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	50MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	91
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12xeq512mal

	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

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.

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).

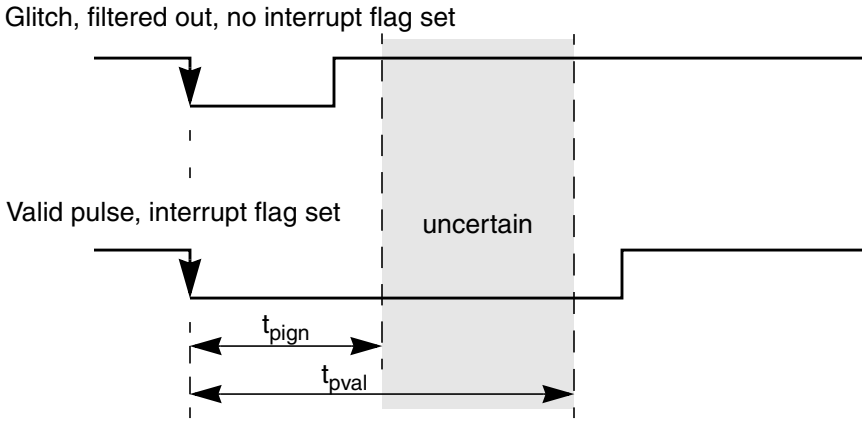


Figure 2-108. Interrupt Glitch Filter on Port P, H and J (PPS=0)

Table 2-104. Pulse Detection Criteria

Pulse	Mode		
	$\overline{STOP}$		STOP(1)
		Unit	
Ignored	$t_{pulse} \leq 3$	bus clocks	$t_{pulse} \leq t_{pign}$
Uncertain	$3 < t_{pulse} < 4$	bus clocks	$t_{pign} < t_{pulse} < t_{pval}$
Valid	$t_{pulse} \geq 4$	bus clocks	$t_{pulse} \geq t_{pval}$

1. These values include the spread of the oscillator frequency over temperature, voltage and process.

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

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.

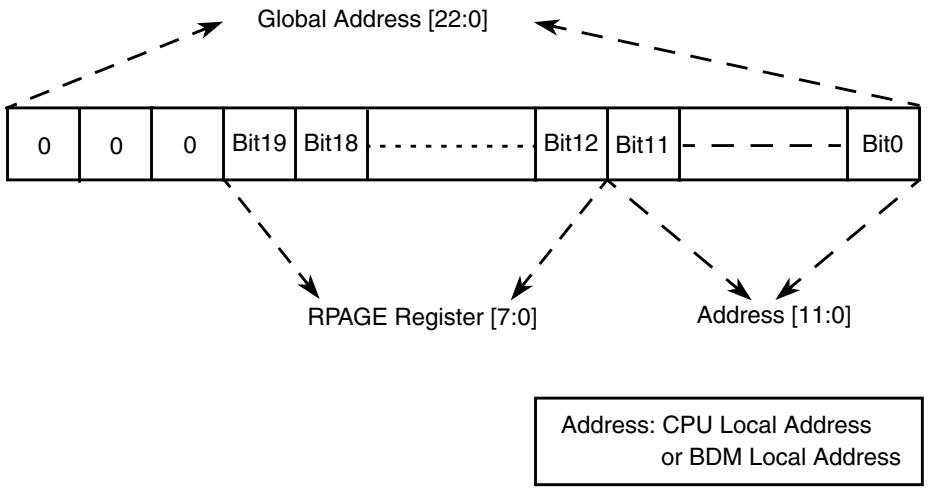


Figure 3-14. RPAGE Address Mapping

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.

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.

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]	...		**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 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]	...		**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 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]	...		**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	...

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**	...

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.

