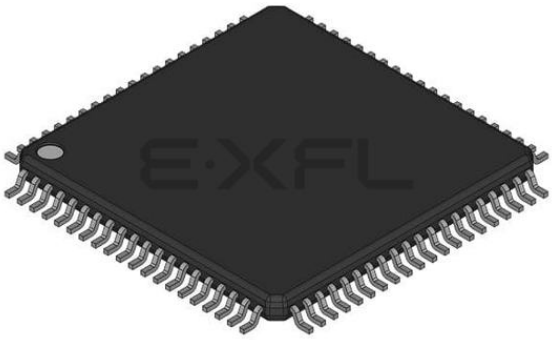


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	HCS12X
Core Size	16-Bit
Speed	80MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	59
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	4K 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/pro/item?MUrl=&PartUrl=mc9s12xd64maa

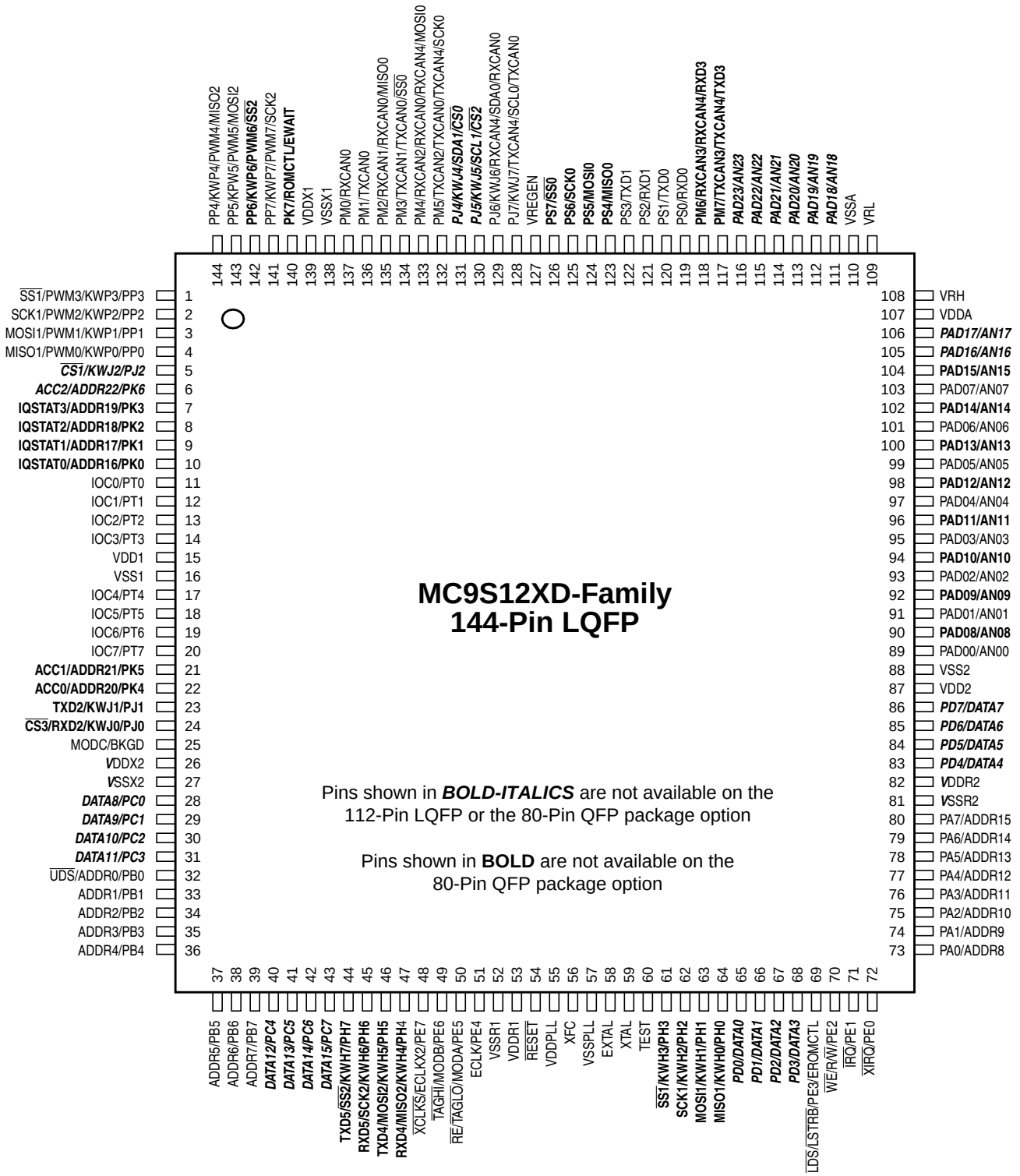


Figure 1-7. MC9S12XD Family Pin Assignment 144-Pin LQFP Package

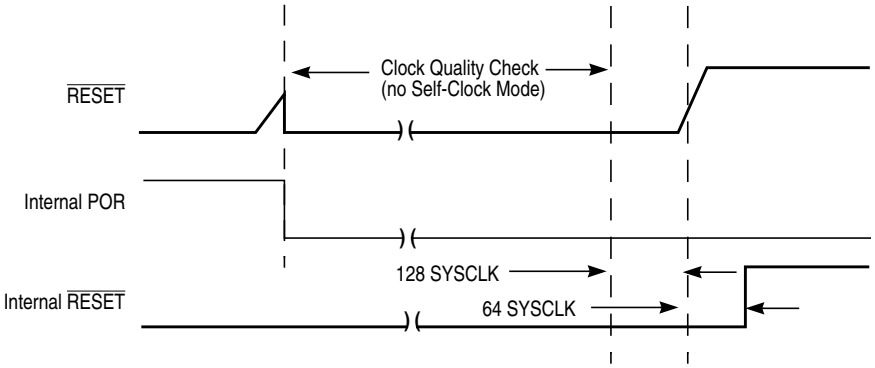


Figure 2-26. $\overline{\text{RESET}}$ Pin Tied to V_{DD} (by a pull-up resistor)

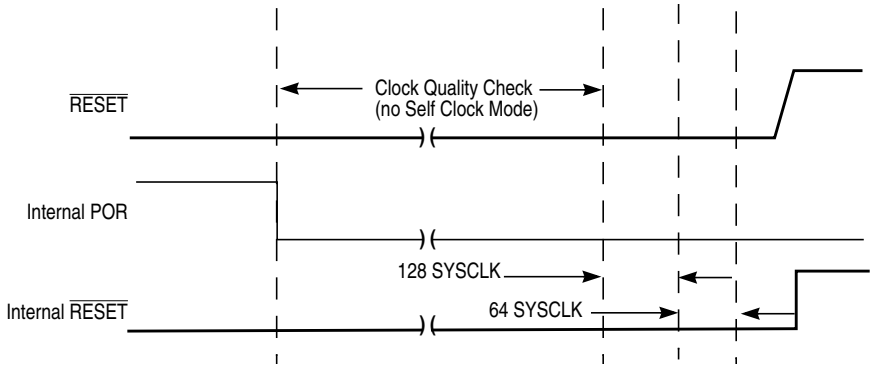


Figure 2-27. $\overline{\text{RESET}}$ Pin Held Low Externally

2.6 Interrupts

The interrupts/reset vectors requested by the CRG are listed in Table 2-16. Refer to MCU specification for related vector addresses and priorities.

Table 2-16. CRG Interrupt Vectors

Interrupt Source	CCR Mask	Local Enable
Real time interrupt	I bit	CRGINT (RTIE)
LOCK interrupt	I bit	CRGINT (LOCKIE)
SCM interrupt	I bit	CRGINT (SCMIE)

2.6.1 Real Time Interrupt

The CRG generates a real time interrupt when the selected interrupt time period elapses. RTI interrupts are locally disabled by setting the RTIE bit to 0. The real time interrupt flag (RTIF) is set to1 when a timeout occurs, and is cleared to 0 by writing a 1 to the RTIF bit.

The RTI continues to run during pseudo stop mode if the PRE bit is set to 1. This feature can be used for periodic wakeup from pseudo stop if the RTI interrupt is enabled.

