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	Not For New Designs
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	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	6K 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/mc9s12xb128maa

4.3.2.6 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. If external trigger is enabled (ETRIGE = 1) an initial write to ATDCTL5 is required to allow starting of a conversion sequence which will then occur on each trigger event. Start of conversion means the beginning of the sampling phase.

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

Figure 4-8. ATD Control Register 5 (ATDCTL5)

Read: Anytime

Write: Anytime

Table 4-14. ATDCTL5 Field Descriptions

Field	Description
7 DJM	**Result Register Data Justification** — This bit controls justification of conversion data in the result registers. See Section 4.3.2.16, “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 <st-bold>4.3.2.16 ATD Conversion Result Registers (ATDDRx) for details. 0 Unsigned data representation in the result registers. 1 Signed data representation in the result registers. Table 4-15 summarizes the result data formats available and how they are set up using the control bits. Table 4-16 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. If external trigger is enabled (ETRIGE=1) setting this bit has no effect, that means each trigger event starts a single conversion sequence. 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 CD/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 or wrapping around to AN0 (channel 0. 0 Sample only one channel 1 Sample across several channels

6.8.1.8 Dyadic Addressing (DYA)

In this mode the result of an operation between two registers is stored in one of the registers used as operands.

RD = RD ∗ RS is the general register to register format, with register RD being the first operand and RS the second. RD and RS can be any of the 8 general purpose registers R0 … R7. If R0 is used as the destination register, only the condition code flags are updated. This addressing mode is used only for shift operations with a variable shift value

Examples:

```
LSL     R4,R5    ; R4 = R4 << R5
LSR     R4,R5    ; R4 = R4 >> R5
```

6.8.1.9 Triadic Addressing (TRI)

In this mode the result of an operation between two or three registers is stored into a third one.
RD = RS1 ∗ RS2 is the general format used in the order RD, RS1, RS1. RD, RS1, RS2 can be any of the 8 general purpose registers R0 … R7. If R0 is used as the destination register RD, only the condition code flags are updated. This addressing mode is used for all arithmetic and logical operations.

Examples:

```
ADC     R5,R6,R7          ; R5 = R6 + R7 + Carry
SUB     R5,R6,R7          ; R5 = R6 - R7
```

6.8.1.10 Relative Addressing 9-Bit Wide (REL9)

A 9-bit signed word address offset is included in the instruction word. This addressing mode is used for conditional branch instructions.

Examples:

```
BCC     REL9              ; PC = PC + 2 + (REL9 << 1)
BEQ     REL9              ; PC = PC + 2 + (REL9 << 1)
```

6.8.1.11 Relative Addressing 10-Bit Wide (REL10)

An 11-bit signed word address offset is included in the instruction word. This addressing mode is used for the unconditional branch instruction.

Examples:

```
BRA     REL10             ; PC = PC + 2 + (REL10 << 1)
```

6.8.1.12 Index Register plus Immediate Offset (IDO5)

(RS, #offset5) provides an unsigned offset from the base register.

Examples:

```
LDB     R4,(R1,#offset)   ; loads a byte from R1+offset into R4
STW     R4,(R1,#offset)   ; stores R4 as a word to R1+offset
```

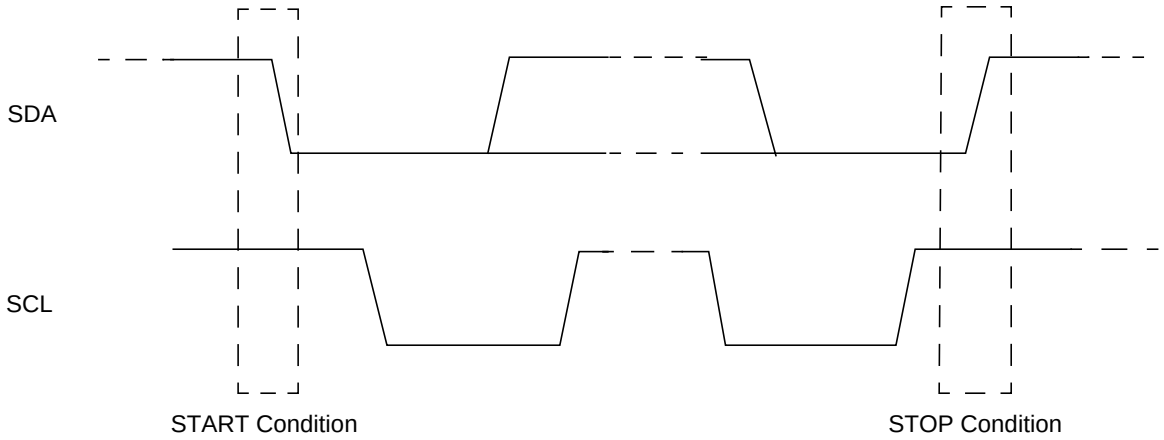


Figure 9-10. Start and Stop Conditions

9.4.1.2 Slave Address Transmission

The first byte of data transfer immediately after the START signal is the slave address transmitted by the master. This is a seven-bit calling address followed by a R/W bit. The R/W bit tells the slave the desired direction of data transfer.

1 = Read transfer, the slave transmits data to the master.
0 = Write transfer, the master transmits data to the slave.

Only the slave with a calling address that matches the one transmitted by the master will respond by sending back an acknowledge bit. This is done by pulling the SDA low at the 9th clock (see Figure 9-9).

No two slaves in the system may have the same address. If the IIC bus is master, it must not transmit an address that is equal to its own slave address. The IIC bus cannot be master and slave at the same time.However, if arbitration is lost during an address cycle the IIC bus will revert to slave mode and operate correctly even if it is being addressed by another master.