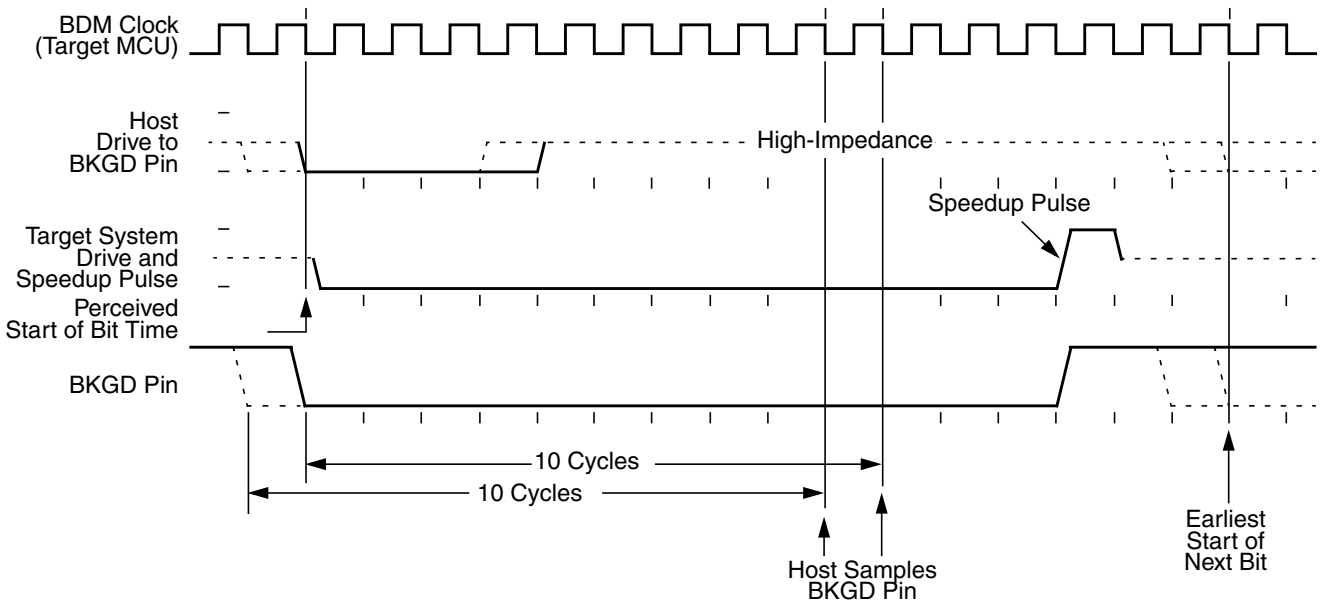


Figure 7-10. BDM Target-to-Host Serial Bit Timing (Logic 0)

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

Writing a "1" to XGDBG and XGDBGM in the same write access causes the XGATE to enter debug mode upon completion of the current instruction.

NOTE

After writing to the XGDBG bit the XGATE will not immediately enter debug mode. Depending on the instruction that is executed at this time there may be a delay of several clock cycles. The XGDBG will read "0" until debug mode is entered.

2. Software breakpoints

   XGATE programs which are stored in the internal RAM allow the use of software breakpoints. A software breakpoint is set by replacing an instruction of the program code with the "BRK" instruction.

   As soon as the program execution reaches the "BRK" instruction, the XGATE enters debug mode. Additionally a software breakpoint request is sent to the S12X_DBG module (see section 4.9 of the **S12X_DBG Section**).

   Upon entering debug mode, the program counter will point to the "BRK" instruction. The other RISC core registers will hold the result of the previous instruction.

   To resume program execution, the "BRK" instruction must be replaced by the original instruction before leaving debug mode.

3. Tagged Breakpoints

   The S12X_DBG module is able to place tags on fetched opcodes. The XGATE is able to enter debug mode right before a tagged opcode is executed (see section 4.9 of the **S12X_DBG Section**). Upon entering debug mode, the program counter will point to the tagged instruction. The other RISC core registers will hold the result of the previous instruction.

4. Forced Breakpoints

   Forced breakpoints are triggered by the S12X_DBG module (see section 4.9 of the **S12X_DBG Section**). When a forced breakpoint occurs, the XGATE will enter debug mode upon completion of the current instruction.

10.6.2 Leaving Debug Mode

Debug mode can only be left by setting the XGDBG bit to "0". If a thread is active (XGCHID has not been cleared in debug mode), program execution will resume at the value of XGPC.