Table 4-18. ATDSTAT0 Field Descriptions (continued)

Field	Description
4 FIFOR	**FIFO Over Run Flag** — This bit indicates that a result register has been written to before its associated conversion complete flag (CCF) has been cleared. This flag is most useful when using the FIFO mode because the flag potentially indicates that result registers are out of sync with the input channels. However, it is also practical for non-FIFO modes, and indicates that a result register has been over written before it has been read (i.e., the old data has been lost). This flag is cleared when one of the following occurs: • Write “1” to FIFOR • Start a new conversion sequence (write to ATDCTL5 or external trigger) 0 No over run has occurred 1 Overrun condition exists (result register has been written while associated CCFx flag remained set)
3:0 CC[3:0}	**Conversion Counter** — These 4 read-only bits are the binary value of the conversion counter. The conversion counter points to the result register that will receive the result of the current conversion. For example, CC3 = 0, CC2 = 1, CC1 = 1, CC0 = 0 indicates that the result of the current conversion will be in ATD Result Register 6. If in non-FIFO mode (FIFO = 0) the conversion counter is initialized to zero at the begin and end of the conversion sequence. If in FIFO mode (FIFO = 1) the register counter is not initialized. The conversion counters wraps around when its maximum value is reached. Aborting a conversion or starting a new conversion by write to an ATDCTL register (ATDCTL5-0) clears the conversion counter even if FIFO=1.

5.3.2.6 ATD Control Register 5 (ATDCTL5)

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

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

= Unimplemented or Reserved

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

Read: Anytime

Write: Anytime

Table 5-13. ATDCTL5 Field Descriptions

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

CPCH — Compare Immediate 8 bit Constant with Carry (High Byte) — CPCH

Operation

RS.H - IMM8 - C ⇒ NONE, only condition code flags get updated

Subtracts the carry bit and the 8 bit constant IMM8 contained in the instruction code from the high byte of the source register RD using binary subtraction and updates the condition code register accordingly. The carry bit and Zero bits are taken into account to allow a 16 bit compare in the form of

```
CMPL    R2,#LOWBYTE
CPCH    R2,#HIGHBYTE
BCC                         ; branch condition
```

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

CCR Effects

N	Z	V	C
Δ	Δ	Δ	Δ

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

Z: Set if the result is $00 and Z was set before this operation; cleared otherwise.

V: Set if a two´s complement overflow resulted from the operation; cleared otherwise.
RS[15] & $\overline{\text{IMM8[7]}}$ & $\overline{\text{result[15]}}$ | $\overline{\text{RS[15]}}$ & IMM8[7] & result[15]

C: Set if there is a carry from the bit 15 of the result; cleared otherwise.
$\overline{\text{RS[15]}}$ & IMM8[7] | $\overline{\text{RS[15]}}$ & result[15] | IMM8[7] & result[15]

Code and CPU Cycles

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

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
PWMCNT0	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT1	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT2	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT3	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT4	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT5	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT6	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMCNT7	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
PWMPER0	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMPER1	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMPER2	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMPER3	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMPER4	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMPER5	R W	Bit 7	6	5	4	3	2	1	Bit 0
PWMPER6	R W	Bit 7	6	5	4	3	2	1	Bit 0

= Unimplemented or Reserved

Figure 8-2. PWM Register Summary (Sheet 2 of 3)

On the front end of the PWM timer, the clock is enabled to the PWM circuit by the PWMEx bit being high. There is an edge-synchronizing circuit to guarantee that the clock will only be enabled or disabled at an edge. When the channel is disabled (PWMEx = 0), the counter for the channel does not count.

8.4.2.2 PWM Polarity

Each channel has a polarity bit to allow starting a waveform cycle with a high or low signal. This is shown on the block diagram as a mux select of either the Q output or the $\overline{Q}$ output of the PWM output flip flop. When one of the bits in the PWMPOL register is set, the associated PWM channel output is high at the beginning of the waveform, then goes low when the duty count is reached. Conversely, if the polarity bit is zero, the output starts low and then goes high when the duty count is reached.

8.4.2.3 PWM Period and Duty

Dedicated period and duty registers exist for each channel and are double buffered so that if they change while the channel is enabled, the change will NOT take effect until one of the following occurs:

- The effective period ends
- The counter is written (counter resets to $00)
- The channel is disabled

In this way, the output of the PWM will always be either the old waveform or the new waveform, not some variation in between. If the channel is not enabled, then writes to the period and duty registers will go directly to the latches as well as the buffer.

A change in duty or period can be forced into effect “immediately” by writing the new value to the duty and/or period registers and then writing to the counter. This forces the counter to reset and the new duty and/or period values to be latched. In addition, since the counter is readable, it is possible to know where the count is with respect to the duty value and software can be used to make adjustments

NOTE

When forcing a new period or duty into effect immediately, an irregular PWM cycle can occur.