9.4.1.3 Data Transfer

As soon as successful slave addressing is achieved, the data transfer can proceed byte-by-byte in a direction specified by the R/W bit sent by the calling master

All transfers that come after an address cycle are referred to as data transfers, even if they carry sub-address information for the slave device.

Each data byte is 8 bits long. Data may be changed only while SCL is low and must be held stable while SCL is high as shown in Figure 9-9. There is one clock pulse on SCL for each data bit, the MSB being transferred first. Each data byte has to be followed by an acknowledge bit, which is signalled from the receiving device by pulling the SDA low at the ninth clock. So one complete data byte transfer needs nine clock pulses.

If the slave receiver does not acknowledge the master, the SDA line must be left high by the slave. The master can then generate a stop signal to abort the data transfer or a start signal (repeated start) to commence a new calling.

Table 10-9. CANRFLG Register Field Descriptions (continued)

Field	Description
1 OVRIF	**Overrun Interrupt Flag** — This flag is set when a data overrun condition occurs. If not masked, an error interrupt is pending while this flag is set. 0 No data overrun condition 1 A data overrun detected
0 RXF[2]	**Receive Buffer Full Flag** — RXF is set by the MSCAN when a new message is shifted in the receiver FIFO. This flag indicates whether the shifted buffer is loaded with a correctly received message (matching identifier, matching cyclic redundancy code (CRC) and no other errors detected). After the CPU has read that message from the RxFG buffer in the receiver FIFO, the RXF flag must be cleared to release the buffer. A set RXF flag prohibits the shifting of the next FIFO entry into the foreground buffer (RxFG). If not masked, a receive interrupt is pending while this flag is set. 0 No new message available within the RxFG 1 The receiver FIFO is not empty. A new message is available in the RxFG

1 Redundant Information for the most critical CAN bus status which is "bus-off". This only occurs if the Tx error counter exceeds a number of 255 errors. Bus-off affects the receiver state. As soon as the transmitter leaves its bus-off state the receiver state skips to RxOK too. Refer also to TSTAT[1:0] coding in this register.

2 To ensure data integrity, do not read the receive buffer registers while the RXF flag is cleared. For MCUs with dual CPUs, reading the receive buffer registers while the RXF flag is cleared may result in a CPU fault condition.

10.3.2.6 MSCAN Receiver Interrupt Enable Register (CANRIER)

This register contains the interrupt enable bits for the interrupt flags described in the CANRFLG register.

	7	6	5	4	3	2	1	0
R W	WUPIE	CSCIE	RSTATE1	RSTATE0	TSTATE1	TSTATE0	OVRIE	RXFIE
Reset:	0	0	0	0	0	0	0	0

Figure 10-9. MSCAN Receiver Interrupt Enable Register (CANRIER)

NOTE

The CANRIER 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).

The RSTATE[1:0], TSTATE[1:0] bits are not affected by initialization mode.

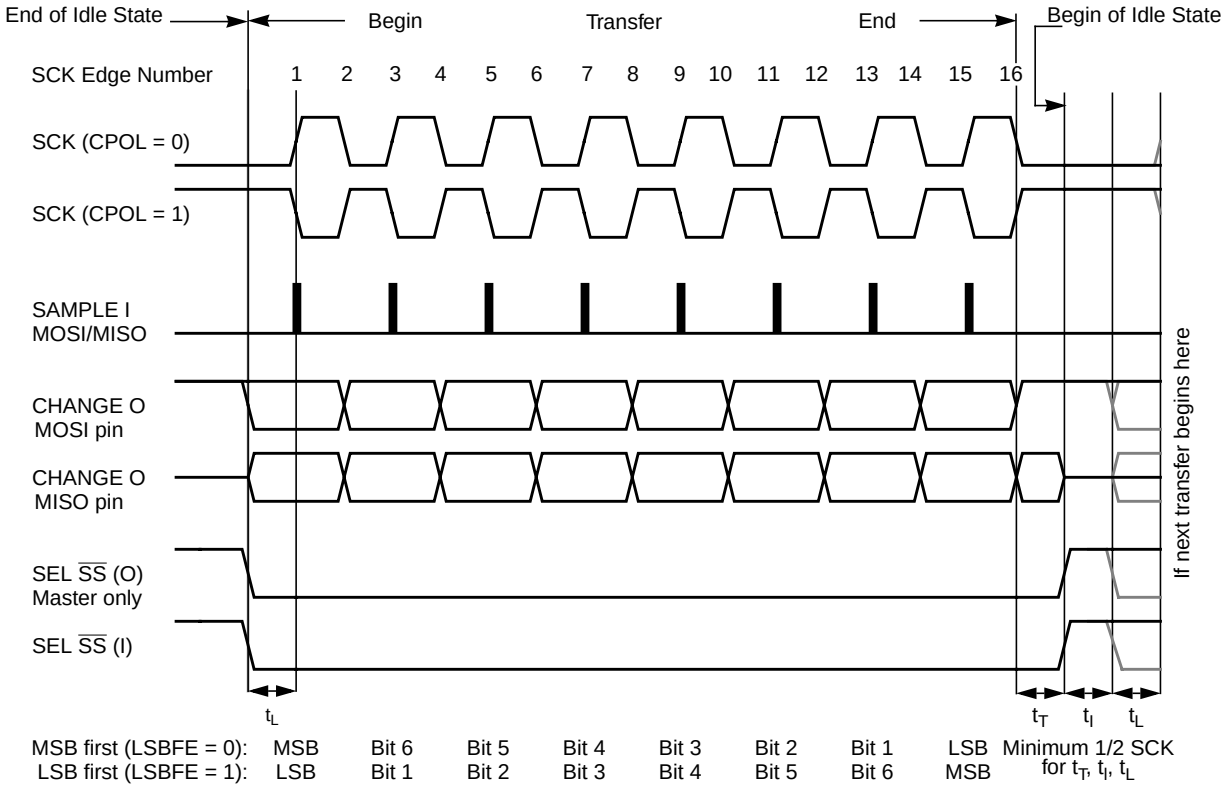
Read: Anytime
Write: Anytime when not in initialization mode

