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	40MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	44
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12xs128mae

Chapter 1
Device Overview S12XS Family

1.1 Introduction

The new S12XS family of 16-bit micro controllers is a compatible, reduced version of the S12XE family. These families provide an easy approach to develop common platforms from low-end to high-end applications, minimizing the redesign of software and hardware.

Targeted at generic automotive applications and CAN nodes, some typical examples of these applications are: Body Controllers, Occupant Detection, Door Modules, RKE Receivers, Smart Actuators, Lighting Modules and Smart Junction Boxes amongst many others.

The S12XS family retains many of the features of the S12XE family including Error Correction Code (ECC) on Flash memory, a separate Data-Flash Module for code or data storage, a Frequency Modulated Locked Loop (IPLL) that improves the EMC performance and a fast ATD converter.

S12XS family delivers 32-bit performance with all the advantages and efficiencies of a 16-bit MCU while retaining the low cost, power consumption, EMC and code-size efficiency advantages currently enjoyed by users of Freescale's existing 16-bit S12 and S12X MCU families. Like members of other S12X families, the S12XS family runs 16-bit wide accesses without wait states for all peripherals and memories.

The S12XS family is available in 112-pin LQFP, 80-pin QFP, 64-pin LQFP package options and maintains a high level of pin compatibility with the S12XE family. In addition to the I/O ports available in each module, up to 18 further I/O ports are available with interrupt capability allowing Wake-Up from stop or wait modes.

The peripheral set includes MSCAN, SPI, two SCIs, an 8-channel 24-bit periodic interrupt timer, 8-channel 16-bit Timer, 8-channel PWM and up to 16- channel 12-bit ATD converter.

Software controlled peripheral-to-port routing enables access to a flexible mix of the peripheral modules in the lower pin count package options.

1.1.1 Features

Features of the S12XS Family are listed here. Please see Table D-1 for memory options and Table D-2 for the peripheral features that are available on the different family members.

- 16-bit CPU12X
  - Upward compatible with S12 instruction set with the exception of five Fuzzy instructions (MEM, WAV, WAVR, REV, REVW) which have been removed
  - Enhanced indexed addressing
  - Access to large data segments independent of PPAGE

NOTE

This document assumes the availability of all features (112-pin package option). Some functions are not available on lower pin count package options. Refer to the pin-out summary section.

2.1.2 Features

The Port Integration Module includes these distinctive registers:

- Data and data direction registers for Ports A, B, E, K, T, S, M, P, H, J, and AD when used as general-purpose I/O
- Control registers to enable/disable pull-device and select pull-ups/pull-downs on Ports T, S, M, P, H, and J on per-pin basis
- Control registers to enable/disable pull-up devices on Port AD on per-pin basis
- Single control register to enable/disable pull-ups on Ports A, B, E, and K on per-port basis and on BKGD pin
- Control registers to enable/disable reduced output drive on Ports T, S, M, P, H, J, and AD on per-pin basis
- Single control register to enable/disable reduced output drive on Ports A, B, E, and K on per-port basis
- Control registers to enable/disable open-drain (wired-or) mode on Ports S, and M
- Interrupt flag register for pin interrupts on Ports P, H, and J
- Control register to configure $\overline{\text{IRQ}}$ pin operation
- Routing registers to support module port relocation
- Free-running clock outputs

A standard port pin has the following minimum features:

- Input/output selection
- 5V output drive with two selectable drive strengths
- 5V digital and analog input
- Input with selectable pull-up or pull-down device

Optional features supported on dedicated pins:

- Open drain for wired-or connections
- Interrupt inputs with glitch filtering

2.2 External Signal Description

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

Table 2-1 shows all the pins and their functions that are controlled by the Port Integration Module. *Refer to the device definition for the availability of the individual pins in the different package options.*

Table 2-3. Pin Configuration Summary

DDR	IO	RDR	PE	PS[1]	IE[2]	Function	Pull Device	Interrupt
0	x	x	0	x	0	Input	Disabled	Disabled
0	x	x	1	0	0	Input	Pull Up	Disabled
0	x	x	1	1	0	Input	Pull Down	Disabled
0	x	x	0	0	1	Input	Disabled	Falling edge
0	x	x	0	1	1	Input	Disabled	Rising edge
0	x	x	1	0	1	Input	Pull Up	Falling edge
0	x	x	1	1	1	Input	Pull Down	Rising edge
1	0	0	x	x	0	Output, full drive to 0	Disabled	Disabled
1	1	0	x	x	0	Output, full drive to 1	Disabled	Disabled
1	0	1	x	x	0	Output, reduced drive to 0	Disabled	Disabled
1	1	1	x	x	0	Output, reduced drive to 1	Disabled	Disabled
1	0	0	x	0	1	Output, full drive to 0	Disabled	Falling edge
1	1	0	x	1	1	Output, full drive to 1	Disabled	Rising edge
1	0	1	x	0	1	Output, reduced drive to 0	Disabled	Falling edge
1	1	1	x	1	1	Output, reduced drive to 1	Disabled	Rising edge

