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	Obsolete
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	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	20K 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/product-detail/nxp-semiconductors/mc9s12xdt512val

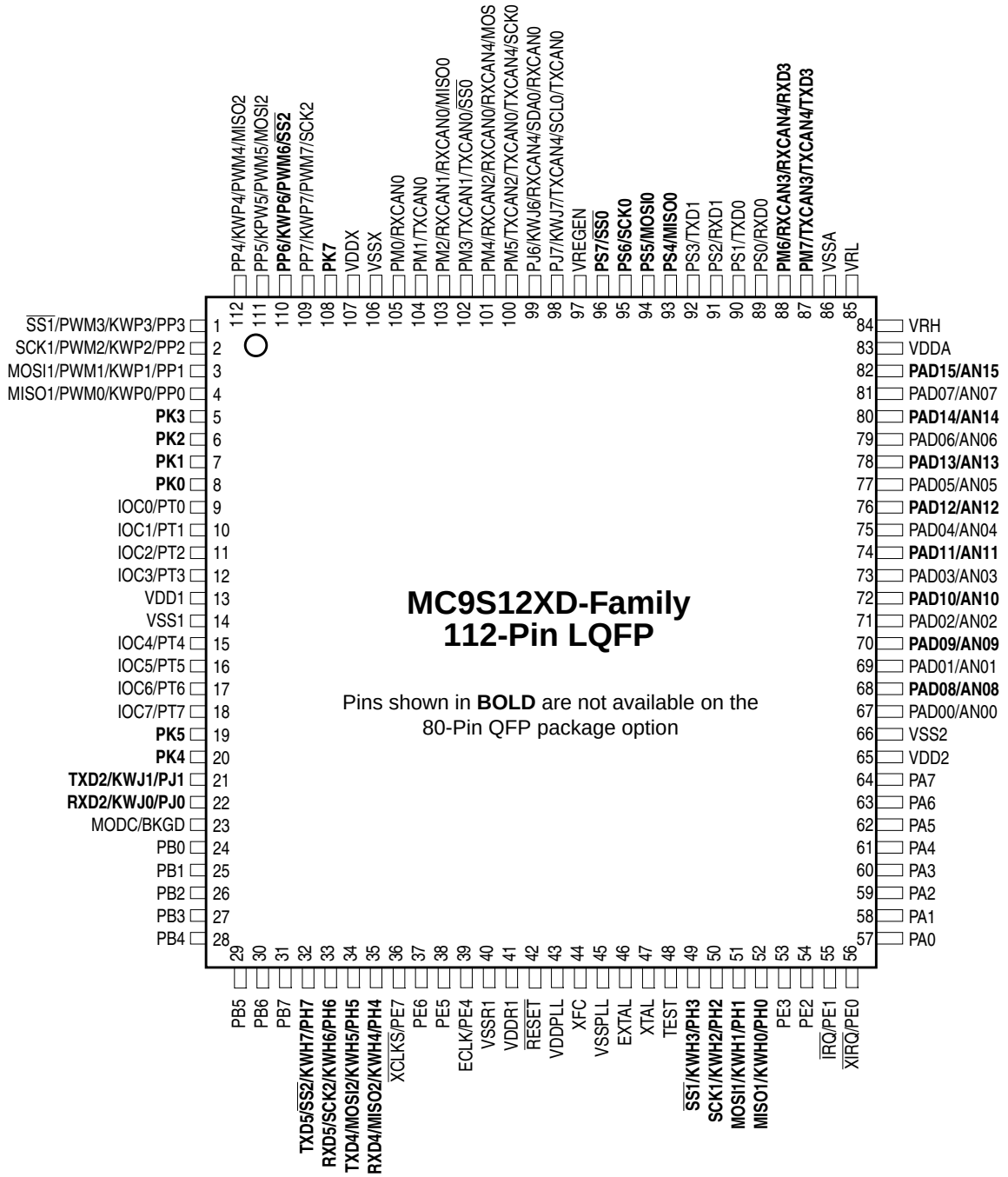

Figure 1-8. MC9S12XD Family Pin Assignments 112-Pin LQFP Package

2.2 External Signal Description

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

2.2.1 V_{DDPLL} and V_{SSPLL} — Operating and Ground Voltage Pins

These pins provide operating voltage (V_{DDPLL}) and ground (V_{SSPLL}) for the PLL circuitry. This allows the supply voltage to the PLL to be independently bypassed. Even if PLL usage is not required, V_{DDPLL} and V_{SSPLL} must be connected to properly.