Data reception is double buffered. Data is shifted serially into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After the 16th (last) SCK edge:

- Data that was previously in the master SPI data register should now be in the slave data register and the data that was in the slave data register should be in the master.
- The SPIF flag in the SPI status register is set, indicating that the transfer is complete.

Figure 12-11 is a timing diagram of an SPI transfer where CPHA = 0. SCK waveforms are shown for CPOL = 0 and CPOL = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave and the MOSI signal is the output from the master. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.



t_L = Minimum leading time before the first SCK edge
t_T = Minimum trailing time after the last SCK edge
t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time)
t_L, t_T, and t_I are guaranteed for the master mode and required for the slave mode.

Figure 12-11. SPI Clock Format 0 (CPHA = 0)

15.1.2.3 Low-Power Modes

The BDM can be used until all bus masters (e.g., CPU or XGATE) are in stop mode. When CPU is in a low power mode (wait or stop mode) all BDM firmware commands as well as the hardware BACKGROUND command can not be used respectively are ignored. In this case the CPU can not enter BDM active mode, and only hardware read and write commands are available. Also the CPU can not enter a low power mode during BDM active mode.

If all bus masters are in stop mode, the BDM clocks are stopped as well. When BDM clocks are disabled and one of the bus masters exits from stop mode the BDM clocks will restart and BDM will have a soft reset (clearing the instruction register, any command in progress and disable the ACK function). The BDM is now ready to receive a new command.

15.1.3 Block Diagram

A block diagram of the BDM is shown in Figure 15-1.

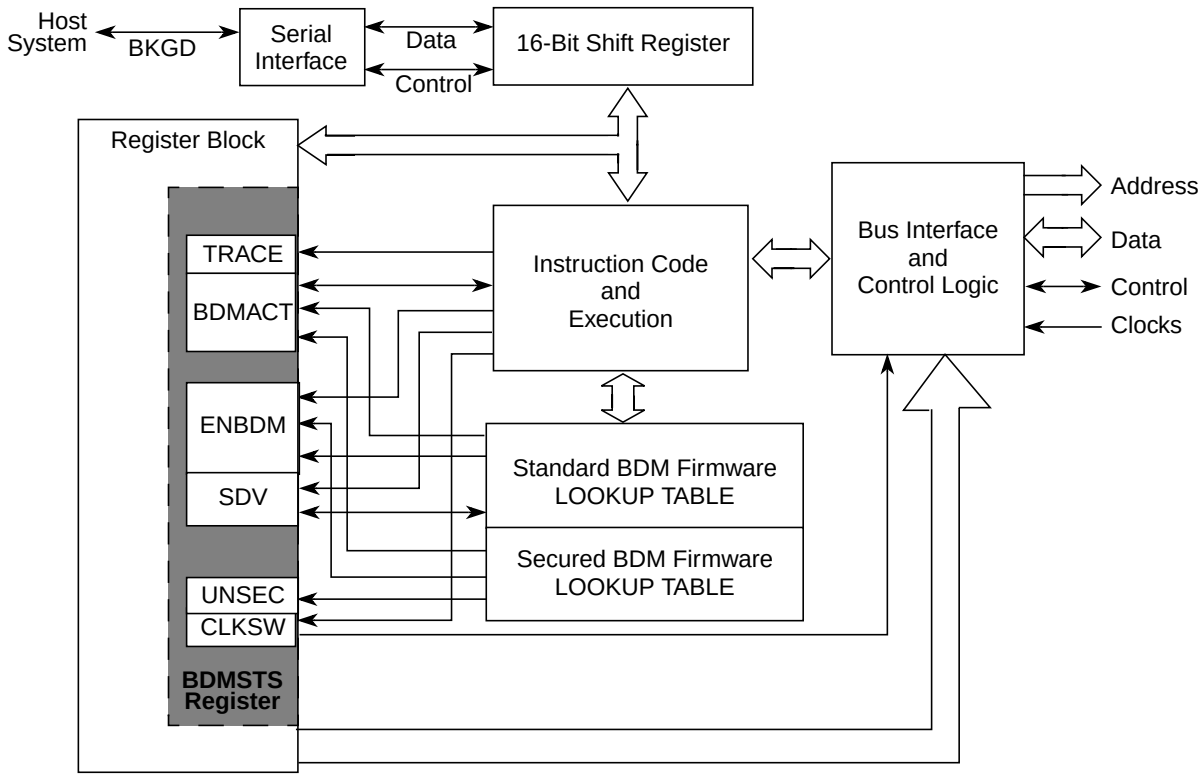


Figure 15-1. BDM Block Diagram

17.4.4 Chip Bus Control

The MMC controls the address buses and the data buses that interface the S12X masters (CPU, BDM and XGATE) with the rest of the system (master buses). In addition the MMC handles all CPU read data bus swapping operations. All internal and external resources are connected to specific target buses (see Figure 1-26).