10.7 Security

In order to protect XGATE application code on secured S12X devices, a few restrictions in the debug features have been made. These are:

- Registers XGCCR, XGPC, and XGR1–XGR7 will read zero on a secured device
- Registers XGCCR, XGPC, and XGR1–XGR7 can not be written on a secured device
- Single stepping is not possible on a secured device

10.8.1.3 Immediate 3-Bit Wide (IMM3)

Operands for immediate mode instructions are included in the instruction stream and are fetched into the instruction queue along with the rest of the 16 bit instruction. The '#' symbol is used to indicate an immediate addressing mode operand. This address mode is used for semaphore instructions.

Examples:

```
CSEM    #1      ; Unlock semaphore 1
SSEM    #3      ; Lock Semaphore 3
```

10.8.1.4 Immediate 4 Bit Wide (IMM4)

The 4 bit wide immediate addressing mode is supported by all shift instructions.

RD = RD ∗ IMM4

Examples:

```
LSL     R4,#1   ; R4 = R4 << 1; shift register R4 by 1 bit to the left
LSR     R4,#3   ; R4 = R4 >> 3; shift register R4 by 3 bits to the right
```

10.8.1.5 Immediate 8 Bit Wide (IMM8)

The 8 bit wide immediate addressing mode is supported by four major commands (ADD, SUB, LD, CMP).

RD = RD ∗ imm8

Examples:

```
ADDL    R1,#1   ; adds an 8 bit value to register R1
SUBL    R2,#2   ; subtracts an 8 bit value from register R2
LDH     R3,#3   ; loads an 8 bit immediate into the high byte of Register R3
CMPL    R4,#4   ; compares the low byte of register R4 with an immediate value
```

10.8.1.6 Immediate 16 Bit Wide (IMM16)

The 16 bit wide immediate addressing mode is a construct to simplify assembler code. Instructions which offer this mode are translated into two opcodes using the eight bit wide immediate addressing mode.

RD = RD ∗ IMM16

Examples:

```
LDW     R4,#$1234       ; translated to LDL R4,#$34; LDH R4,#$12
ADD     R4,#$5678       ; translated to ADDL R4,#$78; ADDH R4,#$56
```

10.8.1.7 Monadic Addressing (MON)

In this addressing mode only one operand is explicitly given. This operand can either be the source (*f*(RD)), the target (RD = *f*()), or both source and target of the operation (RD = *f*(RD)).

Examples:

```
JAL     R1      ; PC = R1, R1 = PC+2
SIF     R2      ; Trigger IRQ associated with the channel number in R2.L
```

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x001C TC6 (High)	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
0x001D TC6 (Low)	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
0x001E TC7 (High)	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
0x001F TC7 (Low)	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
0x0020 PACTL	R	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PA0VI	PAI
	W								
0x0021 PAFLG	R	0	0	0	0	0	0	PA0VF	PAIF
	W								
0x0022 PACN3	R	PACNT7(15)	PACNT6(14)	PACNT5(13)	PACNT4(12)	PACNT3(11)	PACNT2(10)	PACNT1(9)	PACNT0(8)
	W								
0x0023 PACN2	R	PACNT7	PACNT6	PACNT5	PACNT4	PACNT3	PACNT2	PACNT1	PACNT0
	W								
0x0024 PACN1	R	PACNT7(15)	PACNT6(14)	PACNT5(13)	PACNT4(12)	PACNT3(11)	PACNT2(10)	PACNT1(9)	PACNT0(8)
	W								
0x0025 PACN0	R	PACNT7	PACNT6	PACNT5	PACNT4	PACNT3	PACNT2	PACNT1	PACNT0
	W								
0x0026 MCCTL	R	MCZI	MODMC	RDMCL	0	0	MCEN	MCPR1	MCPR0
	W				ICLAT	FLMC			
0x0027 MCFLG	R	MCZF	0	0	0	POLF3	POLF2	POLF1	POLF0
	W								
0x0028 ICPAR	R	0	0	0	0	PA3EN	PA2EN	PA1EN	PA0EN
	W								
0x0029 DLYCT	R	DLY7	DLY6	DLY5	DLY4	DLY3	DLY2	DLY1	DLY0
	W								
0x002A ICOVW	R	NOVW7	NOVW6	NOVW5	NOVW4	NOVW3	NOVW2	NOVW1	NOVW0
	W								

= Unimplemented or Reserved

Figure 14-2. ECT Register Summary (Sheet 3 of 5)

Module Base + 0x0006 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	TXE2	TXE1	TXE0
W								
Reset:	0	0	0	0	0	1	1	1