1 Always “0” on Port A, B, E, K, and AD.

2 Applicable only on Port P, H, and J.

NOTE

All register bits in this module are completely synchronous to internal clocks during a register read.

NOTE

Figures of port data registers also display the alternative functions if applicable on the related pin as defined in Table 2-1. Names in brackets denote the availability of the function when using a specific routing option.

NOTE

Figures of module routing registers also display the module instance or module channel associated with the related routing bit.

2.3.71 Port AD0 Reduced Drive Register 1 (RDR1AD0)

Address 0x0275 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	RDR1AD07	RDR1AD06	RDR1AD05	RDR1AD04	RDR1AD03	RDR1AD02	RDR1AD01	RDR1AD00
Reset	0	0	0	0	0	0	0	0

Figure 2-69. Port AD0 Reduced Drive Register 1 (RDR1AD0)

[1] Read: Anytime
Write: Anytime

Table 2-68. RDR1AD0 Register Field Descriptions

Field	Description
7-0 RDR1AD0	**Port AD0 reduced drive**—Select reduced drive for output pin This bit configures the drive strength of the associated output pin as either full or reduced. If a pin is used as input this bit has no effect. The reduced drive function is independent of which function is being used on a particular pin. 1 Reduced drive selected (approx. 1/5 of the full drive strength) 0 Full drive strength enabled

2.3.72 Port AD0 Pull Up Enable Register 0 (PER0AD0)

Address 0x0276 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PER0AD07	PER0AD06	PER0AD05	PER0AD04	PER0AD03	PER0AD02	PER0AD01	PER0AD00
Reset	0	0	0	0	0	0	0	0

Figure 2-70. Port AD0 Pull Device Up Register 0 (PER0AD0)

[1] Read: Anytime
Write: Anytime

Table 2-69. PER0AD0 Register Field Descriptions

Field	Description
7-0 PER0AD0	**Port AD0 pull device enable**—Enable pull-up device on input pin This bit controls whether a pull device on the associated port input pin is active. If a pin is used as output this bit has no effect. The polarity is selected by the related polarity select register bit. 1 Pull device enabled 0 Pull device disabled

5.4.3 BDM Hardware Commands

Hardware commands are used to read and write target system memory locations and to enter active background debug mode. Target system memory includes all memory that is accessible by the CPU on the SOC which can be on-chip RAM, non-volatile memory (e.g. EEPROM, Flash EEPROM), I/O and control registers, and all external memory.

Hardware commands are executed with minimal or no CPU intervention and do not require the system to be in active BDM for execution, although, they can still be executed in this mode. When executing a hardware command, the BDM sub-block waits for a free bus cycle so that the background access does not disturb the running application program. If a free cycle is not found within 128 clock cycles, the CPU is momentarily frozen so that the BDM can steal a cycle. When the BDM finds a free cycle, the operation does not intrude on normal CPU operation provided that it can be completed in a single cycle. However, if an operation requires multiple cycles the CPU is frozen until the operation is complete, even though the BDM found a free cycle.

The BDM hardware commands are listed in Table 5-6.

The READ_BD and WRITE_BD commands allow access to the BDM register locations. These locations are not normally in the system memory map but share addresses with the application in memory. To distinguish between physical memory locations that share the same address, BDM memory resources are enabled just for the READ_BD and WRITE_BD access cycle. This allows the BDM to access BDM locations unobtrusively, even if the addresses conflict with the application memory map.

