Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	HCS12X
Core Size	16-Bit
Speed	80MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	91
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	14K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12xd256val

BHI Branch if Higher BHI

Operation

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

Branch instruction to compare unsigned numbers.

Branch if RS1 > RS2:

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

CCR Effects

N	Z	V	C
—	—	—	—

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

Code and CPU Cycles

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

SUBH — Subtract Immediate 8 bit Constant (High Byte) — SUBH

Operation

RD – IMM8:$00 ⇒ RD

Subtracts a signed immediate 8 bit constant from the content of high byte of register RD and using binary subtraction and stores the result in the high byte of destination register RD. This instruction can be used after an SUBL for a 16 bit immediate subtraction.

Example:

```
SUBL    R2,#LOWBYTE
SUBH    R2,#HIGHBYTE      ; R2 = R2 - 16 bit immediate
```

CCR Effects

N	Z	V	C
Δ	Δ	Δ	Δ

N: Set if bit 15 of the result is set; cleared otherwise.

Z: Set if the result is $0000; cleared otherwise.

V: Set if a two´s complement overflow resulted from the operation; cleared otherwise.
$RD[15]_{old} \& \overline{IMM8[7]} \& \overline{RD[15]}_{new} \mid \overline{RD[15]}_{old} \& IMM8[7] \& RD[15]_{new}$

C: Set if there is a carry from the bit 15 of the result; cleared otherwise.
$\overline{RD[15]}_{old} \& IMM8[7] \mid \overline{RD[15]}_{old} \& RD[15]_{new} \mid IMM8[7] \& RD[15]_{new}$

Code and CPU Cycles

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

Either left aligned or center aligned output mode can be used in concatenated mode and is controlled by the low order CAEx bit. The high order CAEx bit has no effect.

Table 8-11 is used to summarize which channels are used to set the various control bits when in 16-bit mode.

Table 8-11. 16-bit Concatenation Mode Summary

CONxx	PWMEx	PPOLx	PCLKx	CAEx	PWMx Output
CON67	PWME7	PPOL7	PCLK7	CAE7	PWM7
CON45	PWME5	PPOL5	PCLK5	CAE5	PWM5
CON23	PWME3	PPOL3	PCLK3	CAE3	PWM3
CON01	PWME1	PPOL1	PCLK1	CAE1	PWM1

8.4.2.8 PWM Boundary Cases

Table 8-12 summarizes the boundary conditions for the PWM regardless of the output mode (left aligned or center aligned) and 8-bit (normal) or 16-bit (concatenation).

Table 8-12. PWM Boundary Cases

PWMDTYx	PWMPERx	PPOLx	PWMx Output
$00 (indicates no duty)	>$00	1	Always low
$00 (indicates no duty)	>$00	0	Always high
XX	$00[1] (indicates no period)	1	Always high
XX	$00[1] (indicates no period)	0	Always low
>= PWMPERx	XX	1	Always high
>= PWMPERx	XX	0	Always low

[1] Counter = $00 and does not count.

8.5 Resets

The reset state of each individual bit is listed within the Section 8.3.2, “Register Descriptions” which details the registers and their bit-fields. All special functions or modes which are initialized during or just following reset are described within this section.

- The 8-bit up/down counter is configured as an up counter out of reset.
- All the channels are disabled and all the counters do not count.

11.4.6.6.1 Idle Input line Wakeup (WAKE = 0)

In this wakeup method, an idle condition on the RXD pin clears the RWU bit and wakes up the SCI. The initial frame or frames of every message contain addressing information. All receivers evaluate the addressing information, and receivers for which the message is addressed process the frames that follow. Any receiver for which a message is not addressed can set its RWU bit and return to the standby state. The RWU bit remains set and the receiver remains on standby until another idle character appears on the RXD pin.

Idle line wakeup requires that messages be separated by at least one idle character and that no message contains idle characters.

The idle character that wakes a receiver does not set the receiver idle bit, IDLE, or the receive data register full flag, RDRF.

The idle line type bit, ILT, determines whether the receiver begins counting logic 1s as idle character bits after the start bit or after the stop bit. ILT is in SCI control register 1 (SCICR1).