2.2.2 XFC — External Loop Filter Pin

A passive external loop filter must be placed on the XFC pin. The filter is a second-order, low-pass filter that eliminates the VCO input ripple. The value of the external filter network and the reference frequency determines the speed of the corrections and the stability of the PLL. Refer to the device specification for calculation of PLL Loop Filter (XFC) components**.** If PLL usage is not required, the XFC pin must be tied to V_{DDPLL}.

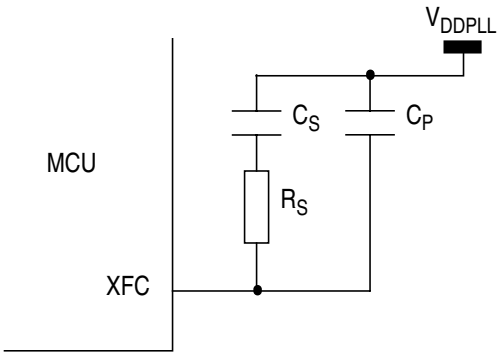

Figure 2-2. PLL Loop Filter Connections

2.2.3 $\overline{\text{RESET}}$ — Reset Pin

$\overline{\text{RESET}}$ is an active low bidirectional reset pin. As an input. it initializes the MCU asynchronously to a known start-up state. As an open-drain output, it indicates that a system reset (internal to the MCU) has been triggered.

2.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the CRG.

NOTE

Remember that in parallel to additional actions caused by self clock mode or clock monitor reset[1] handling the clock quality checker continues to check the OSCCLK signal.

The clock quality checker enables the PLL and the voltage regulator (VREG) anytime a clock check has to be performed. An ongoing clock quality check could also cause a running PLL (f_{SCM}) and an active VREG during pseudo stop mode or wait mode.

2.4.1.5 Computer Operating Properly Watchdog (COP)

The COP (free running watchdog timer) enables the user to check that a program is running and sequencing properly. When the COP is being used, software is responsible for keeping the COP from timing out. If the COP times out it is an indication that the software is no longer being executed in the intended sequence; thus a system reset is initiated (see Section 2.4.1.5, “Computer Operating Properly Watchdog (COP)”). The COP runs with a gated OSCCLK. Three control bits in the COPCTL register allow selection of seven COP time-out periods.

When COP is enabled, the program must write 0x_55 and 0x_AA (in this order) to the ARMCOP register during the selected time-out period. Once this is done, the COP time-out period is restarted. If the program fails to do this and the COP times out, the part will reset. Also, if any value other than 0x_55 or 0x_AA is written, the part is immediately reset.

Windowed COP operation is enabled by setting WCOP in the COPCTL register. In this mode, writes to the ARMCOP register to clear the COP timer must occur in the last 25% of the selected time-out period. A premature write will immediately reset the part.

If PCE bit is set, the COP will continue to run in pseudo stop mode.

2.4.1.6 Real Time Interrupt (RTI)

The RTI can be used to generate a hardware interrupt at a fixed periodic rate. If enabled (by setting RTIE = 1), this interrupt will occur at the rate selected by the RTICTL register. The RTI runs with a gated OSCCLK. At the end of the RTI time-out period the RTIF flag is set to 1 and a new RTI time-out period starts immediately.

A write to the RTICTL register restarts the RTI time-out period.

If the PRE bit is set, the RTI will continue to run in pseudo stop mode.

2.4.2 Operating Modes

2.4.2.1 Normal Mode

The CRG block behaves as described within this specification in all normal modes.

1. A Clock Monitor Reset will always set the SCME bit to logical 1.

AND — Logical AND — AND

Operation

RS1 & RS2 ⇒ RD
RD & IMM16 ⇒ RD (translates to ANDL RD, #IMM16[7:0]; ANDH RD, #IMM16[15:8])

Performs a bit wise logical AND of two 16 bit values and stores the result in the destination register RD.

Remark: There is no complement to the BITH and BITL functions. This can be imitated by using R0 as a destination register. AND R0, RS1, RS2 performs a bit wise test without storing a result.

CCR Effects

N	Z	V	C
Δ	Δ	0	—

N: Set if bit 15 of the result is set; cleared otherwise.
Z: Set if the result is $0000; cleared otherwise.
Refer to ANDH instruction for #IMM16 operations.
V: 0; cleared.
C: Not affected.

