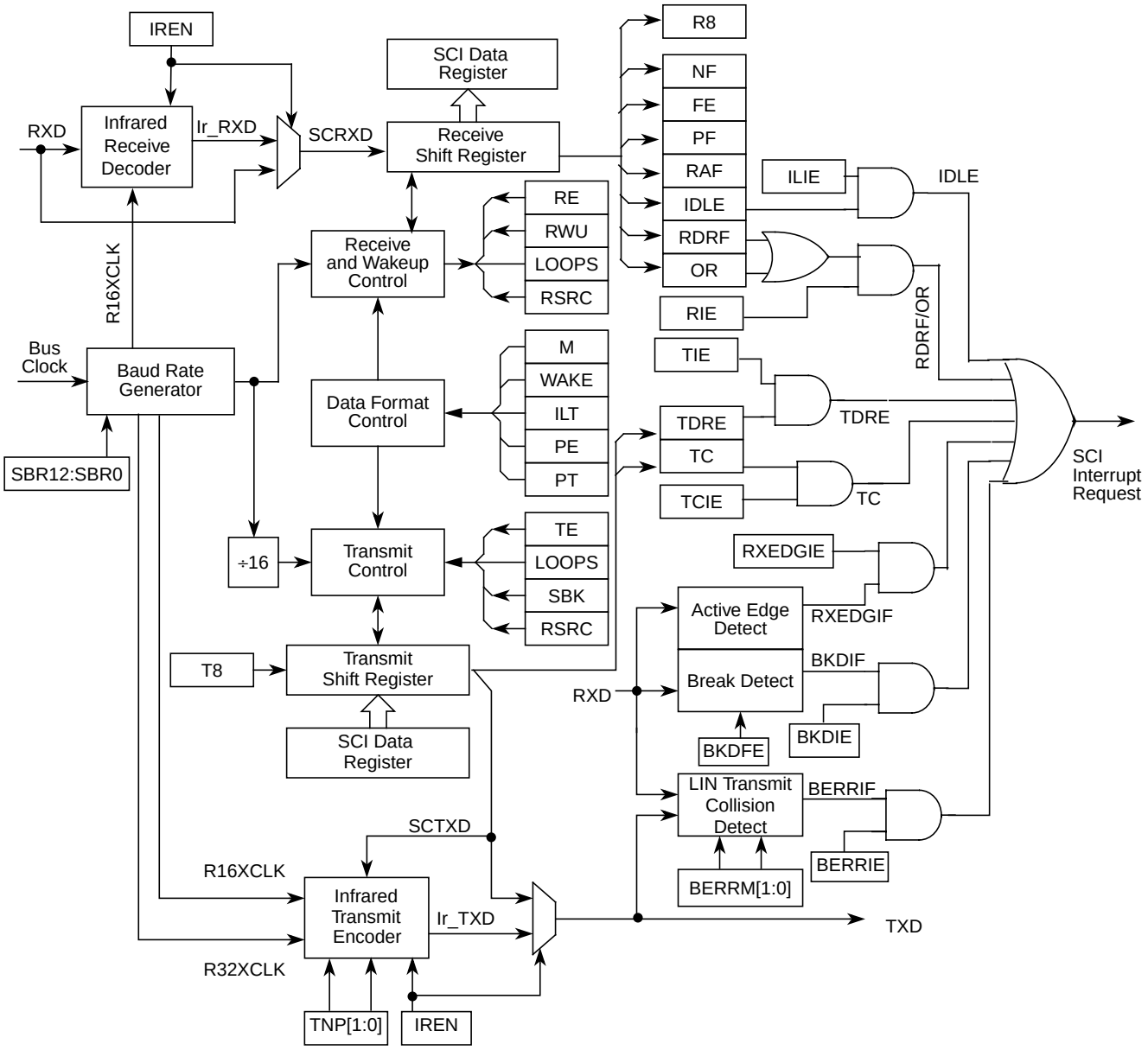

Figure 11-14. Detailed SCI Block Diagram

17.3.2.2 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 1-2).

= Unimplemented or Reserved

Figure 17-4. 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 1-5). 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 17-6. 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 1-5). Write restrictions exist to disallow transitions between certain modes. Figure 1-5 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 0b101 (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).