Table 5-6. Hardware Commands

Command	Opcode (hex)	Data	Description
BACKGROUND	90	None	Enter background mode if firmware is enabled. If enabled, an ACK will be issued when the part enters active background mode.
ACK_ENABLE	D5	None	Enable Handshake. Issues an ACK pulse after the command is executed.
ACK_DISABLE	D6	None	Disable Handshake. This command does not issue an ACK pulse.
READ_BD_BYTE	E4	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table in map. Odd address data on low byte; even address data on high byte.
READ_BD_WORD	EC	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table in map. Must be aligned access.
READ_BYTE	E0	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table out of map. Odd address data on low byte; even address data on high byte.
READ_WORD	E8	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table out of map. Must be aligned access.
WRITE_BD_BYTE	C4	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table in map. Odd address data on low byte; even address data on high byte.
WRITE_BD_WORD	CC	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table in map. Must be aligned access.
WRITE_BYTE	C0	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table out of map. Odd address data on low byte; even address data on high byte.

within a few percent of the actual target speed and the communication protocol can easily tolerate speed errors of several percent.

As soon as the SYNC request is detected by the target, any partially received command or bit retrieved is discarded. This is referred to as a soft-reset, equivalent to a time-out in the serial communication. After the SYNC response, the target will consider the next negative edge (issued by the host) as the start of a new BDM command or the start of new SYNC request.

Another use of the SYNC command pulse is to abort a pending ACK pulse. The behavior is exactly the same as in a regular SYNC command. Note that one of the possible causes for a command to not be acknowledged by the target is a host-target synchronization problem. In this case, the command may not have been understood by the target and so an ACK response pulse will not be issued.

5.4.10 Instruction Tracing

When a TRACE1 command is issued to the BDM in active BDM, the CPU exits the standard BDM firmware and executes a single instruction in the user code. Once this has occurred, the CPU is forced to return to the standard BDM firmware and the BDM is active and ready to receive a new command. If the TRACE1 command is issued again, the next user instruction will be executed. This facilitates stepping or tracing through the user code one instruction at a time.

If an interrupt is pending when a TRACE1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. Once back in standard BDM firmware execution, the program counter points to the first instruction in the interrupt service routine.

Be aware when tracing through the user code that the execution of the user code is done step by step but all peripherals are free running. Hence possible timing relations between CPU code execution and occurrence of events of other peripherals no longer exist.

Do not trace the CPU instruction BGND used for soft breakpoints. Tracing the BGND instruction will result in a return address pointing to BDM firmware address space.

When tracing through user code which contains stop or wait instructions the following will happen when the stop or wait instruction is traced:

The CPU enters stop or wait mode and the TRACE1 command can not be finished before leaving the low power mode. This is the case because BDM active mode can not be entered after CPU executed the stop instruction. However all BDM hardware commands except the BACKGROUND command are operational after tracing a stop or wait instruction and still being in stop or wait mode. If system stop mode is entered (all bus masters are in stop mode) no BDM command is operational.

As soon as stop or wait mode is exited the CPU enters BDM active mode and the saved PC value points to the entry of the corresponding interrupt service routine.

In case the handshake feature is enabled the corresponding ACK pulse of the TRACE1 command will be discarded when tracing a stop or wait instruction. Hence there is no ACK pulse when BDM active mode is entered as part of the TRACE1 command after CPU exited from stop or wait mode. All valid commands sent during CPU being in stop or wait mode or after CPU exited from stop or wait mode will have an ACK pulse. The handshake feature becomes disabled only when system stop mode has been reached. Hence

Chapter 6
S12X Debug (S12XDBGV3) Module

Table 6-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V03.20	14 Sep 2007	6.3.2.7/6-205	- Clarified reserved State Sequencer encodings.
V03.21	23 Oct 2007	6.4.2.2/6-218 6.4.2.4/6-219	- Added single databyte comparison limitation information - Added statement about interrupt vector fetches whilst tagging.
V03.22	12 Nov 2007	6.4.5.2/6-223 6.4.5.5/6-227	- Removed LOOP1 tracing restriction NOTE. - Added pin reset effect NOTE.
V03.23	13 Nov 2007	General	- Text readability improved, typo removed.
V03.24	04 Jan 2008	6.4.5.3/6-225	- Corrected bit name.
V03.25	14 May 2008	General	- Updated Revision History Table format. Corrected other paragraph formats.
V03.26	12 Sep 2012	General	- Added missing full stops. Removed redundant quotation marks.

6.1 Introduction

The S12XDBG module provides an on-chip trace buffer with flexible triggering capability to allow non-intrusive debug of application software. The S12XDBG module is optimized for the S12X 16-bit architecture and allows debugging of CPU12X module operations.

Typically the S12XDBG module is used in conjunction with the S12XBDM module, whereby the user configures the S12XDBG module for a debugging session over the BDM interface. Once configured the S12XDBG module is armed and the device leaves BDM Mode returning control to the user program, which is then monitored by the S12XDBG module. Alternatively the S12XDBG module can be configured over a serial interface using SWI routines.