Code and CPU Cycles

Source Form	Address Mode	Machine Code										Cycles
AND RD, RS1, RS2	TRI	0	0	0	1	0	RD	RS1	RS2	0	0	P
AND RD, #IMM16	IMM8	1	0	0	0	0	RD	IMM16[7:0]				P
	IMM8	1	0	0	0	1	RD	IMM16[15:8]				P

LDH

Load Immediate 8 bit Constant (High Byte)

LDH

Operation

IMM8 ⇒ RD.H;

Loads an eight bit immediate constant into the high byte of register RD. The low byte is not affected.

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
LDH RD, #IMM8	IMM8	1	1	1	1	1	RD	IMM8	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

Table 7-18. PACTL Field Descriptions (continued)

Field	Description
5 PAMOD	**Pulse Accumulator Mode** — This bit is active only when the Pulse Accumulator A is enabled (PAEN = 1). 0 Event counter mode 1 Gated time accumulation mode
4 PEDGE	**Pulse Accumulator Edge Control** — This bit is active only when the Pulse Accumulator A is enabled (PAEN = 1). Refer to Table 7-19. For PAMOD bit = 0 (event counter mode). 0 Falling edges on PT7 pin cause the count to be incremented 1 Rising edges on PT7 pin cause the count to be incremented For PAMOD bit = 1 (gated time accumulation mode). 0 PT7 input pin high enables bus clock divided by 64 to Pulse Accumulator and the trailing falling edge on PT7 sets the PAIF flag. 1 PT7 input pin low enables bus clock divided by 64 to Pulse Accumulator and the trailing rising edge on PT7 sets the PAIF flag. If the timer is not active (TEN = 0 in TSCR1), there is no divide-by-64 since the ÷64 clock is generated by the timer prescaler.
3:2 CLK[1:0]	**Clock Select Bits** — For the description of PACLK please refer to Figure 7-70. If the pulse accumulator is disabled (PAEN = 0), the prescaler clock from the timer is always used as an input clock to the timer counter. The change from one selected clock to the other happens immediately after these bits are written. Refer to Table 7-20.
2 PAOVI	**Pulse Accumulator A Overflow Interrupt Enable** 0 Interrupt inhibited 1 Interrupt requested if PAOVF is set
0 PAI	**Pulse Accumulator Input Interrupt Enable** 0 Interrupt inhibited 1 Interrupt requested if PAIF is set

Table 7-19. Pin Action

PAMOD	PEDGE	Pin Action
0	0	Falling edge
0	1	Rising edge
1	0	Divide by 64 clock enabled with pin high level
1	1	Divide by 64 clock enabled with pin low level

Table 7-20. Clock Selection

CLK1	CLK0	Clock Source
0	0	Use timer prescaler clock as timer counter clock
0	1	Use PACLK as input to timer counter clock
1	0	Use PACLK/256 as timer counter clock frequency
1	1	Use PACLK/65536 as timer counter clock frequency

9.2 External Signal Description

The IICV2 module has two external pins.

9.2.1 IIC_SCL — Serial Clock Line Pin

This is the bidirectional serial clock line (SCL) of the module, compatible to the IIC bus specification.

9.2.2 IIC_SDA — Serial Data Line Pin

This is the bidirectional serial data line (SDA) of the module, compatible to the IIC bus specification.

9.3 Memory Map and Register Definition

This section provides a detailed description of all memory and registers for the IIC module.

9.3.1 Module Memory Map

The memory map for the IIC module is given below in Table 1-1. The address listed for each register is the address offset.The total address for each register is the sum of the base address for the IIC module and the address offset for each register.

Table 11-3. SCICR1 Field Descriptions (continued)

Field	Description
2 ILT	**Idle Line Type Bit** — ILT determines when the receiver starts counting logic 1s as idle character bits. The counting begins either after the start bit or after the stop bit. If the count begins after the start bit, then a string of logic 1s preceding the stop bit may cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. 0 Idle character bit count begins after start bit 1 Idle character bit count begins after stop bit
1 PE	**Parity Enable Bit** — PE enables the parity function. When enabled, the parity function inserts a parity bit in the most significant bit position. 0 Parity function disabled 1 Parity function enabled
0 PT	**Parity Type Bit** — PT determines whether the SCI generates and checks for even parity or odd parity. With even parity, an even number of 1s clears the parity bit and an odd number of 1s sets the parity bit. With odd parity, an odd number of 1s clears the parity bit and an even number of 1s sets the parity bit. 1 Even parity 1 Odd parity