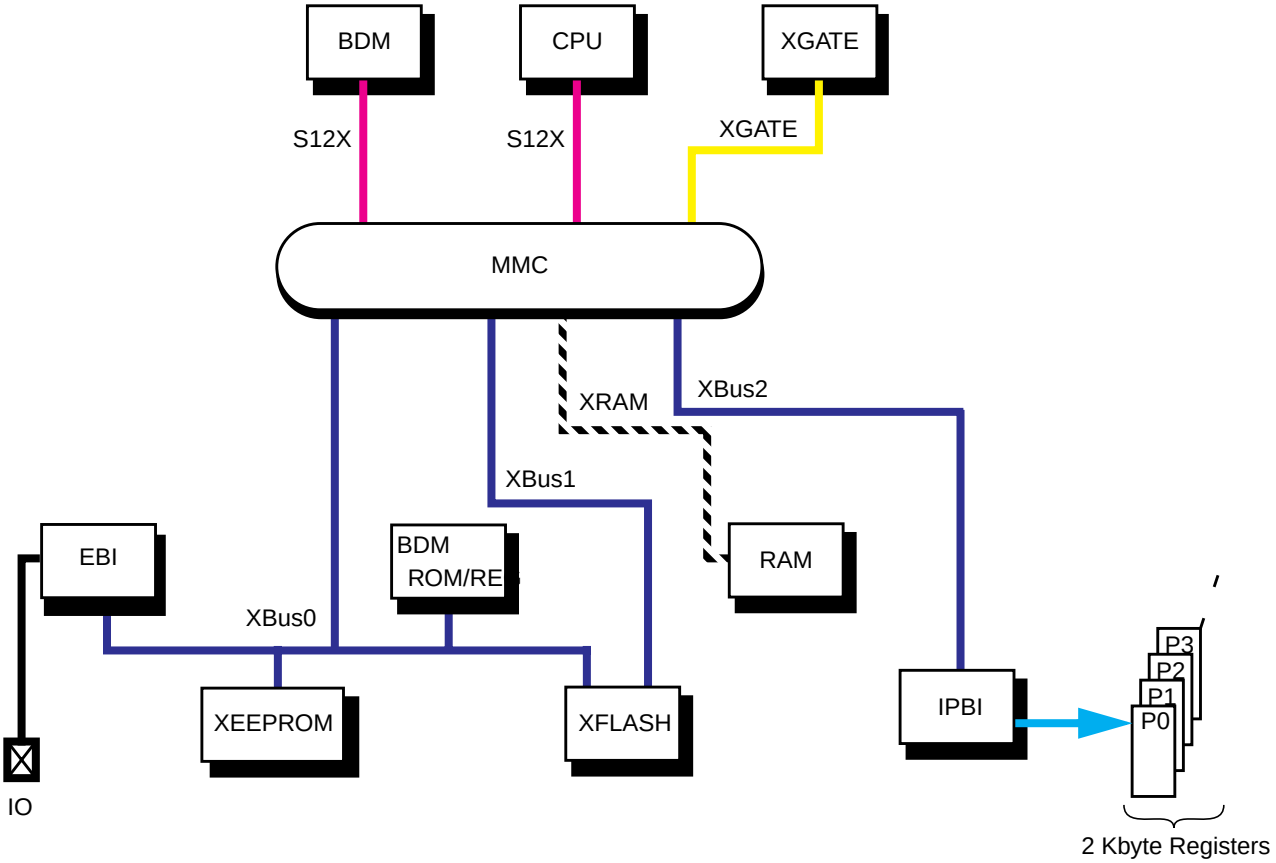


Figure 17-26. S12X Architecture

17.4.4.1 Master Bus Prioritization

The following rules apply when prioritizing accesses over master buses:

- The CPU has priority over the BDM, unless the BDM access is stalled for more than 128 cycles. In the later case the CPU will be stalled after finishing the current operation and the BDM will gain access to the bus.
- XGATE access to PRU registers constitutes a special case. It is always granted and stalls the CPU and BDM for its duration.

Due to internal visibility of CPU accesses the CPU will be halted during XGATE or BDM access to any PRR. This rule applies also in normal modes to ensure that operation of the device is the same as in emulation modes.

A summary of PRR accesses:

- An aligned word access to a PRR will take 2 bus cycles.
- A misaligned word access to a PRRs will take 4 cycles. If one of the two bytes accessed by the misaligned word access is not a PRR, the access will take only 3 cycles.
- A byte access to a PRR will take 2 cycles.

Table 18-23. PRR Listing

PRR Name	PRR Local Address	PRR Location
PORTA	0x0000	PIM
PORTB	0x0001	PIM
DDRA	0x0002	PIM
DDRB	0x0003	PIM
PORTC	0x0004	PIM
PORTD	0x0005	PIM
DDRC	0x0006	PIM
DDRD	0x0007	PIM
PORTE	0x0008	PIM
DDRE	0x0009	PIM
MMCCTL0	0x000A	MMC
MODE	0x000B	MMC
PUCR	0x000C	PIM
RDRIV	0x000D	PIM
EBICTL0	0x000E	EBI
EBICTL1	0x000F	EBI
Reserved	0x0012	MMC
MMCCTL1	0x0013	MMC
ECLKCTL	0x001C	PIM
Reserved	0x001D	PIM
PORTK	0x0032	PIM
DDRK	0x0033	PIM

22.3.2.24 Port S Input Register (PTIS)

	7	6	5	4	3	2	1	0
R	PTIS7	PTIS6	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0
W								
Reset[1]	—	—	—	—	—	—	—	—

= Unimplemented or Reserved

Figure 22-26. Port S Input Register (PTIS)

1 These registers are reset to zero. Two bus clock cycles after reset release the register values are updated with the associated pin values.