= Unimplemented

Figure 16-10. MSCAN Transmitter Flag Register (CANTFLG)

1. Read: Anytime
   Write: Anytime when not in initialization mode; write of 1 clears flag, write of 0 is ignored

NOTE

The CANTFLG register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 16-13. CANTFLG Register Field Descriptions

Field	Description
2-0 TXE[2:0]	**Transmitter Buffer Empty** — This flag indicates that the associated transmit message buffer is empty, and thus not scheduled for transmission. The CPU must clear the flag after a message is set up in the transmit buffer and is due for transmission. The MSCAN sets the flag after the message is sent successfully. The flag is also set by the MSCAN when the transmission request is successfully aborted due to a pending abort request (see Section 16.3.2.9, "MSCAN Transmitter Message Abort Request Register (CANTARQ)"). If not masked, a transmit interrupt is pending while this flag is set. Clearing a TXEx flag also clears the corresponding ABTAKx (see Section 16.3.2.10, "MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)"). When a TXEx flag is set, the corresponding ABTRQx bit is cleared (see Section 16.3.2.9, "MSCAN Transmitter Message Abort Request Register (CANTARQ)"). When listen-mode is active (see Section 16.3.2.2, "MSCAN Control Register 1 (CANCTL1)") the TXEx flags cannot be cleared and no transmission is started. Read and write accesses to the transmit buffer will be blocked, if the corresponding TXEx bit is cleared (TXEx = 0) and the buffer is scheduled for transmission. 0 The associated message buffer is full (loaded with a message due for transmission) 1 The associated message buffer is empty (not scheduled)

16.3.2.8 MSCAN Transmitter Interrupt Enable Register (CANTIER)

This register contains the interrupt enable bits for the transmit buffer empty interrupt flags.

Module Base + 0x0007 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	TXEIE2	TXEIE1	TXEIE0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 16-11. MSCAN Transmitter Interrupt Enable Register (CANTIER)

3. Shutdown mode

   Controlled by VREGEN (see device level specification for connectivity of VREGEN).

   This mode is characterized by minimum power consumption. The regulator outputs are in a high-impedance state, only the POR feature is available, LVD, LVR and HTD are disabled. The API internal RC oscillator clock is not available.

   This mode must be used to disable the chip internal regulator VREG_3V3, i.e., to bypass the VREG_3V3 to use external supplies.

23.1.3 Block Diagram

Figure 23-1 shows the function principle of VREG_3V3 by means of a block diagram. The regulator core REG consists of three parallel subblocks, REG1, REG2 and REG3, providing three independent output voltages.

24.3 Memory Map and Registers

This section describes the memory map and registers for the Flash module. Read data from unimplemented memory space in the Flash module is undefined. Write access to unimplemented or reserved memory space in the Flash module will be ignored by the Flash module.

24.3.1 Module Memory Map

The S12X architecture places the P-Flash memory between global addresses 0x78_0000 and 0x7F_FFFF as shown in Table 24-2. The P-Flash memory map is shown in Figure 24-2.

Table 24-2. P-Flash Memory Addressing

Global Address	Size (Bytes)	Description
0x7F_0000 – 0x7F_FFFF	64 K	P-Flash Block 0 Contains Flash Configuration Field (see Table 24-3)
0x79_0000 – 0x7E_FFFF	384 K	No P-Flash Memory
0x78_0000 – 0x78_FFFF	64 K	P-Flash Block 1

The FPROT register, described in Section 24.3.2.9, can be set to protect regions in the Flash memory from accidental program or erase. Three separate memory regions, one growing upward from global address 0x7F_8000 in the Flash memory (called the lower region), one growing downward from global address 0x7F_FFFF in the Flash memory (called the higher region), and the remaining addresses in the Flash memory, can be activated for protection. The Flash memory addresses covered by these protectable regions are shown in the P-Flash memory map. The higher address region is mainly targeted to hold the boot loader code since it covers the vector space. Default protection settings as well as security information that allows the MCU to restrict access to the Flash module are stored in the Flash configuration field as described in Table 24-3.