18.3.2.10 RAM XGATE Upper Boundary Register (RAMXGU)

Address: 0x011D

	7	6	5	4	3	2	1	0
R	1	XGU6	XGU5	XGU4	XGU3	XGU2	XGU1	XGU0
W								
Reset	1	1	1	1	1	1	1	1

= Unimplemented or Reserved

Figure 18-18. RAM XGATE Upper Boundary Register (RAMXGU)

Read: Anytime

Write: Anytime when RWPE = 0

Table 18-16. RAMXGU Field Descriptions

Field	Description
6–0 XGU[6:0]	**XGATE Region Upper Boundary Bits 6-0** — These bits define the upper boundary of the RAM region allocated to the XGATE module in multiples of 256 bytes. The 256 byte block selected by this register is included in the region. See Figure 18-25 for details.

18.3.2.11 RAM Shared Region Lower Boundary Register (RAMSHL)

Address: 0x011E

	7	6	5	4	3	2	1	0
R	1	SHL6	SHL5	SHL4	SHL3	SHL2	SHL1	SHL0
W								
Reset	1	1	1	1	1	1	1	1

= Unimplemented or Reserved

Figure 18-19. RAM Shared Region Lower Boundary Register (RAMSHL)

Read: Anytime

Write: Anytime when RWPE = 0

Table 18-17. RAMSHL Field Descriptions

Field	Description
6–0 SHL[6:0]	**RAM Shared Region Lower Boundary Bits 6–0** — These bits define the lower boundary of the shared memory region in multiples of 256 bytes. The block selected by this register is included in the region. See Figure 18-25 for details.

22.3.2.30 Port M Data Register (PTM)

	7	6	5	4	3	2	1	0
R W	PTM7	PTM6	PTM5	PTM4	PTM3	PTM2	PTM1	PTM0
CAN	TXCAN3	RXCAN3	TXCAN2	RXCAN2	TXCAN1	RXCAN1	TXCAN0	RXCAN0
Routed CAN0			TXCAN0	RXCAN0	TXCAN0	RXCAN0		
Routed CAN4	TXCAN4	RXCAN4	TXCAN4	RXCAN4				
Routed SPIO			SCK0	MOSI0	$\overline{SS0}$	MISO0		
Reset	0	0	0	0	0	0	0	0

Figure 22-32. Port M Data Register (PTM)

Read: Anytime.

Write: Anytime.

Port M pins 75–0 are associated with the CAN0, CAN1, CAN2, CAN3, SCI3, as well as the routed CAN0, CAN4, and SPI0 modules. When not used with any of the peripherals, these pins can be used as general purpose I/O.

If the data direction bits of the associated I/O pins are set to logic level “1”, a read returns the value of the port register, otherwise the buffered pin input state is read.

Table 22-32. PTM Field Descriptions

Field	Description
7–6 PTM[7:6]	The CAN3 function (TXCAN3 and RXCAN3) takes precedence over the CAN4, SCI3 and the general purpose I/O function if the CAN3 module is enabled. *Refer to MSCAN section for details.* The CAN4 function (TXCAN4 and RXCAN4) takes precedence over the SCI3 and the general purpose I/O function if the CAN4 module is enabled. *Refer to MSCAN section for details.* The SCI3 function (TXD3 and RXD3) takes precedence over the general purpose I/O function if the SCI3 module is enabled. *Refer to SCI section for details.*
5–4 PTM[5:4]	The CAN2 function (TXCAN2 and RXCAN2) takes precedence over the routed CAN0, routed CAN4, the routed SPI0 and the general purpose I/O function if the CAN2 module is enabled{pim_9xd_prio.m}. The routed CAN0 function (TXCAN0 and RXCAN0) 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 (TXCAN4 and RXCAN4) takes precedence over the routed SPI0 and general purpose I/O function if the routed CAN4 module is enabled. *Refer to MSCAN section for details.* The routed SPI0 function (SCK0 and MOSI0) takes precedence of the general purpose I/O function if the routed SPI0 is enabled. *Refer to SPI section for details.*

22.4.2.3 Port C and D

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

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

NOTE

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

22.4.2.4 Port E

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

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

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

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

Port E pin PE[1] can be used for either general-purpose input or as the level- or falling edge-sensitive $\overline{IRQ}$ interrupt input. $\overline{IRQ}$ will be enabled by setting the IRQEN configuration bit (Section 22.3.2.14, “IRQ Control Register (IRQCR)”) and clearing the I-bit in the CPU’s condition code register. It is inhibited at reset so this pin is initially configured as a simple input with a pull-up.