Read: Anytime.

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

This register always reads back the buffered state of the associated pins. This also can be used to detect overload or short circuit conditions on output pins.

23.0.5.11 S12X_EBI Ports, BKGD, VREGEN Pin Pull-up Control Register (PUCR)

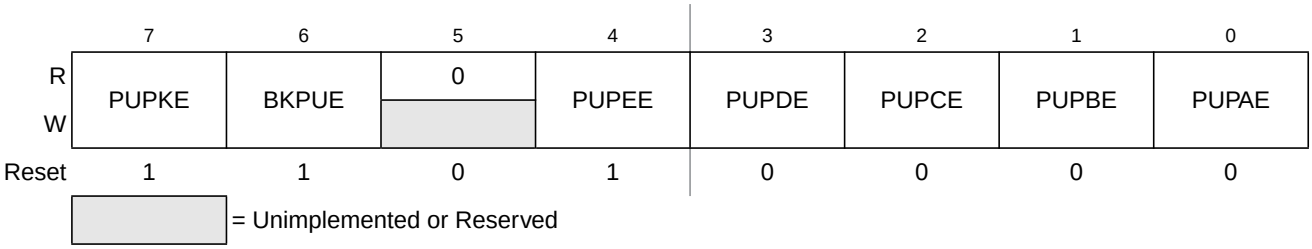


Figure 23-13. S12X_EBI Ports, BKGD, VREGEN Pin Pull-up Control Register (PUCR)

Read: Anytime in single-chip modes.

Write: Anytime, except BKPUE which is writable in special test mode only.

This register is used to enable pull-up devices for the associated ports A, B, C, D, E, and K. Pull-up devices are assigned on a per-port basis and apply to any pin in the corresponding port currently configured as an input.

Table 23-14. PUCR Field Descriptions

Field	Description
7 PUPKE	**Pull-up Port K Enable** 0 Port K pull-up devices are disabled. 1 Enable pull-up devices for Port K input pins.
6 BKPUE	**BKGD and VREGEN Pin Pull-up Enable** 0 BKGD and V_{REGEN} pull-up devices are disabled. 1 Enable pull-up devices on BKGD and V_{REGEN} pins.
4 PUPEE	**Pull-up Port E Enable** 0 Port E pull-up devices on bit 7, 4–0 are disabled. 1 Enable pull-up devices for Port E input pins bits 7, 4–0. **Note:** Bits 5 and 6 of Port E have pull-down devices which are only enabled during reset. This bit has no effect on these pins.
3 PUPDE	**Pull-up Port D Enable** 0 Port D pull-up devices are disabled. 1 Enable pull-up devices for all Port D input pins.
2 PUPCE	**Pull-up Port C Enable** 0 Port C pull-up devices are disabled. 1 Enable pull-up devices for all Port C input pins.
1 PUPBE	**Pull-up Port B Enable** 0 Port B pull-up devices are disabled. 1 Enable pull-up devices for all Port B input pins.
0 PUPAE	**Pull-up Port A Enable** 0 Port A pull-up devices are disabled. 1 Enable pull-up devices for all Port A input pins.

Table 24-2. PIM Memory Map (Sheet 3 of 3)

Address	Use	Access
0x0267	Port H Interrupt Flag Register (PIFH)	Read / Write
0x0268	Port J Data Register (PTJ)	Read / Write[1]
0x0269	Port J Input Register (PTIJ)	Read
0x026A	Port J Data Direction Register (DDRJ)	Read / Write[1]
0x026B	Port J Reduced Drive Register (RDRJ)	Read / Write[1]
0x026C	Port J Pull Device Enable Register (PERJ)	Read / Write[1]
0x026D	Port J Polarity Select Register (PPSJ)	Read / Write[1]
0x026E	Port J Interrupt Enable Register (PIEJ)	Read / Write[1]
0x026F	Port J Interrupt Flag Register (PIFJ)	Read / Write[1]
0x0270 : 0x0277	PIM Reserved	—
0x0278	Port AD1 Data Register 0 (PT0AD1)	Read / Write
0x0279	Port AD1 Data Register 1 (PT1AD1)	Read / Write
0x027A	Port AD1 Data Direction Register 0 (DDR0AD1)	Read / Write
0x027B	Port AD1 Data Direction Register 1 (DDR1AD1)	Read / Write
0x027C	Port AD1 Reduced Drive Register 0 (RDR0AD1)	Read / Write
0x027D	Port AD1 Reduced Drive Register 1 (RDR1AD1)	Read / Write
0x027E	Port AD1 Pull Up Enable Register 0 (PER0AD1)	Read / Write
0x027F	Port AD1 Pull Up Enable Register 1 (PER1AD1)	Read / Write

1. Write access not applicable for one or more register bits. Refer to Section 24.0.5, “Register Descriptions”.

24.0.5 Register Descriptions

Table 24-3 summarizes the effect on the various configuration bits, data direction (DDR), output level (IO), reduced drive (RDR), pull enable (PE), pull select (PS), and interrupt enable (IE) for the ports.