11.4.6.6.2 Address Mark Wakeup (WAKE = 1)

In this wakeup method, a logic 1 in the most significant bit (MSB) position of a frame clears the RWU bit and wakes up the SCI. The logic 1 in the MSB position marks a frame as an address frame that contains addressing information. All receivers evaluate the addressing information, and the receivers for which the message is addressed process the frames that follow.Any receiver for which a message is not addressed can set its RWU bit and return to the standby state. The RWU bit remains set and the receiver remains on standby until another address frame appears on the RXD pin.

The logic 1 MSB of an address frame clears the receiver's RWU bit before the stop bit is received and sets the RDRF flag.

Address mark wakeup allows messages to contain idle characters but requires that the MSB be reserved for use in address frames.

NOTE

With the WAKE bit clear, setting the RWU bit after the RXD pin has been idle can cause the receiver to wake up immediately.

11.4.7 Single-Wire Operation

Normally, the SCI uses two pins for transmitting and receiving. In single-wire operation, the RXD pin is disconnected from the SCI. The SCI uses the TXD pin for both receiving and transmitting.

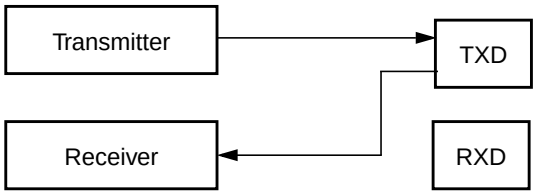

Figure 11-30. Single-Wire Operation (LOOPS = 1, RSRC = 1)

12.4 Functional Description

The SPI module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or SPI operation can be interrupt driven.

The SPI system is enabled by setting the SPI enable (SPE) bit in SPI control register 1. While SPE is set, the four associated SPI port pins are dedicated to the SPI function as:

- Slave select ($\overline{SS}$)
- Serial clock (SCK)
- Master out/slave in (MOSI)
- Master in/slave out (MISO)

The main element of the SPI system is the SPI data register. The 8-bit data register in the master and the 8-bit data register in the slave are linked by the MOSI and MISO pins to form a distributed 16-bit register. When a data transfer operation is performed, this 16-bit register is serially shifted eight bit positions by the S-clock from the master, so data is exchanged between the master and the slave. Data written to the master SPI data register becomes the output data for the slave, and data read from the master SPI data register after a transfer operation is the input data from the slave.

A read of SPISR with SPTEF = 1 followed by a write to SPIDR puts data into the transmit data register. When a transfer is complete and SPIF is cleared, received data is moved into the receive data register. This 8-bit data register acts as the SPI receive data register for reads and as the SPI transmit data register for writes. A single SPI register address is used for reading data from the read data buffer and for writing data to the transmit data register.

The clock phase control bit (CPHA) and a clock polarity control bit (CPOL) in the SPI control register 1 (SPICR1) select one of four possible clock formats to be used by the SPI system. The CPOL bit simply selects a non-inverted or inverted clock. The CPHA bit is used to accommodate two fundamentally different protocols by sampling data on odd numbered SCK edges or on even numbered SCK edges (see Section 12.4.3, “Transmission Formats”).

The SPI can be configured to operate as a master or as a slave. When the MSTR bit in SPI control register1 is set, master mode is selected, when the MSTR bit is clear, slave mode is selected.

NOTE

A change of CPOL or MSTR bit while there is a received byte pending in the receive shift register will destroy the received byte and must be avoided.

Table 18-11. 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).

18.3.2.6 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 18-11. 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 18-12). This supports accessing up to 1022 Kbytes 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.

19.4.7.5 DBG Breakpoint Priorities

XGATE software breakpoints have the highest priority. Active tracing sessions are terminated immediately.

If a TRIG triggers occur after begin or mid aligned tracing has already been triggered by a comparator instigated transition to final state, then TRIG no longer has an effect. When the associated tracing session is complete, the breakpoint occurs. Similarly if a TRIG is followed by a subsequent trigger from a comparator channel whose BRK=0, it has no effect, since tracing has already started.

If a comparator tag hit occurs simultaneously with an external $\overline{\text{TAGHI}}/\overline{\text{TAGLO}}$ hit, the state sequencer enters State0. $\overline{\text{TAGHI}}/\overline{\text{TAGLO}}$ triggers are always end aligned, to end tracing immediately, independent of the tracing trigger alignment bits TALIGN[1:0].