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

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

22.4.2.5 Port K

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

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

NOTE

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

Table 25-6. ESTAT Field Descriptions

Field	Description
7 CBEIF	**Command Buffer Empty Interrupt Flag** — The CBEIF flag indicates that the address, data, and command buffers are empty so that a new command write sequence can be started. The CBEIF flag is cleared by writing a 1 to CBEIF. Writing a 0 to the CBEIF flag has no effect on CBEIF. Writing a 0 to CBEIF after writing an aligned word to the EEPROM address space but before CBEIF is cleared will abort a command write sequence and cause the ACCERR flag to be set. Writing a 0 to CBEIF outside of a command write sequence will not set the ACCERR flag. The CBEIF flag is used together with the CBEIE bit in the ECNFG register to generate an interrupt request (see Figure 25-24). 0 Buffers are full. 1 Buffers are ready to accept a new command.
6 CCIF	**Command Complete Interrupt Flag** — The CCIF flag indicates that there are no more commands pending. The CCIF flag is cleared when CBEIF is clear and sets automatically upon completion of all active and pending commands. The CCIF flag does not set when an active commands completes and a pending command is fetched from the command buffer. Writing to the CCIF flag has no effect on CCIF. The CCIF flag is used together with the CCIE bit in the ECNFG register to generate an interrupt request (see Figure 25-24). 0 Command in progress. 1 All commands are completed.
5 PVIOL	**Protection Violation Flag** — The PVIOL flag indicates an attempt was made to program or erase an address in a protected area of the EEPROM memory during a command write sequence. The PVIOL flag is cleared by writing a 1 to PVIOL. Writing a 0 to the PVIOL flag has no effect on PVIOL. While PVIOL is set, it is not possible to launch a command or start a command write sequence. 0 No failure. 1 A protection violation has occurred.
4 ACCERR	**Access Error Flag** — The ACCERR flag indicates an illegal access has occurred to the EEPROM memory caused by either a violation of the command write sequence (see Section 25.4.1.2, “Command Write Sequence”), issuing an illegal EEPROM command (see Table 25-8), launching the sector erase abort command terminating a sector erase operation early (see Section 25.4.2.5, “Sector Erase Abort Command”) or the execution of a CPU STOP instruction while a command is executing (CCIF = 0). The ACCERR flag is cleared by writing a 1 to ACCERR. Writing a 0 to the ACCERR flag has no effect on ACCERR. While ACCERR is set, it is not possible to launch a command or start a command write sequence. If ACCERR is set by an erase verify operation, any buffered command will not launch. 0 No access error detected. 1 Access error has occurred.
2 BLANK	**Flag Indicating the Erase Verify Operation Status** — When the CCIF flag is set after completion of an erase verify command, the BLANK flag indicates the result of the erase verify operation. The BLANK flag is cleared by the EEPROM module when CBEIF is cleared as part of a new valid command write sequence. Writing to the BLANK flag has no effect on BLANK. 0 EEPROM block verified as not erased. 1 EEPROM block verified as erased.
1 FAIL	**Flag Indicating a Failed EEPROM Operation** — The FAIL flag will set if the erase verify operation fails (EEPROM block verified as not erased). The FAIL flag is cleared by writing a 1 to FAIL. Writing a 0 to the FAIL flag has no effect on FAIL. 0 EEPROM operation completed without error. 1 EEPROM operation failed.

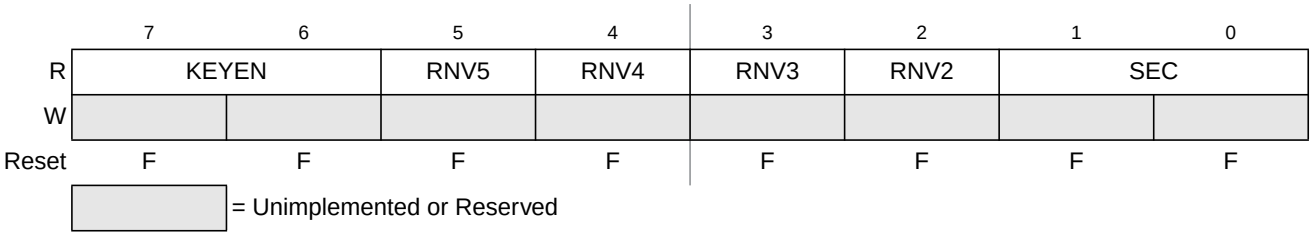

Figure 27-5. Flash Security Register (FSEC)

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

The FSEC register is loaded from the Flash Configuration Field at address 0x7F_FF0F during the reset sequence, indicated by F in Figure 27-5.