The configuration bit PS is used for two purposes:

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

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
PTJ	R	PTJ7	PTJ6	0	0	0	0	PTJ1	PTJ0
	W								
PTIJ	R	PTIJ7	PTIJ6	0	0	0	0	PTIJ1	PTIJ0
	W								
DDRJ	R	DDRJ7	DDRJ6	0	0	0	0	DDRJ1	DDRJ0
	W								
RDRJ	R	RDRJ7	RDRJ6	0	0	0	0	RDRJ1	RDRJ0
	W								
PERJ	R	PERJ7	PERJ6	0	0	0	0	PERJ1	PERJ0
	W								
PPSJ	R	PPSJ7	PPSJ6	0	0	0	0	PPSJ1	PPSJ0
	W								
PIEJ	R	PIEJ7	PIEJ6	0	0	0	0	PIEJ1	PIEJ0
	W								
PIFJ	R	PPSJ7	PPSJ6	0	0	0	0	PPSJ1	PPSJ0
	W								
Reserved	R	0	0	0	0	0	0	0	0
	W								
Reserved	R	0	0	0	0	0	0	0	0
	W								
Reserved	R	0	0	0	0	0	0	0	0
	W								
Reserved	R	0	0	0	0	0	0	0	0
	W								
Reserved	R	0	0	0	0	0	0	0	0
	W								
Reserved	R	0	0	0	0	0	0	0	0
	W								
Reserved	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented or Reserved

Figure 24-2. PIM Register Summary (Sheet 6 of 7)

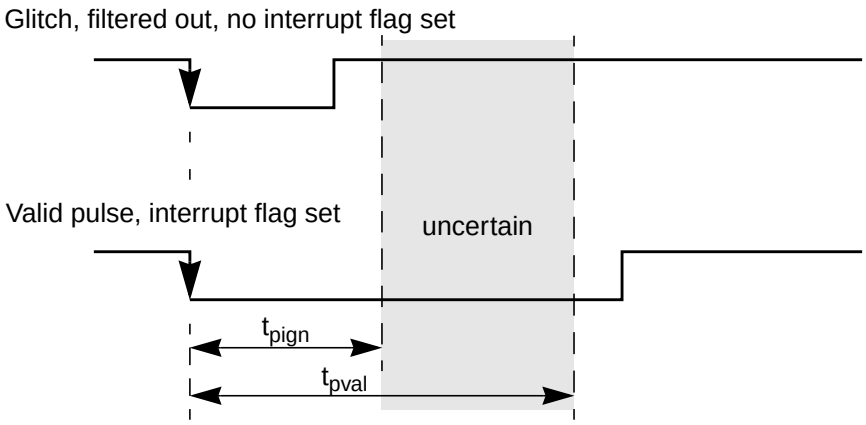


Figure 24-69. Interrupt Glitch Filter on Port P, H, and J (PPS = 0)

Table 24-62. Pulse Detection Criteria

Pulse	Mode		
	$\overline{STOP}$	Unit	STOP[1]
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.

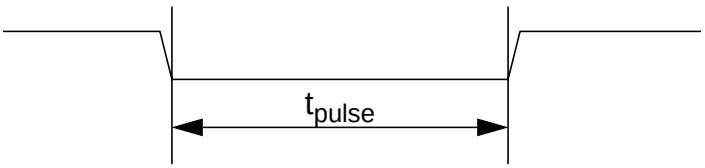


Figure 24-70. Pulse Illustration

A valid edge on an input is detected if 4 consecutive samples of a passive level are followed by 4 consecutive samples of an active level directly or indirectly.

The filters are continuously clocked by the bus clock in run and wait mode. In stop mode, the clock is generated by an RC-oscillator in the port integration module. To maximize current saving the RC oscillator runs only if the following condition is true on any pin individually:

Sample count <= 4 and interrupt enabled (PIE = 1) and interrupt flag not set (PIF = 0).

24.0.9 Low-Power Options

24.0.9.1 Run Mode

No low-power options exist for this module in run mode.

Table 27-10. Flash Register Bank Selects

BKSEL[1:0]	Selected Block
10	Flash Block 2
11	Flash Block 3

27.3.2.5 Flash Protection Register (FPROT)

The FPROT register defines which Flash sectors are protected against program or erase operations.

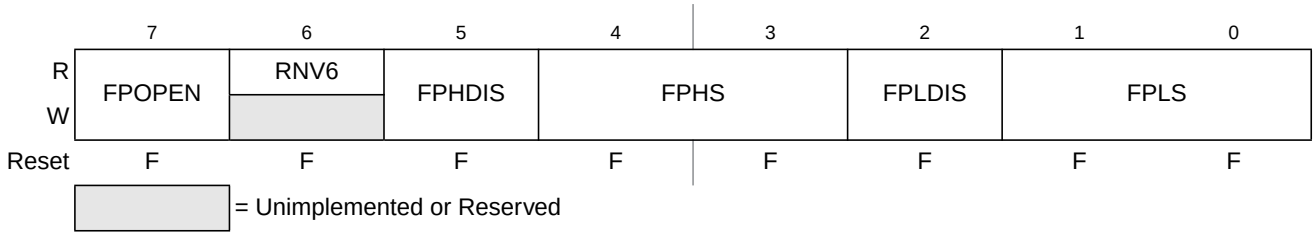


Figure 27-10. Flash Protection Register (FPROT)

All bits in the FPROT register are readable and writable with restrictions (see Section 27.3.2.5.1, “Flash Protection Restrictions”) except for RNV[6] which is only readable.

During the reset sequence, the FPROT register is loaded from the Flash Configuration Field at global address 0x7F_FF0D. To change the Flash protection that will be loaded during the reset sequence, the upper sector of the Flash memory must be unprotected, then the Flash Protect/Security byte located as described in Table 27-1 must be reprogrammed.