19.4.7.5.1 DBG Breakpoint priorities, mapping and BDM interfacing

Breakpoint operation is dependent on the state of the BDM module. If the BDM module is active, the CPU is executing out of BDM firmware and S12X breakpoints are disabled. In addition, while executing a BDM TRACE command, tagging into BDM is disabled.

Table 19-45. Breakpoint Mapping Summary

DBGBRK[1] (DBGC1[3])[1]	BDM bit (DBGC1[4])	BDM enabled	BDM active	Type of Debug Session
0	X	X	X	No Breakpoint
1	0	0	0	Breakpoint to SWI
1	0	1	X	Illegal Configuration. Do Not Use.[2]
1	1	0	0	Illegal Configuration. Do Not Use. [3]
1	1	1	0	Breakpoint to BDM
1	1	1	1	No Breakpoint

[1] All sources except XGATE software BKP, which are independent of this bit.
[2]The DBGC1[4] bit (BDM) must be set if using the BDM interface together with the DBG module. Failure to set this bit could result in XGATE generated breakpoints to SWI during BDM firmware execution corrupting the S12X PC return address, should the user have entered BDM via the BACKGROUND command or BGND instruction.
[3]End aligned tagged Breakpoint to SWI. Begin, Mid aligned and Forced Breakpoints disabled

If BDM is not active, the breakpoint will give priority to BDM requests over SWI requests if the breakpoint happens to coincide with a SWI instruction in the user's code. On returning from BDM, the SWI from user code gets executed.

BDM cannot be entered from a breakpoint unless the ENABLE bit is set in the BDM. If entry to BDM via a BGND instruction is attempted and the ENABLE bit in the BDM is cleared, the CPU actually executes the BDM firmware code. It checks the ENABLE and returns if ENABLE is not set. If not serviced by the monitor then the breakpoint is re-asserted when the BDM returns to normal CPU flow.

If the comparator register contents coincide with the SWI/BDM vector address then an SWI in user code and DBG breakpoint could occur simultaneously. The CPU ensures that BDM requests have a higher priority than SWI requests. Returning from the BDM/SWI service routine care must be taken to avoid re-triggering a breakpoint.

Table 21-3. 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]						
	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 EWAITE = 1	Standard	Reduced if EWAITE = 1	Standard

Table 21-4. External Address Bus Size

ASIZ[4:0]	Available External Address Lines
00000	None
00001	$\overline{\text{UDS}}$
00010	ADDR1, $\overline{\text{UDS}}$
00011	ADDR[2:1], $\overline{\text{UDS}}$
:	:
10110	ADDR[21:1], $\overline{\text{UDS}}$
10111 : 11111	ADDR[22:1], $\overline{\text{UDS}}$

21.4.2.2.2 Write Access Timing

Table 21-12. 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]	...		**acc 0**		acc 1		acc 2	...
ADDR[19:16] / IQSTAT[3:0]	...	**addr 0**	iqstat -1	addr 1	**iqstat 0**	addr 2	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 21-13. 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]	...		**acc 0**		**000**		acc 1	...
ADDR[19:16] / IQSTAT[3:0]	...	**addr 0**	iqstat-1	**addr 0**	**iqstat 0**	addr 1	**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 21-14. 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]	...		**acc 0**		**000**		**000**	**...**		acc 1	...
ADDR[19:16] / IQSTAT[3:0]	...	**addr 0**	iqstat-1	**addr 0**	**iqstat 0**	**addr 0**	**0000**	**...**	addr 1	**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	...

21.5.2 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).

22.3.2.33 Port M Reduced Drive Register (RDRM)

	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 22-35. Port M Reduced Drive Register (RDRM)

Read: Anytime.

Write: Anytime.

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

Table 22-34. RDRM Field Descriptions

Field	Description
7–0 RDRM[7:0]	**Reduced Drive Port M** 0 Full drive strength at output. 1 Associated pin drives at about 1/6 of the full drive strength.

22.3.2.34 Port M Pull Device Enable Register (PERM)

	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 22-36. Port M Pull Device Enable Register (PERM)

Read: Anytime.

Write: Anytime.

This register configures whether a pull-up or a pull-down device is activated, if the port is used as input or wired-OR output. This bit has no effect if the port is used as push-pull output. Out of reset no pull device is enabled.