Depending on the polarity bit, the duty registers will contain the count of either the high time or the low time.

8.4.2.4 PWM Timer Counters

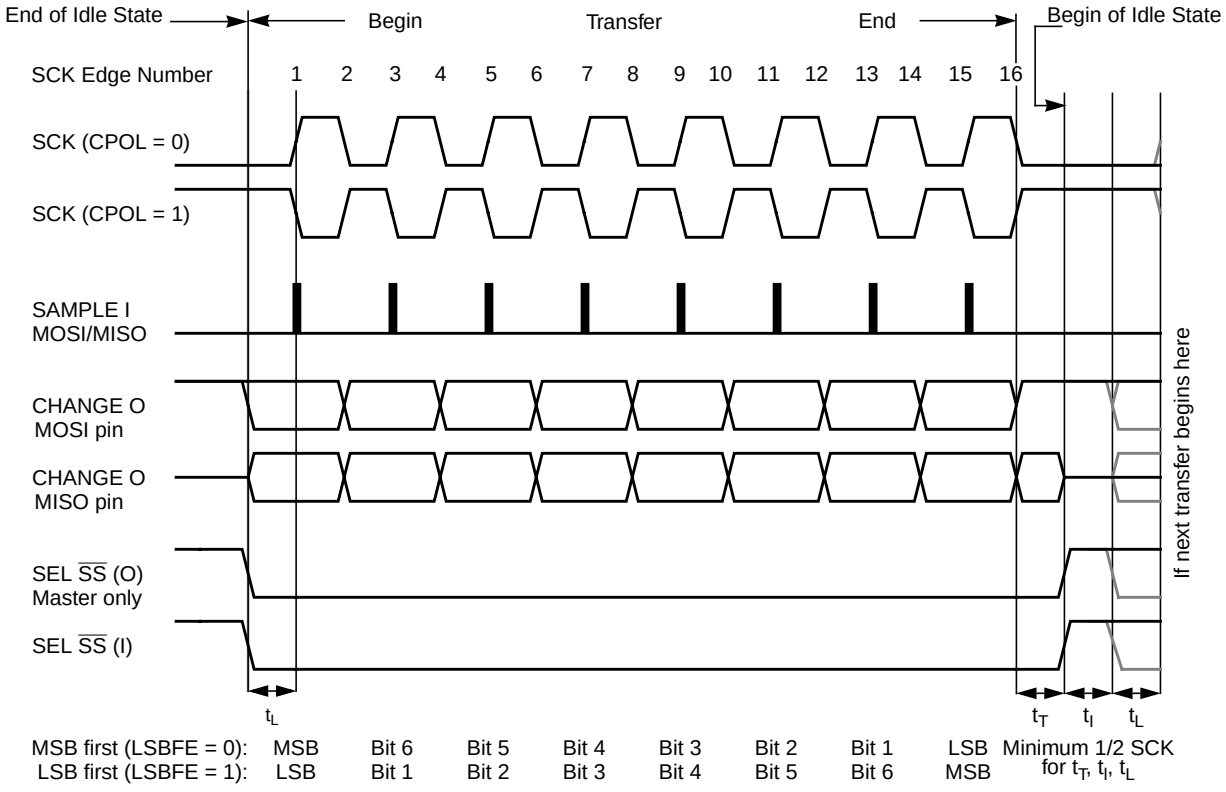
Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source (see Section 8.4.1, “PWM Clock Select” for the available clock sources and rates). The counter compares to two registers, a duty register and a period register as shown in Figure 8-19. When the PWM counter matches the duty register, the output flip-flop changes state, causing the PWM waveform to also change state. A match between the PWM counter and the period register behaves differently depending on what output mode is selected as shown in Figure 8-19 and described in Section 8.4.2.5, “Left Aligned Outputs” and Section 8.4.2.6, “Center Aligned Outputs”.

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)

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

Expansion of the BDM Local Address Map

PPAGE, RPAGE, and EPAGE registers are also used for the expansion of the BDM local address to the global address. These registers can be read and written by the BDM.

The BDM expansion scheme is the same as the CPU expansion scheme.

17.4.2.2 Global Addresses Based on the Global Page

CPU Global Addresses Based on the Global Page

The seven global page index bits allow access to the full 8 Mbyte address map that can be accessed with 23 address bits. This provides an alternative way to access all of the various pages of FLASH, RAM and EEPROM as well as additional external memory.