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

Table 27-11. FPROT Field Descriptions

Field	Description
7 FPOPEN	**Flash Protection Open** — The FPOPEN bit determines the protection function for program or erase as shown in Table 27-12. 0 The FPHDIS and FPLDIS bits define unprotected address ranges as specified by the corresponding FPHS[1:0] and FPLS[1:0] bits. For an MCU without an EEPROM module, the FPOPEN clear state allows the main part of the Flash block to be protected while a small address range can remain unprotected for EEPROM emulation. 1 The FPHDIS and FPLDIS bits enable protection for the address range specified by the corresponding FPHS[1:0] and FPLS[1:0] bits.
6 RNV6	**Reserved Nonvolatile Bit** — The RNV[6] 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 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[1:0] bits determine the size of the protected/unprotected area as shown inTable 27-13. The FPHS[1:0] bits can only be written to while the FPHDIS bit is set.

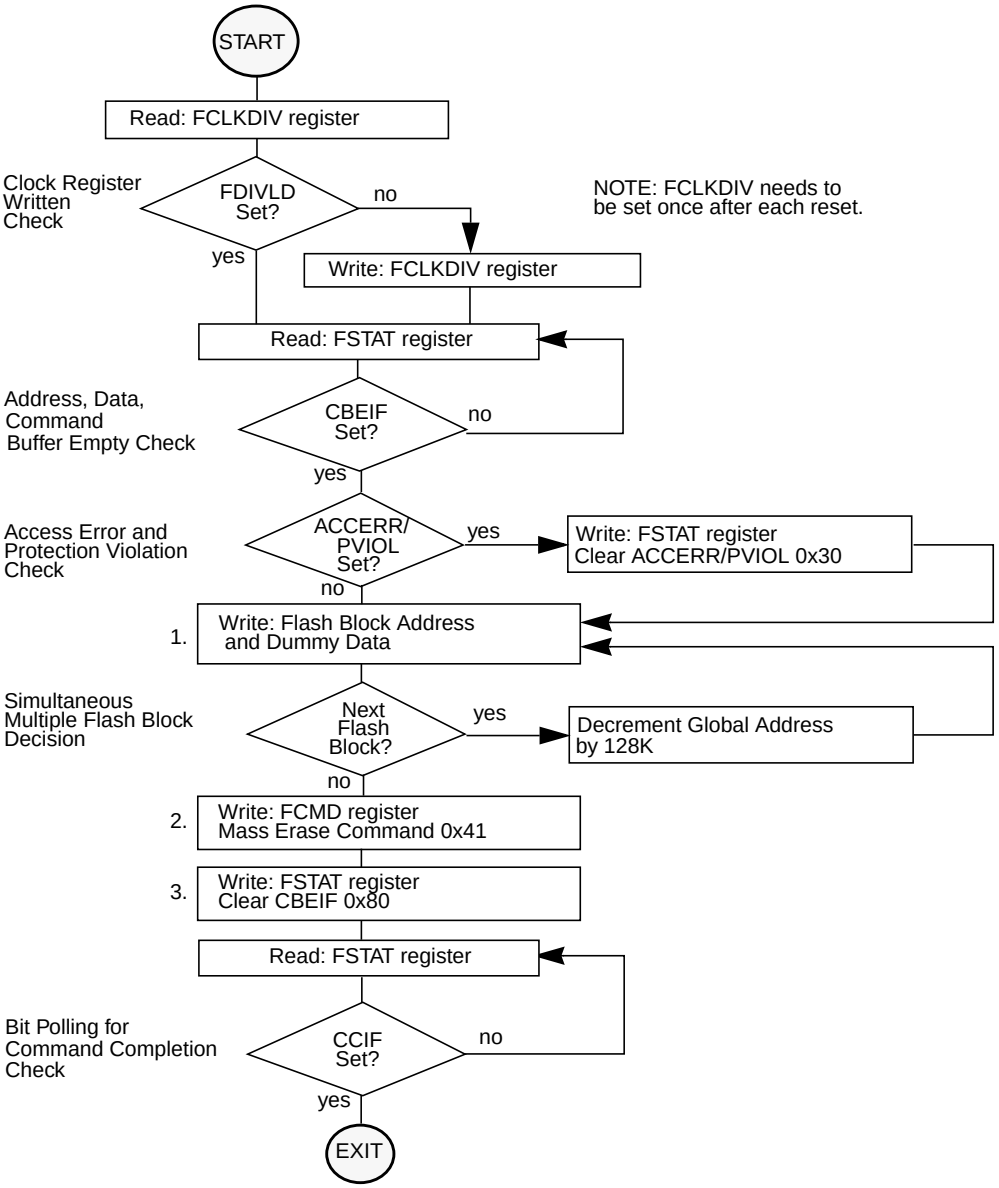


Figure 27-30. Example Mass Erase Command Flow

27.4.3 Illegal Flash Operations

The ACCERR flag will be set during the command write sequence if any of the following illegal steps are performed, causing the command write sequence to immediately abort:

1. Writing to a Flash address before initializing the FCLKDIV register.
2. Writing a byte or misaligned word to a valid Flash address.
3. Starting a command write sequence while a data compress operation is active.
4. Starting a command write sequence while a sector erase abort operation is active.
5. Writing a Flash address in step 1 of a command write sequence that is not the same relative address as the first one written in the same command write sequence.
6. Writing to any Flash register other than FCMD after writing to a Flash address.
7. Writing a second command to the FCMD register in the same command write sequence.
8. Writing an invalid command to the FCMD register.
9. When security is enabled, writing a command other than mass erase to the FCMD register when the write originates from a non-secure memory location or from the Background Debug Mode.
10. Writing to a Flash address after writing to the FCMD register.
11. Writing to any Flash register other than FSTAT (to clear CBEIF) after writing to the FCMD register.
12. Writing a 0 to the CBEIF flag in the FSTAT register to abort a command write sequence.