Table 22-35. PERM Field Descriptions

Field	Description
7–0 PERM[7:0]	**Pull Device Enable Port M** 0 Pull-up or pull-down device is disabled. 1 Either a pull-up or pull-down device is enabled.

Table 23-42. PPSP Field Descriptions

Field	Description
7–0 PPSP[7:0]	**Polarity Select Port P** 0 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. 1 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.

23.0.5.44 Port P Interrupt Enable Register (PIEP)

	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 23-46. Port P Interrupt Enable Register (PIEP)

Read: Anytime.

Write: Anytime.

This register disables or enables on a per-pin basis the edge sensitive external interrupt associated with Port P.

Table 23-43. PIEP Field Descriptions

Field	Description
7–0 PIEP[7:0]	**Interrupt Enable Port P** 0 Interrupt is disabled (interrupt flag masked). 1 Interrupt is enabled.

23.0.5.45 Port P Interrupt Flag Register (PIFP)

	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 23-47. Port P Interrupt Flag Register (PIFP)

Read: Anytime.

Write: Anytime.

23.0.5.71 Port AD1 Reduced Drive Register 1 (RDR1AD1)

	7	6	5	4	3	2	1	0
R W	RDR1AD115	RDR1AD114	RDR1AD113	RDR1AD112	RDR1AD111	RDR1AD110	RDR1AD19	RDR1AD18
Reset	0	0	0	0	0	0	0	0

Figure 23-73. Port AD1 Reduced Drive Register 1 (RDR1AD1)

Read: Anytime.

Write: Anytime.

This register configures the drive strength of each PAD[15:8] output pin as either full or reduced. If the port is used as input this bit is ignored.

Table 23-64. RDR1AD1 Field Descriptions

Field	Description
7–0 RDR1AD1[15:8]	**Reduced Drive Port AD1 Register 1** 0 Full drive strength at output. 1 Associated pin drives at about 1/6 of the full drive strength.

23.0.5.72 Port AD1 Pull Up Enable Register 0 (PER0AD1)

	7	6	5	4	3	2	1	0
R W	PER0AD123	PER0AD122	PER0AD121	PER0AD120	PER0AD119	PER0AD118	PER0AD117	PER0AD116
Reset	0	0	0	0	0	0	0	0

Figure 23-74. Port AD1 Pull Up Enable Register 0 (PER0AD1)

Read: Anytime.

Write: Anytime.

This register activates a pull-up device on the respective PAD[23:16] pin if the port is used as input. This bit has no effect if the port is used as output. Out of reset no pull-up device is enabled.

Table 23-65. PER0AD1 Field Descriptions

Field	Description
7–0 PER0AD1[23:16]	**Pull Device Enable Port AD1 Register 0** 0 Pull-up device is disabled. 1 Pull-up device is enabled.

24.0.5.8 S12X_EBI Ports Reduced Drive Register (RDRIV)

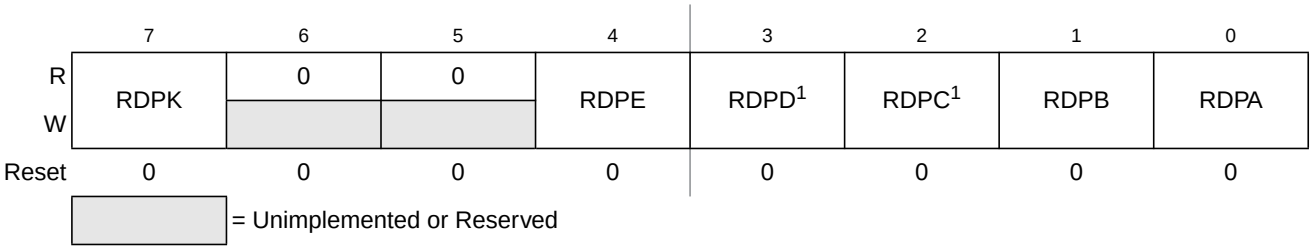

Figure 24-10. S12X_EBI Ports Reduced Drive Register (RDRIV)

1. Register implemented, function disabled: Written values can be read back.

Read: Anytime.

Write: Anytime.

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

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

Table 24-11. RDRIV Field Descriptions

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

24.0.7.9 Port H

This port is associated with the SPI1, . Port H pins PH[7:0] can be used for either general purpose I/O, or with the SPI and SCI subsystems. Port H pins can be used with the routed SPI1. *Refer to Section 24.0.5.33, "Module Routing Register (MODRR)"*.