Table 11-4. Loop Functions

LOOPS	RSRC	Function
0	x	Normal operation
1	0	Loop mode with transmitter output internally connected to receiver input
1	1	Single-wire mode with TXD pin connected to receiver input

12.3.2.5 SPI Data Register (SPIDR)

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

Figure 12-7. SPI Data Register (SPIDR)

Read: Anytime; normally read only when SPIF is set

Write: Anytime

The SPI data register is both the input and output register for SPI data. A write to this register allows a data byte to be queued and transmitted. For an SPI configured as a master, a queued data byte is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag SPTEF in the SPISR register indicates when the SPI data register is ready to accept new data.

Received data in the SPIDR is valid when SPIF is set.

If SPIF is cleared and a byte has been received, the received byte is transferred from the receive shift register to the SPIDR and SPIF is set.

If SPIF is set and not serviced, and a second byte has been received, the second received byte is kept as valid byte in the receive shift register until the start of another transmission. The byte in the SPIDR does not change.

If SPIF is set and a valid byte is in the receive shift register, and SPIF is serviced before the start of a third transmission, the byte in the receive shift register is transferred into the SPIDR and SPIF remains set (see Figure 12-8).

If SPIF is set and a valid byte is in the receive shift register, and SPIF is serviced after the start of a third transmission, the byte in the receive shift register has become invalid and is not transferred into the SPIDR (see Figure 12-9).

Chapter 15
Background Debug Module (S12XBDMV2)

15.1 Introduction

This section describes the functionality of the background debug module (BDM) sub-block of the HCS12X core platform.

The background debug module (BDM) sub-block is a single-wire, background debug system implemented in on-chip hardware for minimal CPU intervention. All interfacing with the BDM is done via the BKGD pin.

The BDM has enhanced capability for maintaining synchronization between the target and host while allowing more flexibility in clock rates. This includes a sync signal to determine the communication rate and a handshake signal to indicate when an operation is complete. The system is backwards compatible to the BDM of the S12 family with the following exceptions:

- TAGGO command no longer supported by BDM
- External instruction tagging feature now part of DBG module
- BDM register map and register content extended/modified
- Global page access functionality
- Enabled but not active out of reset in emulation modes
- CLKSW bit set out of reset in emulation mode.
- Family ID readable from firmware ROM at global address 0x7FFF0F (value for HCS12X devices is 0xC1)

15.1.1 Features

The BDM includes these distinctive features:

- Single-wire communication with host development system
- Enhanced capability for allowing more flexibility in clock rates
- SYNC command to determine communication rate
- GO_UNTIL command
- Hardware handshake protocol to increase the performance of the serial communication
- Active out of reset in special single chip mode
- Nine hardware commands using free cycles, if available, for minimal CPU intervention
- Hardware commands not requiring active BDM
- 14 firmware commands execute from the standard BDM firmware lookup table

15.4 Functional Description

The BDM receives and executes commands from a host via a single wire serial interface. There are two types of BDM commands: hardware and firmware commands.

Hardware commands are used to read and write target system memory locations and to enter active background debug mode, see Section 15.4.3, “BDM Hardware Commands”. Target system memory includes all memory that is accessible by the CPU.

Firmware commands are used to read and write CPU resources and to exit from active background debug mode, see Section 15.4.4, “Standard BDM Firmware Commands”. The CPU resources referred to are the accumulator (D), X index register (X), Y index register (Y), stack pointer (SP), and program counter (PC).

Hardware commands can be executed at any time and in any mode excluding a few exceptions as highlighted (see Section 15.4.3, “BDM Hardware Commands”) and in secure mode (see Section 15.4.1, “Security”). Firmware commands can only be executed when the system is not secure and is in active background debug mode (BDM).

15.4.1 Security

If the user resets into special single chip mode with the system secured, a secured mode BDM firmware lookup table is brought into the map overlapping a portion of the standard BDM firmware lookup table. The secure BDM firmware verifies that the on-chip EEPROM and Flash EEPROM are erased. This being the case, the UNSEC and ENBDM bit will get set. The BDM program jumps to the start of the standard BDM firmware and the secured mode BDM firmware is turned off and all BDM commands are allowed. If the EEPROM or Flash do not verify as erased, the BDM firmware sets the ENBDM bit, without asserting UNSEC, and the firmware enters a loop. This causes the BDM hardware commands to become enabled, but does not enable the firmware commands. This allows the BDM hardware to be used to erase the EEPROM and Flash.