The GPAGE Register is used only when the CPU is executing a global instruction (see Section 1.3.2.3, “Global Page Index Register (GPAGE)”). The generated global address is the result of concatenation of the CPU local address [15:0] with the GPAGE register [22:16] (see Figure 1-7).

BDM Global Addresses Based on the Global Page

The seven BDMGPR Global Page index bits allow access to the full 8 Mbyte address map that can be accessed with 23 address bits. This provides an alternative way to access all of the various pages of FLASH, RAM and EEPROM as well as additional external memory.

The BDM global page index register (BDMGPR) is used only in the case the CPU is executing a firmware command which uses a global instruction (like GLDD, GSTD) or by a BDM hardware command (like WRITE_W, WRITE_BYTE, READ_W, READ_BYTE). See the BDM Block Guide for further details.

The generated global address is a result of concatenation of the BDM local address with the BDMGPR register [22:16] in the case of a hardware command or concatenation of the CPU local address and the BDMGPR register [22:16] in the case of a firmware command (see Figure 1-22).

18.1.4.2 Functional Modes

- Single chip modes

  In normal and special single chip mode the internal memory is used. External bus is not active.
- Expanded modes

  Address, data, and control signals are activated in normal expanded and special test modes when accessing the external bus. Access to internal resources will not cause activity on the external bus.
- Emulation modes

  External bus is active to emulate, via an external tool, the normal expanded or the normal single chip mode.

18.1.5 Block Diagram

Figure 18-1[1] shows a block diagram of the MMC.

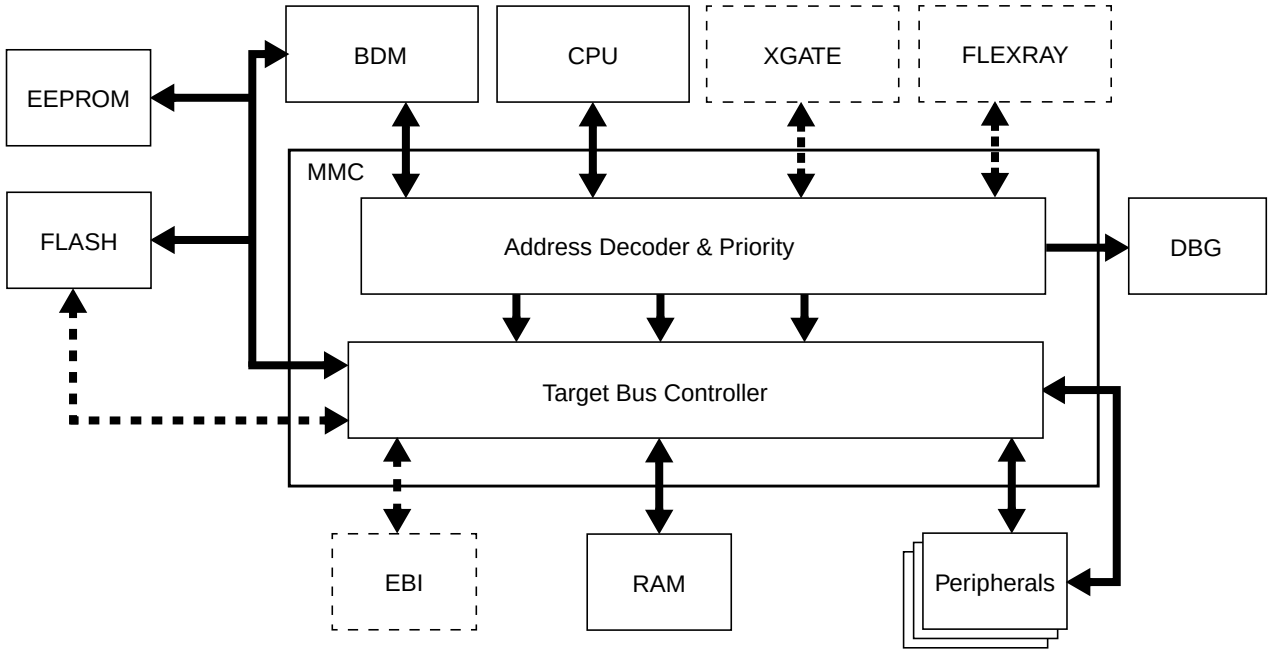


Figure 18-1. MMC Block Diagram

18.2 External Signal Description

The user is advised to refer to the SoC Guide for port configuration and location of external bus signals. Some pins may not be bonded out in all implementations.