The ACCERR flag will not be set if any Flash register is read during a valid command write sequence.

The ACCERR flag will also be set if any of the following events occur:

1. Launching the sector erase abort command while a sector erase operation is active which results in the early termination of the sector erase operation (see Section 27.4.2.6, “Sector Erase Abort Command”).
2. The MCU enters stop mode and a program or erase operation is in progress. The operation is aborted immediately and any pending command is purged (see Section 27.5.2, “Stop Mode”).

If the Flash memory is read during execution of an algorithm (CCIF = 0), the read operation will return invalid data and the ACCERR flag will not be set.

If the ACCERR flag is set in the FSTAT register, the user must clear the ACCERR flag before starting another command write sequence (see Section 27.3.2.6, “Flash Status Register (FSTAT)”).

The PVIOL flag will be set after the command is written to the FCMD register during a command write sequence if any of the following illegal operations are attempted, causing the command write sequence to immediately abort:

1. Writing the program command if an address written in the command write sequence was in a protected area of the Flash memory
2. Writing the sector erase command if an address written in the command write sequence was in a protected area of the Flash memory
3. Writing the mass erase command to a Flash block while any Flash protection is enabled in the block

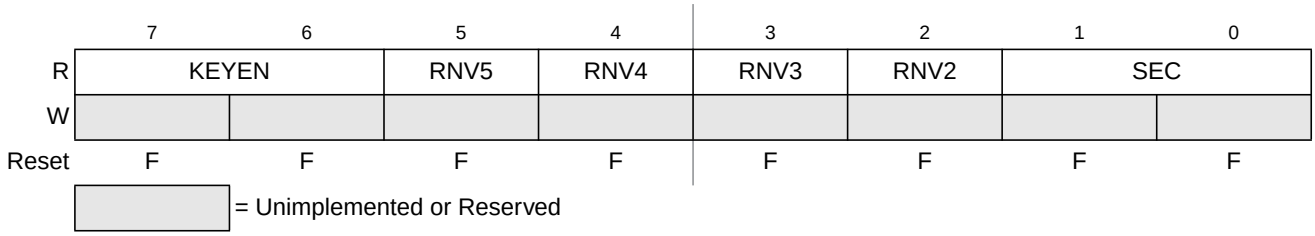


Figure 28-5. Flash Security Register (FSEC)

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 28-5.

Table 28-3. 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 28-4.
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 28-5. If the Flash module is unsecured using backdoor key access, the SEC[1:0] bits are forced to 1:0.

Table 28-4. 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 28-5. 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 28.6, “Flash Module Security”.

28.3.2.3 Flash Test Mode Register (FTSTMOD)

The FTSTMOD register is used to control Flash test features.

28.4.2.1 Erase Verify Command

The erase verify operation will verify that a Flash block is erased.

An example flow to execute the erase verify operation is shown in Figure 28-25. The erase verify command write sequence is as follows:

1. Write to a Flash block address to start the command write sequence for the erase verify command. The address and data written will be ignored. Multiple Flash blocks can be simultaneously erase verified by writing to the same relative address in each Flash block.
2. Write the erase verify command, 0x05, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the erase verify command.

After launching the erase verify command, the CCIF flag in the FSTAT register will set after the operation has completed unless a new command write sequence has been buffered. The number of bus cycles required to execute the erase verify operation is equal to the number of addresses in a Flash block plus 14 bus cycles as measured from the time the CBEIF flag is cleared until the CCIF flag is set. Upon completion of the erase verify operation, the BLANK flag in the FSTAT register will be set if all addresses in the selected Flash blocks are verified to be erased. If any address in a selected Flash block is not erased, the erase verify operation will terminate and the BLANK flag in the FSTAT register will remain clear. The MRDS bits in the FTSTMOD register will determine the sense-amp margin setting during the erase verify operation.

A.1.7 Operating Conditions

This section describes the operating conditions of the device. Unless otherwise noted those conditions apply to all the following data.

NOTE

Please refer to the temperature rating of the device (C, V, M) with regards to the ambient temperature T_A and the junction temperature T_J. For power dissipation calculations refer to Section A.1.8, "Power Dissipation and Thermal Characteristics".

Table A-4. Operating Conditions

Rating	Symbol	Min	Typ	Max	Unit
I/O, regulator and analog supply voltage	V_{DD35}	3.15	5	5.5	V
Digital logic supply voltage[1]	V_{DD}	2.35	2.5	2.75	V
PLL supply voltage[2]	V_{DDPLL}	2.35	2.5	2.75	V
Voltage difference V_{DDX} to V_{DDR} and V_{DDA}	Δ_{VDDX}	–0.1	0	0.1	V
Voltage difference V_{SSX} to V_{SSR} and V_{SSA}	Δ_{VSSX}	–0.1	0	0.1	V
Oscillator	f_{osc}	0.5	—	16	MHz
Bus frequency	f_{bus}	0.5	—	40	MHz
C parts Operating junction temperature range Operating ambient temperature range[2]	 T_J T_A	 –40 –40	 — 27	 100 85	°C
V parts Operating junction temperature range Operating ambient temperature range[2]	 T_J T_A	 –40 –40	 — 27	 120 105	°C
M parts Operating junction temperature range Operating ambient temperature range[2]	 T_J T_A	 –40 –40	 — 27	 140 125	°C

1 The device contains an internal voltage regulator to generate the logic and PLL supply out of the I/O supply. The absolute maximum ratings apply when this regulator is disabled and the device is powered from an external source.

2 Please refer to Section A.1.8, "Power Dissipation and Thermal Characteristics" for more details about the relation between ambient temperature T_A and device junction temperature T_J.