BDM operation is not possible in any other mode than special single chip mode when the device is secured. The device can only be unsecured via BDM serial interface in special single chip mode. For more information regarding security, please see the S12X_9SEC Block Guide.

15.4.2 Enabling and Activating BDM

The system must be in active BDM to execute standard BDM firmware commands. BDM can be activated only after being enabled. BDM is enabled by setting the ENBDM bit in the BDM status (BDMSTS) register. The ENBDM bit is set by writing to the BDM status (BDMSTS) register, via the single-wire interface, using a hardware command such as WRITE_BD_BYTE.

18.4 Functional Description

The MMC block performs several basic functions of the S12X sub-system operation: MCU operation modes, priority control, address mapping, select signal generation and access limitations for the system. Each aspect is described in the following subsections.

18.4.1 MCU Operating Mode

- Normal single-chip mode

  There is no external bus in this mode. The MCU program is executed from the internal memory and no external accesses are allowed.
- Special single-chip mode

  This mode is generally used for debugging single-chip operation, boot-strapping or security related operations. The active background debug mode is in control of the CPU code execution and the BDM firmware is waiting for serial commands sent through the BKGD pin. There is no external bus in this mode.
- Emulation single-chip mode

  Tool vendors use this mode for emulation systems in which the user's target application is normal single-chip mode. Code is executed from external or internal memory depending on the set-up of the EROMON bit (see Section 18.3.2.5, "MMC Control Register (MMCCTL1)). The external bus is active in both cases to allow observation of internal operations (internal visibility).