Table 18-2 and Table 18-3 outline the pin names and functions. It also provides a brief description of their operation.

1. Doted blocks and lines are optional. Please refer to the Device User Guide for their availlibilities.

20.3.2.8.2 Debug Comparator Address High Register (DBGXAH)

Address: 0x0029

	7	6	5	4	3	2	1	0
R	0	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 20-15. Debug Comparator Address High Register (DBGXAH)

Read: Anytime

Write: Anytime when S12XDBG not armed.

Table 20-29. DBGXAH Field Descriptions

Field	Description
6–0 Bit[22:16]	**Comparator Address High Compare Bits** — The Comparator address high compare bits control whether the selected comparator will compare the address bus bits [22:16] to a logic one or logic zero. This register byte is ignored for XGATE compares. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

20.3.2.8.3 Debug Comparator Address Mid Register (DBGXAM)

Address: 0x002A

	7	6	5	4	3	2	1	0
R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
W								
Reset	0	0	0	0	0	0	0	0

Figure 20-16. Debug Comparator Address Mid Register (DBGXAM)

Read: Anytime

Write: Anytime when S12XDBG not armed.

Table 20-30. DBGXAM Field Descriptions

Field	Description
7–0 Bit[15:8]	**Comparator Address Mid Compare Bits** — The Comparator address mid compare bits control whether the selected comparator will compare the address bus bits [15:8] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

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	

Table 24-1. Pin Functions and Priorities (Sheet 2 of 5)

Port	Pin Name	Pin Function and Priority	I/O	Description	Pin Function after Reset
S	PS7	$\overline{SS0}$	I/O	Serial Peripheral Interface 0 slave select output in master mode, input in slave mode or master mode.	GPIO
		GPIO	I/O	General-purpose I/O	
	PS6	SCK0	I/O	Serial Peripheral Interface 0 serial clock pin	
		GPIO	I/O	General-purpose I/O	
	PS5	MOSI0	I/O	Serial Peripheral Interface 0 master out/slave in pin	
		GPIO	I/O	General-purpose I/O	
	PS4	MISO0	I/O	Serial Peripheral Interface 0 master in/slave out pin	
		GPIO	I/O	General-purpose I/O	
	PS3	TXD1	O	Serial Communication Interface 1 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PS2	RXD1	I	Serial Communication Interface 1 receive pin	
		GPIO	I/O	General-purpose I/O	
	PS1	TXD0	O	Serial Communication Interface 0 transmit pin	
		GPIO	I/O	General-purpose I/O	
	PS0	RXD0	I	Serial Communication Interface 0 receive pin	
		GPIO	I/O	General-purpose I/O	

Table 24-38. PPSP Field Descriptions

Field	Description
7–0 PPSP[7:0]	**Polarity Select Port P** 0 Falling edge on the associated port P pin sets the associated flag bit in the PIFP register.A pull-up device is connected to the associated port P pin, if enabled by the associated bit in register PERP and if the port is used as input. 1 Rising edge on the associated port P pin sets the associated flag bit in the PIFP register.A pull-down device is connected to the associated port P pin, if enabled by the associated bit in register PERP and if the port is used as input.

24.0.5.40 Port P Interrupt Enable Register (PIEP)

	7	6	5	4	3	2	1	0
R W	PIEP7	PIEP6	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
Reset	0	0	0	0	0	0	0	0

Figure 24-42. Port P Interrupt Enable Register (PIEP)

Read: Anytime.

Write: Anytime.

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

Table 24-39. PIEP Field Descriptions

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

24.0.5.41 Port P Interrupt Flag Register (PIFP)

	7	6	5	4	3	2	1	0
R W	PIFP7	PIFP6	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
Reset	0	0	0	0	0	0	0	0

Figure 24-43. Port P Interrupt Flag Register (PIFP)

Read: Anytime.

Write: Anytime.

Each flag is set by an active edge on the associated input pin. This could be a rising or a falling edge based on the state of the PPSP register. To clear this flag, write logic level “1” to the corresponding bit in the PIFP register. Writing a “0” has no effect.

Table 25-6. ESTAT Field Descriptions

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

25.4.2.2 Program Command

The program operation will program a previously erased word in the EEPROM memory using an embedded algorithm.