Table 24-3. Flash Configuration Field(1)

Global Address	Size (Bytes)	Description
0x7F_FF00 – 0x7F_FF07	8	Backdoor Comparison Key Refer to Section 24.4.2.11, “Verify Backdoor Access Key Command,” and Section 24.5.1, “Unsecuring the MCU using Backdoor Key Access”
0x7F_FF08 – 0x7F_FF0B(2)	4	Reserved
0x7F_FF0C[2]	1	P-Flash Protection byte. Refer to Section 24.3.2.9, “P-Flash Protection Register (FPROT)”
0x7F_FF0D[2]	1	EEE Protection byte Refer to Section 24.3.2.10, “EEE Protection Register (EPROT)”
0x7F_FF0E[2]	1	Flash Nonvolatile byte Refer to Section 24.3.2.14, “Flash Option Register (FOPT)”

Table 27-12. Flash Security States

SEC[1:0]	Status of Security
00	SECURED
01	SECURED(1)
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 27.5.

27.3.2.3 Flash CCOB Index Register (FCCOBIX)

The FCCOBIX register is used to index the FCCOB register for Flash memory operations.

Offset Module Base + 0x0002

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

= Unimplemented or Reserved

Figure 27-7. FCCOB Index Register (FCCOBIX)

CCOBIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 27-13. FCCOBIX Field Descriptions

Field	Description
2–0 CCOBIX[1:0]	**Common Command Register Index**— The CCOBIX bits are used to select which word of the FCCOB register array is being read or written to. See Section 27.3.2.11, "Flash Common Command Object Register (FCCOB)," for more details.

27.3.2.4 Flash ECCR Index Register (FECCRIX)

The FECCRIX register is used to index the FECCR register for ECC fault reporting.

Offset Module Base + 0x0003

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

= Unimplemented or Reserved

Figure 27-8. FECCR Index Register (FECCRIX)

ECCRIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 27-14. FECCRIX Field Descriptions

Field	Description
2-0 ECCRIX[2:0]	**ECC Error Register Index**— The ECCRIX bits are used to select which word of the FECCR register array is being read. See Section 27.3.2.13, “Flash ECC Error Results Register (FECCR),” for more details.

27.3.2.5 Flash Configuration Register (FCNFG)

The FCNFG register enables the Flash command complete interrupt and forces ECC faults on Flash array read access from the CPU or XGATE.

Offset Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 27-9. Flash Configuration Register (FCNFG)

CCIE, IGNSF, FDFD, and FSFD bits are readable and writable while remaining bits read 0 and are not writable.

Table 27-15. FCNFG Field Descriptions

Field	Description
7 CCIE	**Command Complete Interrupt Enable** — The CCIE bit controls interrupt generation when a Flash command has completed. 0 Command complete interrupt disabled 1 An interrupt will be requested whenever the CCIF flag in the FSTAT register is set (see Section 27.3.2.7)
4 IGNSF	**Ignore Single Bit Fault** — The IGNSF controls single bit fault reporting in the FERSTAT register (see Section 27.3.2.8). 0 All single bit faults detected during array reads are reported 1 Single bit faults detected during array reads are not reported and the single bit fault interrupt will not be generated

27.4.2.11 Unsecure Flash Command

The Unsecure Flash command will erase the entire P-Flash and D-Flash memory space and, if the erase is successful, will release security.

Table 27-53. Unsecure Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0B	Not required

Upon clearing CCIF to launch the Unsecure Flash command, the Memory Controller will erase the entire P-Flash and D-Flash memory space and verify that it is erased. If the Memory Controller verifies that the entire Flash memory space was properly erased, security will be released. If the erase verify is not successful, the Unsecure Flash operation sets MGSTAT1 and terminates without changing the security state. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag is set after the Unsecure Flash operation has completed.

Table 27-54. Unsecure Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if a Load Data Field command sequence is currently active
		Set if command not available in current mode (see Table 27-30)
	FPVIOL	Set if any area of the P-Flash memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation
FERSTAT	EPVIOLIF	Set if any area of the buffer RAM EEE partition is protected

27.4.2.12 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command will only execute if it is enabled by the KEYEN bits in the FSEC register (see Table 27-11). The Verify Backdoor Access Key command releases security if user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see Table 27-3). The Verify Backdoor Access Key command must not be executed from the Flash block containing the backdoor comparison key to avoid code runaway.