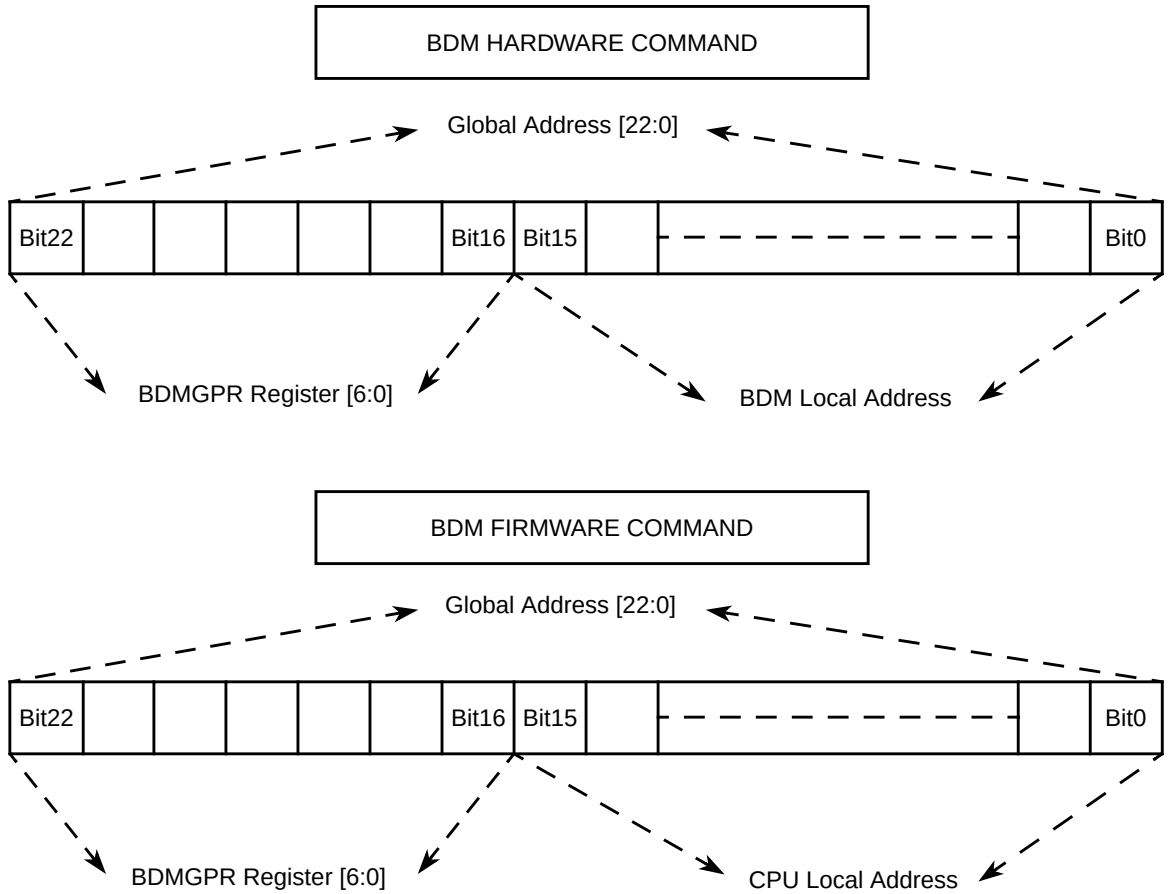

Figure 18-22. BDMGPR Address Mapping

Chapter 21
External Bus Interface (S12XEBIV2)

21.1 Introduction

This document describes the functionality of the XEBI block controlling the external bus interface.

The XEBI controls the functionality of a non-multiplexed external bus (a.k.a. ‘expansion bus’) in relationship with the chip operation modes. Dependent on the mode, the external bus can be used for data exchange with external memory, peripherals or PRU, and provide visibility to the internal bus externally in combination with an emulator.

21.1.1 Features

The XEBI includes the following features:

- Output of up to 23-bit address bus and control signals to be used with a non-muxed external bus
- Bidirectional 16-bit external data bus with option to disable upper half
- Visibility of internal bus activity

21.1.2 Modes of Operation

- Single-chip modes

  The external bus interface is not available in these modes.
- Expanded modes

  Address, data, and control signals are activated on the external bus in normal expanded mode and special test mode.
- Emulation modes

  The external bus is activated to interface to an external tool for emulation of normal expanded mode or normal single-chip mode applications.

Refer to the S12X_MMC section for a detailed description of the MCU operating modes.

22.3.2.38 Port P Data Register (PTP)

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

Figure 22-40. Port P Data Register (PTP)

Read: Anytime.

Write: Anytime.

Port P pins 7, and 5–0 are associated with the PWM as well as the SPI1 and SPI2 modules. These pins can be used as general purpose I/O when not used with any of the peripherals.

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

The PWM function takes precedence over the general purpose I/O and the SPI2 or SPI1 function if the associated PWM channel is enabled. While channels 6 and 5-0 are output only if the respective channel is enabled, channel 7 can be PWM output or input if the shutdown feature is enabled. *Refer to PWM section for details.*

The SPI2 function takes precedence over the general purpose I/O function if enabled. *Refer to SPI section for details.*

The SPI1 function takes precedence over the general purpose I/O function if enabled. *Refer to SPI section for details.*

22.3.2.56 Port J Data Direction Register (DDRJ)

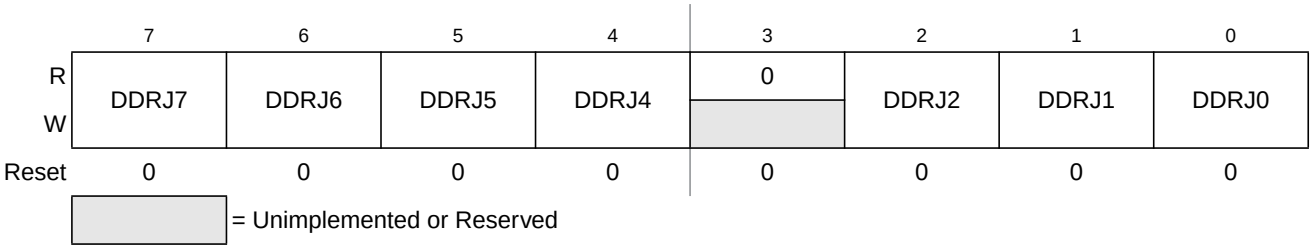

Figure 22-58. Port J Data Direction Register (DDRJ)

Read: Anytime.

Write: Anytime.

This register configures each port J pin as either input or output.

The CAN forces the I/O state to be an output on PJ7 (TXCAN4) and an input on pin PJ6 (RXCAN4). The IIC takes control of the I/O if enabled. In these cases the data direction bits will not change.