An example flow to execute the program operation is shown in Figure 25-19. The program command write sequence is as follows:

1. Write to an EEPROM block address to start the command write sequence for the program command. The data written will be programmed to the address written.
2. Write the program command, 0x20, to the ECMD register.
3. Clear the CBEIF flag in the ESTAT register by writing a 1 to CBEIF to launch the program command.

If a word to be programmed is in a protected area of the EEPROM memory, the PVIOL flag in the ESTAT register will set and the program command will not launch. Once the program command has successfully launched, the CCIF flag in the ESTAT register will set after the program operation has completed unless a new command write sequence has been buffered.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
RESERVED1	R	0	0	0	0	0	0	0	0
	W								
RESERVED2	R	0	0	0	0	0	0	0	0
	W								
RESERVED3	R	0	0	0	0	0	0	0	0
	W								
RESERVED4	R	0	0	0	0	0	0	0	0
	W								

Figure 29-3. FTX128K1 Register Summary (continued)

29.3.2.1 Flash Clock Divider Register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

	7	6	5	4	3	2	1	0
R	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 29-4. Flash Clock Divider Register (FCLKDIV)

All bits in the FCLKDIV register are readable, bits 6-0 are write once and bit 7 is not writable.

Table 29-2. FCLKDIV Field Descriptions

Field	Description
7 FDIVLD	**Clock Divider Loaded.** 0 Register has not been written. 1 Register has been written to since the last reset.
6 PRDIV8	**Enable Prescalar by 8.** 0 The oscillator clock is directly fed into the clock divider. 1 The oscillator clock is divided by 8 before feeding into the clock divider.
5:0 FDIV[5:0]	**Clock Divider Bits** — The combination of PRDIV8 and FDIV[5:0] must divide the oscillator clock down to a frequency of 150 kHz–200 kHz. The maximum divide ratio is 512. Please refer to Section 29.4.1.1, “Writing the FCLKDIV Register” for more information.

29.3.2.2 Flash Security Register (FSEC)

The FSEC register holds all bits associated with the security of the MCU and Flash module.

29.4.2.2.1 Data Compress Operation

The Flash module contains a 16-bit multiple-input signature register (MISR) to generate a 16-bit signature based on selected Flash array data. The final 16-bit signature, found in the FDATA registers after the data compress operation has completed, is based on the following logic equation which is executed on every data compression cycle during the operation:

MISR[15:0] = {MISR[14:0], ^MISR[15,4,2,1]} ^ DATA[15:0] ***Eqn. 29-1***

where MISR is the content of the internal signature register and DATA is the data to be compressed as shown in Figure 29-25.

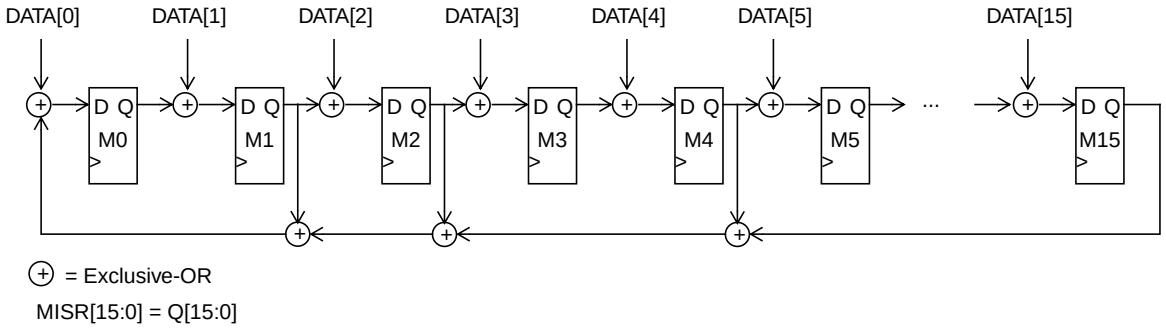


Figure 29-25. 16-Bit MISR Diagram

During the data compress operation, the following steps are executed:

1. MISR is reset to 0xFFFF.
2. Initialized DATA equal to 0xFFFF is compressed into the MISR which results in the MISR containing 0x0001.
3. DATA equal to the selected Flash array data range is read and compressed into the MISR with addresses incrementing.
4. DATA equal to the selected Flash array data range is read and compressed into the MISR with addresses decrementing.
5. DATA equal to the contents of the MISR is compressed into the same MISR.
6. The contents of the MISR are written to the FDATA registers.