Port H offers 8 I/O pins with edge triggered interrupt capability (Section 24.0.8, "Pin Interrupts").

NOTE

Port H is not available in 80-pin packages.

24.0.7.10 Port J

This port is associated with CAN4, CAN0, IIC0. Port J pins PJ[7:4] and PJ[2:0] can be used for either general purpose I/O, or with the CAN, IIC, or SCI subsystems. If IIC takes precedence the associated pins become IIC open-drain output pins. The CAN4 pins can be re-routed. *Refer to Section 24.0.5.33, "Module Routing Register (MODRR)"*.

Port J pins can be used with the routed CAN0 modules. *Refer to Section 24.0.5.33, "Module Routing Register (MODRR)"*.

Port J offers 7 I/O pins with edge triggered interrupt capability (Section 24.0.8, "Pin Interrupts").

NOTE

PJ[5,4,2,1,0] are not available in 80-pin packages.

24.0.7.11 Port AD1

This port is associated with the ATD1. Port AD1 pins PAD15–PAD0 can be used for either general purpose I/O, or with the ATD1 subsystem.

NOTE

PAD[15:8] are not available in 80-pin packages.

24.0.8 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 24-70) shorter than a specified time from generating an interrupt. The minimum time varies over process conditions, temperature and voltage (Figure 24-69 and Table 24-62).

The EADDRHI and EADDRLO registers are the EEPROM address registers.

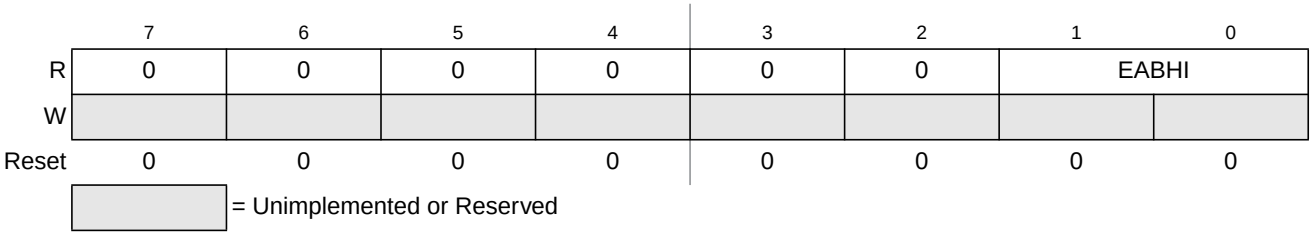

Figure 25-13. EEPROM Address High Register (EADDRHI)

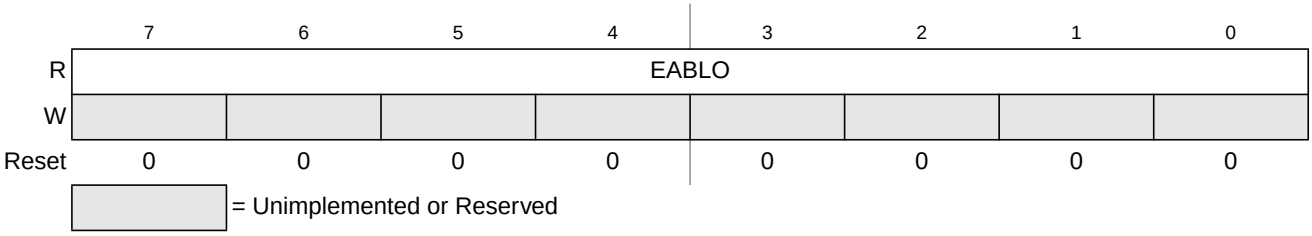

Figure 25-14. EEPROM Address Low Register (EADDRLO)

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

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

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

25.3.2.9 EEPROM Data Registers (EDATA)

The EDATAHI and EDATALO registers are the EEPROM data registers.