The SCI2 forces the I/O state to be an output for each port line associated with an enabled output (TXD2). It also forces the I/O state to be an input for each port line associated with an enabled input (RXD2). In these cases the data direction bits will not change.

The DDRJ bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled.

Table 22-52. DDRJ Field Descriptions

Field	Description
7–0 DDRJ[7:4] DDRJ[2:0]	**Data Direction Port J** 0 Associated pin is configured as input. 1 Associated pin is configured as output. **Note:** Due to internal synchronization circuits, it can take up to 2 bus clock cycles until the correct value is read on PTJ or PTIJ registers, when changing the DDRJ register.

External Signal Description

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

23.0.3 Signal Properties

Table 23-1 shows all the pins and their functions that are controlled by the PIM. *Refer to Section , "Functional Description" for the availability of the individual pins in the different package options.*

NOTE

If there is more than one function associated with a pin, the priority is indicated by the position in the table from top (highest priority) to bottom (lowest priority).

Table 23-1. Pin Functions and Priorities (Sheet 1 of 7)

Port	Pin Name	Pin Function and Priority	I/O	Description	Pin Function after Reset
—	BKGD	MODC[1]	I	MODC input during $\overline{\text{RESET}}$	BKGD
		BKGD	I/O	S12X_BDM communication pin	
A	PA[7:0]	ADDR[15:8] mux IVD[15:8][2]	O	High-order external bus address output (multiplexed with IVIS data)	Mode dependent[3]
		GPIO	I/O	General-purpose I/O	
B	PB[7:1]	ADDR[7:1] mux IVD[7:1][2]	O	Low-order external bus address output (multiplexed with IVIS data)	Mode dependent[3]
		GPIO	I/O	General-purpose I/O	
	PB[0]	ADDR[0] mux IVD0[2]	O	Low-order external bus address output (multiplexed with IVIS data)	
		$\overline{\text{UDS}}$	O	Upper data strobe	
		GPIO	I/O	General-purpose I/O	
C	PC[7:0]	DATA[15:8]	I/O	High-order bidirectional data input/output Configurable for reduced input threshold	Mode dependent[3]
		GPIO	I/O	General-purpose I/O	
D	PD[7:0]	DATA[7:0]	I/O	Low-order bidirectional data input/output Configurable for reduced input threshold	Mode dependent[3]
		GPIO	I/O	General-purpose I/O	

25.4.1.2 Command Write Sequence

The EEPROM command controller is used to supervise the command write sequence to execute program, erase, erase verify, sector erase abort, and sector modify algorithms.

Before starting a command write sequence, the ACCERR and PVIOL flags in the ESTAT register must be clear (see Section 25.3.2.6, “EEPROM Status Register (ESTAT)”) and the CBEIF flag should be tested to determine the state of the address, data and command buffers. If the CBEIF flag is set, indicating the buffers are empty, a new command write sequence can be started. If the CBEIF flag is clear, indicating the buffers are not available, a new command write sequence will overwrite the contents of the address, data and command buffers.

A command write sequence consists of three steps which must be strictly adhered to with writes to the EEPROM module not permitted between the steps. However, EEPROM register and array reads are allowed during a command write sequence. The basic command write sequence is as follows:

1. Write to one address in the EEPROM memory.
2. Write a valid command to the ECMD register.
3. Clear the CBEIF flag in the ESTAT register by writing a 1 to CBEIF to launch the command.

The address written in step 1 will be stored in the EADDR registers and the data will be stored in the EDATA registers. If the CBEIF flag in the ESTAT register is clear when the first EEPROM array write occurs, the contents of the address and data buffers will be overwritten and the CBEIF flag will be set. When the CBEIF flag is cleared, the CCIF flag is cleared on the same bus cycle by the EEPROM command controller indicating that the command was successfully launched. For all command write sequences except sector erase abort, the CBEIF flag will set four bus cycles after the CCIF flag is cleared indicating that the address, data, and command buffers are ready for a new command write sequence to begin. For sector erase abort operations, the CBEIF flag will remain clear until the operation completes. Except for the sector erase abort command, a buffered command will wait for the active operation to be completed before being launched. The sector erase abort command is launched when the CBEIF flag is cleared as part of a sector erase abort command write sequence. Once a command is launched, the completion of the command operation is indicated by the setting of the CCIF flag in the ESTAT register. The CCIF flag will set upon completion of all active and buffered commands.