Table 27-55. Verify Backdoor Access Key Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0C	Not required
001	Key 0	
010	Key 1	
011	Key 2	
100	Key 3	

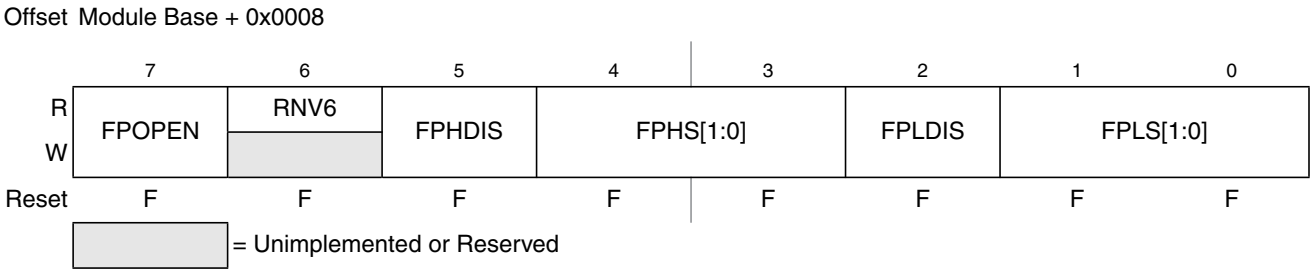


Figure 28-13. Flash Protection Register (FPROT)

The (unreserved) bits of the FPROT register are writable with the restriction that the size of the protected region can only be increased (see Section 28.3.2.9.1, "P-Flash Protection Restrictions," and Table 28-23).

During the reset sequence, the FPROT register is loaded with the contents of the P-Flash protection byte in the Flash configuration field at global address 0x7F_FF0C located in P-Flash memory (see Table 28-3) as indicated by reset condition 'F' in Figure 28-13. To change the P-Flash protection that will be loaded during the reset sequence, the upper sector of the P-Flash memory must be unprotected, then the P-Flash protection byte must be reprogrammed. If a double bit fault is detected while reading the P-Flash phrase containing the P-Flash protection byte during the reset sequence, the FPOPEN bit will be cleared and remaining bits in the FPROT register will be set to leave the P-Flash memory fully protected.