Table 27-4. FSEC Field Descriptions

Field	Description
7-6 KEYEN[1:0]	**Backdoor Key Security Enable Bits** — The KEYEN[1:0] bits define the enabling of backdoor key access to the Flash module as shown in Table 27-5.
5-2 RNV[5:2]	**Reserved Nonvolatile Bits** — The RNV[5:2] bits should remain in the erased state for future enhancements.
1-0 SEC[1:0]	**Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 27-6. If the Flash module is unsecured using backdoor key access, the SEC[1:0] bits are forced to 1:0.

Table 27-5. Flash KEYEN States

KEYEN[1:0]	Status of Backdoor Key Access
00	DISABLED
01[1]	DISABLED
10	ENABLED
11	DISABLED

1 Preferred KEYEN state to disable Backdoor Key Access.

Table 27-6. Flash Security States

SEC[1:0]	Status of Security
00	SECURED
01[1]	SECURED
10	UNSECURED
11	SECURED

1 Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Section 27.6, “Flash Module Security”.

27.3.2.3 Flash Test Mode Register (FTSTMOD)

The FTSTMOD register is used to control Flash test features.

27.3.2.5.1 Flash Protection Restrictions

The general guideline is that Flash protection can only be added and not removed. Table 27-15 specifies all valid transitions between Flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register will be ignored and the FPROT register will remain unchanged. The contents of the FPROT register reflect the active protection scenario. See the FPHS and FPLS descriptions for additional restrictions.

Table 27-15. Flash Protection Scenario Transitions

From Protection Scenario	To Protection Scenario[1]							
	0	1	2	3	4	5	6	7
0	X	X	X	X				
1		X		X				
2			X	X				
3				X				
4				X	X			
5			X	X	X	X		
6		X		X	X		X	
7	X	X	X	X	X	X	X	X

1 Allowed transitions marked with X.

27.3.2.6 Flash Status Register (FSTAT)

The FSTAT register defines the operational status of the module.

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

= Unimplemented or Reserved

Figure 27-12. Flash Status Register (FSTAT — Normal Mode)

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

= Unimplemented or Reserved

Figure 27-13. Flash Status Register (FSTAT — Special Mode)

28.4.2.6 Sector Erase Abort Command

The sector erase abort operation will terminate the active sector erase operation so that other sectors in a Flash block are available for read and program operations without waiting for the sector erase operation to complete.

An example flow to execute the sector erase abort operation is shown in Figure 28-31. The sector erase abort command write sequence is as follows:

1. Write to any Flash block address to start the command write sequence for the sector erase abort command. The address and data written are ignored.
2. Write the sector erase abort command, 0x47, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the sector erase abort command.

If the sector erase abort command is launched resulting in the early termination of an active sector erase operation, the ACCERR flag will set once the operation completes as indicated by the CCIF flag being set. The ACCERR flag sets to inform the user that the Flash sector may not be fully erased and a new sector erase command must be launched before programming any location in that specific sector. If the sector erase abort command is launched but the active sector erase operation completes normally, the ACCERR flag will not set upon completion of the operation as indicated by the CCIF flag being set. Therefore, if the ACCERR flag is not set after the sector erase abort command has completed, a Flash sector being erased when the abort command was launched will be fully erased. The maximum number of cycles required to abort a sector erase operation is equal to four FCLK periods (see Section 28.4.1.1, “Writing the FCLKDIV Register”) plus five bus cycles as measured from the time the CBEIF flag is cleared until the CCIF flag is set. If sectors in multiple Flash blocks are being simultaneously erased, the sector erase abort operation will be applied to all active Flash blocks without writing to each Flash block in the sector erase abort command write sequence.

NOTE

Since the ACCERR bit in the FSTAT register may be set at the completion of the sector erase abort operation, a command write sequence is not allowed to be buffered behind a sector erase abort command write sequence. The CBEIF flag will not set after launching the sector erase abort command to indicate that a command should not be buffered behind it. If an attempt is made to start a new command write sequence with a sector erase abort operation active, the ACCERR flag in the FSTAT register will be set. A new command write sequence may be started after clearing the ACCERR flag, if set.

NOTE

The sector erase abort command should be used sparingly since a sector erase operation that is aborted counts as a complete program/erase cycle.