25.4.2 EEPROM Commands

Table 25-9 summarizes the valid EEPROM commands along with the effects of the commands on the EEPROM block.

Table 25-9. EEPROM Command Description

ECMDB	Command	Function on EEPROM Memory
0x05	Erase Verify	Verify all memory bytes in the EEPROM block are erased. If the EEPROM block is erased, the BLANK flag in the ESTAT register will set upon command completion.
0x20	Program	Program a word (two bytes) in the EEPROM block.
0x40	Sector Erase	Erase all four memory bytes in a sector of the EEPROM block.

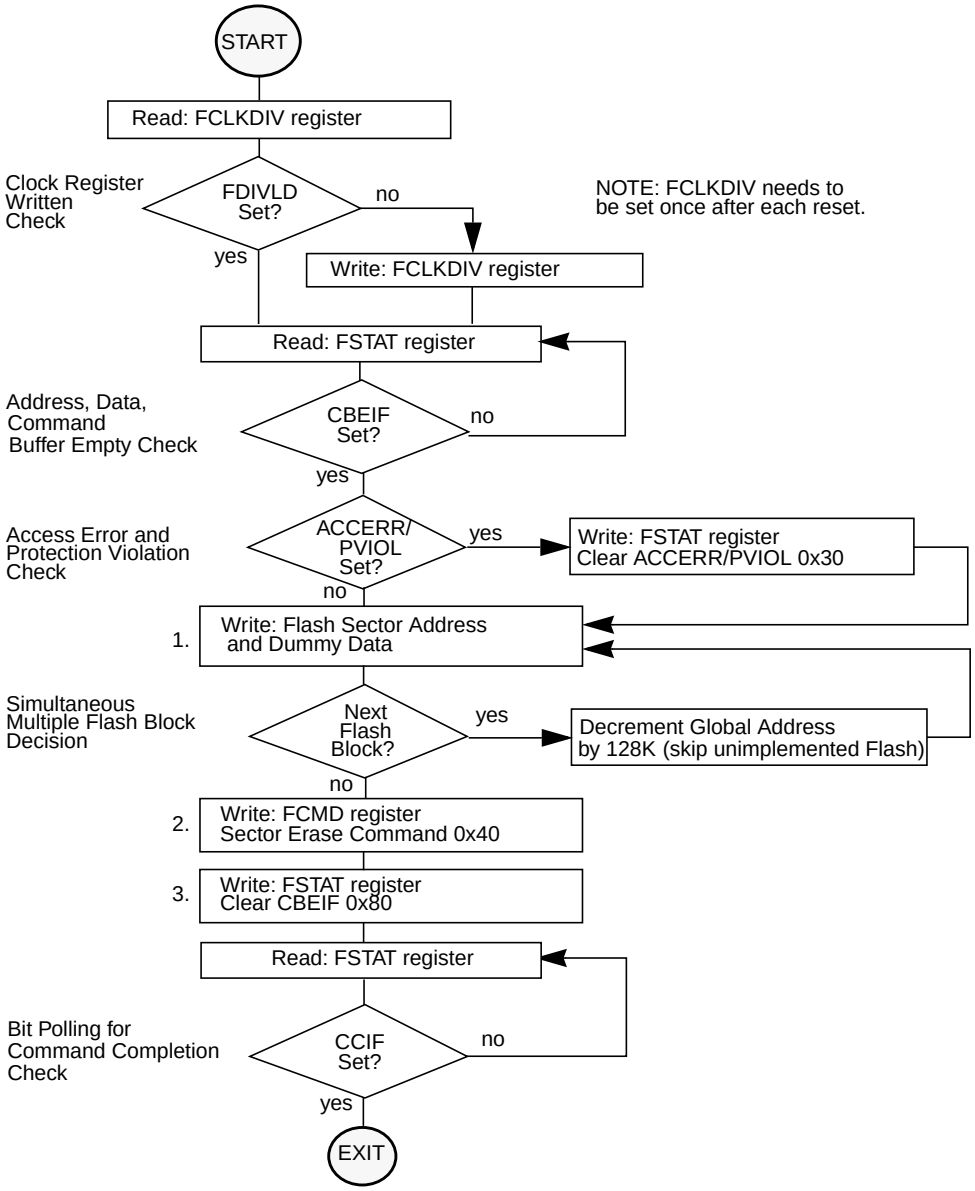

Figure 28-29. Example Sector Erase Command Flow