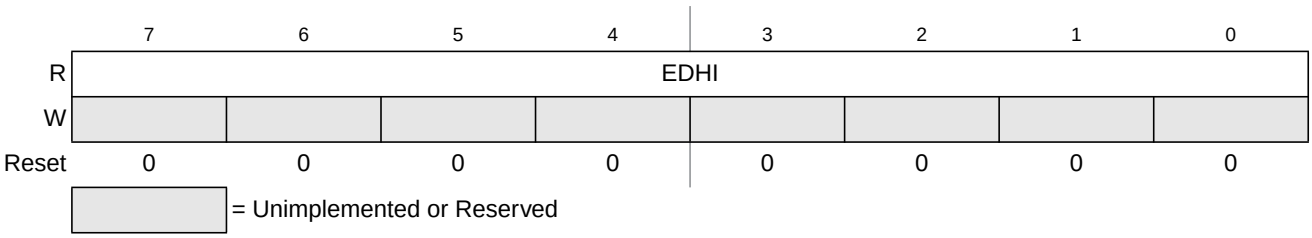

Figure 25-15. EEPROM Data High Register (EDATAHI)

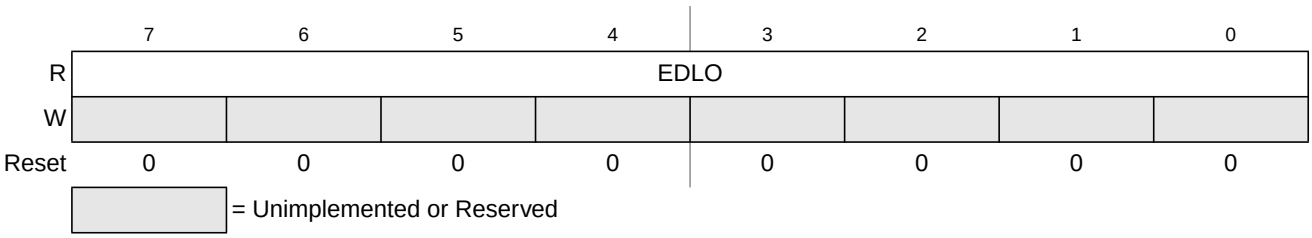

Figure 25-16. EEPROM Data Low Register (EDATALO)

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

unaffected by the backdoor key access sequence. After the next reset of the MCU, the security state of the Flash module is determined by the Flash security byte (0x7F_FF0F). The backdoor key access sequence has no effect on the program and erase protections defined in the Flash protection register.

It is not possible to unsecure the MCU in special single chip mode by using the backdoor key access sequence in background debug mode (BDM).

28.6.2 Unsecuring the MCU in Special Single Chip Mode using BDM

The MCU can be unsecured in special single chip mode by erasing the Flash module by the following method:

- Reset the MCU into special single chip mode, delay while the erase test is performed by the BDM secure ROM, send BDM commands to disable protection in the Flash module, and execute a mass erase command write sequence to erase the Flash memory.

After the CCIF flag sets to indicate that the mass operation has completed, reset the MCU into special single chip mode. The BDM secure ROM will verify that the Flash memory is erased and will assert the UNSEC bit in the BDM status register. This BDM action will cause the MCU to override the Flash security state and the MCU will be unsecured. All BDM commands will be enabled and the Flash security byte may be programmed to the unsecure state by the following method:

- Send BDM commands to execute a word program sequence to program the Flash security byte to the unsecured state and reset the MCU.

28.7 Resets

28.7.1 Flash Reset Sequence

On each reset, the Flash module executes a reset sequence to hold CPU activity while loading the following registers from the Flash memory according to Table 28-1:

- FPROT — Flash Protection Register (see Section 28.3.2.5).
- FCTL - Flash Control Register (see Section 28.3.2.8).
- FSEC — Flash Security Register (see Section 28.3.2.2).

28.7.2 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.

28.8 Interrupts

The Flash module can generate an interrupt when all Flash command operations have completed, when the Flash address, data and command buffers are empty.