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

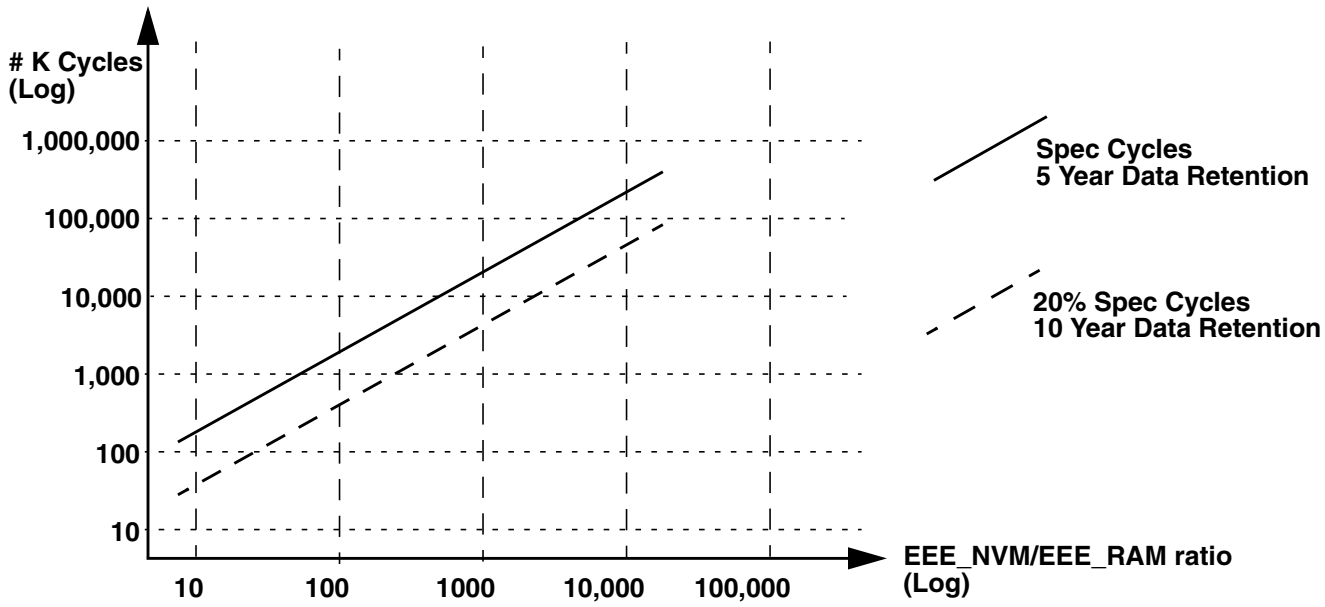
Table 28-19. FPROT Field Descriptions

Field	Description
7 FPOPEN	**Flash Protection Operation Enable** — The FPOPEN bit determines the protection function for program or erase operations as shown in Table 28-20 for the P-Flash block. 0 When FPOPEN is clear, the FPHDIS and FPLDIS bits define unprotected address ranges as specified by the corresponding FPHS and FPLS bits 1 When FPOPEN is set, the FPHDIS and FPLDIS bits enable protection for the address range specified by the corresponding FPHS and FPLS bits
6 RNV[6]	**Reserved Nonvolatile Bit** — The RNV bit should remain in the erased state for future enhancements.
5 FPHDIS	**Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory ending with global address 0x7F_FFFF. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
4–3 FPHS[1:0]	**Flash Protection Higher Address Size** — The FPHS bits determine the size of the protected/unprotected area in P-Flash memory as shown inTable 28-21. The FPHS bits can only be written to while the FPHDIS bit is set.
2 FPLDIS	**Flash Protection Lower Address Range Disable** — The FPLDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory beginning with global address 0x7F_8000. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
1–0 FPLS[1:0]	**Flash Protection Lower Address Size** — The FPLS bits determine the size of the protected/unprotected area in P-Flash memory as shown in Table 28-22. The FPLS bits can only be written to while the FPLDIS bit is set.

5. This represents the number of writes of updated data words to the EEE_RAM partition. Typical endurance performance for the Emulated EEPROM array is based on typical endurance performance and the EEE algorithm implemented on this product family. Spec. table quotes typical endurance evaluated at 25°C for this product family.
6. This is equivalent to using a single byte or aligned word in the EEE_RAM with 32K D-Flash allocated for EEEPROM

The number of program/erase cycles for the EEPROM/D-Flash depends upon the partitioning of D-Flash used for EEPROM Emulation. Defining RAM size allocated for EEE as EEE-RAM and D-Flash partition allocated to EEE as EEE_NVM, the minimum number of program/erase cycles is specified depending upon the ratio of EEE_NVM/EEE_RAM. The minimum ratio EEE_NVM/EEE_RAM =8.

Figure A-2. Program/Erase Dependency on D-Flash Partitioning



0x03D0–0x03FF Timer Module (TIM) Map (Sheet 2 of 2)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x03E4	TC2H	R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0x03E5	TC2L	R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x03E6	TC3H	R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0x03E7	TC3L	R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x03E8	TC4H	R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0x03E9	TC4L	R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x03EA	TC5H	R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0x03EB	TC5L	R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x03EC	TC6H	R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0x03ED	TC6L	R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x03EE	TC7H	R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0x03EF	TC7L	R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x03F0	PACTL	R W	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI
0x03F1	PAFLG	R W	0	0	0	0	0	0	PAOVF	PAIF
0x03F2	PACNTH	R W	PACNT15	PACNT14	PACNT13	PACNT12	PACNT11	PACNT10	PACNT9	PACNT8
0x03F3	PACNTL	R W	PACNT7	PACNT6	PACNT5	PACNT4	PACNT3	PACNT2	PACNT1	PACNT0
0x03F4– 0x03FB	Reserved	R W	0	0	0	0	0	0	0	0
0x03FC	OCPD	R W	OCPD7	OCPD6	OCPD5	OCPD4	OCPD3	OCPD2	OCPD1	OCPD0
0x03FD	Reserved	R W								
0x03FE	PTPSR	R W	PTPSR7	PTPSR6	PTPSR5	PTPSR4	PTPSR3	PTPSR2	PTPSR1	PTPSR0
0x03FF	Reserved	R W								