E.5 Peripheral Sets S12XD - Family

Table E-5. S12XD Peripherals

Device	Package	XGATE	CAN	SCI	SPI	IIC	ECT	PIT	ATD0	ATD1	I/O
9S12XDP512	144LQFP	yes	5	6	3	2	8	4	8ch[1]	16ch[2]	119
	112LQFP		5	6	3	1	8	4	8ch[1]	8ch[3]	91
9S12XDT512	144LQFP		3	6	3	1	8	4	8ch[1]	16ch[2]	119
	112LQFP		3	6	3	1	8	4	8ch[1]	8ch[3]	91
	80QFP		3	2	3	1	8	4	8ch[1]		59
9S12XDT384	144LQFP		3	4	3	1	8	4	8ch[1]	16ch[2]	119
	112LQFP		3	4	3	1	8	4	8ch[1]	8ch[3]	91
	80QFP		3	2	3	1	8	4	8ch[1]		59
9S12XDQ256	144LQFP		4	4	3	1	8	4	8ch[1]	16ch[2]	119
	112LQFP		4	4	3	1	8	4	8ch[1]	8ch[3]	91
	80QFP		4	2	3	1	8	4	8ch[1]		59
9S12XDT256	144LQFP		3	4	3	1	8	4	8ch[1]	16ch[2]	119
	112LQFP		3	4	3	1	8	4	8ch[1]	8ch[3]	91
	80QFP		3	2	3	1	8	4	8ch[1]		59
9S12XD256	144LQFP		1	4	3	1	8	4	8ch[1]	16ch[2]	119
	112LQFP		1	4	3	1	8	4	8ch[1]	8ch[3]	91
	80QFP		1	2	3	1	8	4	8ch[1]		59
3S12XDT256	144LQFP		3	4	3	1	8	4	8ch[1]	16ch[2]	119
	112LQFP		3	4	3	1	8	4	8ch[1]	8ch[3]	91
	80QFP		3	2	3	1	8	4	8ch[1]		59
9S12XDG128	112LQFP	yes but XGATE has no Flash Access	2	2	2	1	8	4		16ch[4]	91
	80QFP		2	2	2	1	8	4		8ch[5]	59
3S12XDG128	112LQFP		2	2	2	1	8	4		16ch[4]	91
	80QFP		2	2	2	1	8	4		8ch[5]	59
9S12XD128	112LQFP		1	2	2	1	8	4		16ch[4]	91
	80QFP		1	2	2	1	8	4		8ch[5]	59
9S12XD64	80QFP		1	2	2	1	8	2		8ch[5]	59

1 ATD0 routed to PAD[7:0]

0x0138–0x013F Asynchronous Serial Interface (SCI5) Map (continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x013D	SCI5SR2	R	AMAP	0	0	TXPOL	RXPOL	BRK13	TXDIR	RAF
		W								
0x013E	SCI5DRH	R	R8	T8	0	0	0	0	0	0
		W								
0x013F	SCI5DRL	R	R7	R6	R5	R4	R3	R2	R1	R0
		W	T7	T6	T5	T4	T3	T2	T1	T0

1 Those registers are accessible if the AMAP bit in the SCI5SR2 register is set to zero

2 Those registers are accessible if the AMAP bit in the SCI5SR2 register is set to one

0x0140–0x017F Freescale Scalable CAN — MSCAN (CAN0) Map

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0140	CAN0CTL0	R	RXFRM	RXACT	CSWAI	SYNCH	TIME	WUPE	SLPRQ	INITRQ
		W								
0x0141	CAN0CTL1	R	CANE	CLKSRC	LOOPB	LISTEN	BORM	WUPM	SLPAK	INITAK
		W								
0x0142	CAN0BTR0	R	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
		W								
0x0143	CAN0BTR1	R	SAMP	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
		W								
0x0144	CAN0RFLG	R	WUPIF	CSCIF	RSTAT1	RSTAT0	TSTAT1	TSTAT0	OVRIF	RXF
		W								
0x0145	CAN0RIER	R	WUPIE	CSCIE	RSTATE1	RSTATE0	TSTATE1	TSTATE0	OVRIE	RXFIE
		W								
0x0146	CAN0TFLG	R	0	0	0	0	0	TXE2	TXE1	TXE0
		W								
0x0147	CAN0TIER	R	0	0	0	0	0	TXEIE2	TXEIE1	TXEIE0
		W								
0x0148	CAN0TARQ	R	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
		W								
0x0149	CAN0TAAK	R	0	0	0	0	0	ABTAK2	ABTAK1	ABTAK0
		W								
0x014A	CAN0TBSEL	R	0	0	0	0	0	TX2	TX1	TX0
		W								
0x014B	CAN0IDAC	R	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
		W								
0x014C	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x014D	CAN0MISC	R	0	0	0	0	0	0	0	BOHOLD
		W								
0x014E	CAN0RXERR	R	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
		W								
0x014F	CAN0TXERR	R	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
		W								