6.1.1 Glossary

Table 6-2. Glossary Of Terms

Term	Definition
COF	Change Of Flow. Change in the program flow due to a conditional branch, indexed jump or interrupt
BDM	Background Debug Mode
DUG	Device User Guide, describing the features of the device into which the DBG is integrated
WORD	16-bit data entity

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0019 PWMPER5	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x001A PWMPER6	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x001B PWMPER7	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x001C PWMDTY0	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x001D PWMDTY1	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x001E PWMDTY2	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x001F PWMDTY3	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0010 PWMDTY4	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0021 PWMDTY5	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0022 PWMDTY6	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0023 PWMDTY7	R W	Bit 7	6	5	4	3	2	1	Bit 0
0x0024 PWMSDN	R W	PWMIF	PWMIE	0 PWMRSTRT	PWMLVL	0	PWM7IN	PWM7INL	PWM7ENA

= Unimplemented or Reserved

Figure 13-2. PWM Register Summary (Sheet 3 of 3)

1 Intended for factory test purposes only.

13.3.2.1 PWM Enable Register (PWME)

Each PWM channel has an enable bit (PWMEx) to start its waveform output. When any of the PWMEx bits are set (PWMEx = 1), the associated PWM output is enabled immediately. However, the actual PWM waveform is not available on the associated PWM output until its clock source begins its next cycle due to the synchronization of PWMEx and the clock source.

As an example of a center aligned output, consider the following case:

Clock Source = E, where E = 10 MHz (100 ns period)

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10 MHz/8 = 1.25 MHz

PWMx Period = 800 ns

PWMx Duty Cycle = 3/4 *100% = 75%

Shown in Figure 13-23 is the output waveform generated.

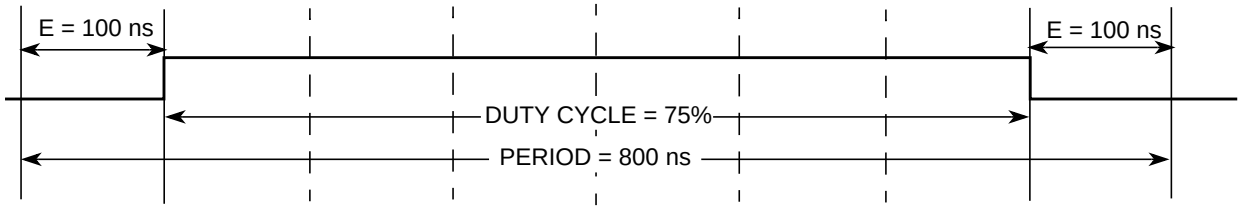


Figure 13-23. PWM Center Aligned Output Example Waveform

13.4.2.7 PWM 16-Bit Functions

The PWM timer also has the option of generating 8-channels of 8-bits or 4-channels of 16-bits for greater PWM resolution. This 16-bit channel option is achieved through the concatenation of two 8-bit channels.

The PWMCTL register contains four control bits, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. Channels 6 and 7 are concatenated with the CON67 bit, channels 4 and 5 are concatenated with the CON45 bit, channels 2 and 3 are concatenated with the CON23 bit, and channels 0 and 1 are concatenated with the CON01 bit.

NOTE

Change these bits only when both corresponding channels are disabled.

When channels 6 and 7 are concatenated, channel 6 registers become the high order bytes of the double byte channel, as shown in Figure 13-24. Similarly, when channels 4 and 5 are concatenated, channel 4 registers become the high order bytes of the double byte channel. When channels 2 and 3 are concatenated, channel 2 registers become the high order bytes of the double byte channel. When channels 0 and 1 are concatenated, channel 0 registers become the high order bytes of the double byte channel.

When using the 16-bit concatenated mode, the clock source is determined by the low order 8-bit channel clock select control bits. That is channel 7 when channels 6 and 7 are concatenated, channel 5 when channels 4 and 5 are concatenated, channel 3 when channels 2 and 3 are concatenated, and channel 1 when channels 0 and 1 are concatenated. The resulting PWM is output to the pins of the corresponding low order 8-bit channel as also shown in Figure 13-24. The polarity of the resulting PWM output is controlled by the PPOLx bit of the corresponding low order 8-bit channel as well.

14.4.3 Data Format

The SCI uses the standard NRZ mark/space data format. When Infrared is enabled, the SCI uses RZI data format where zeroes are represented by light pulses and ones remain low. See Figure 14-15 below.

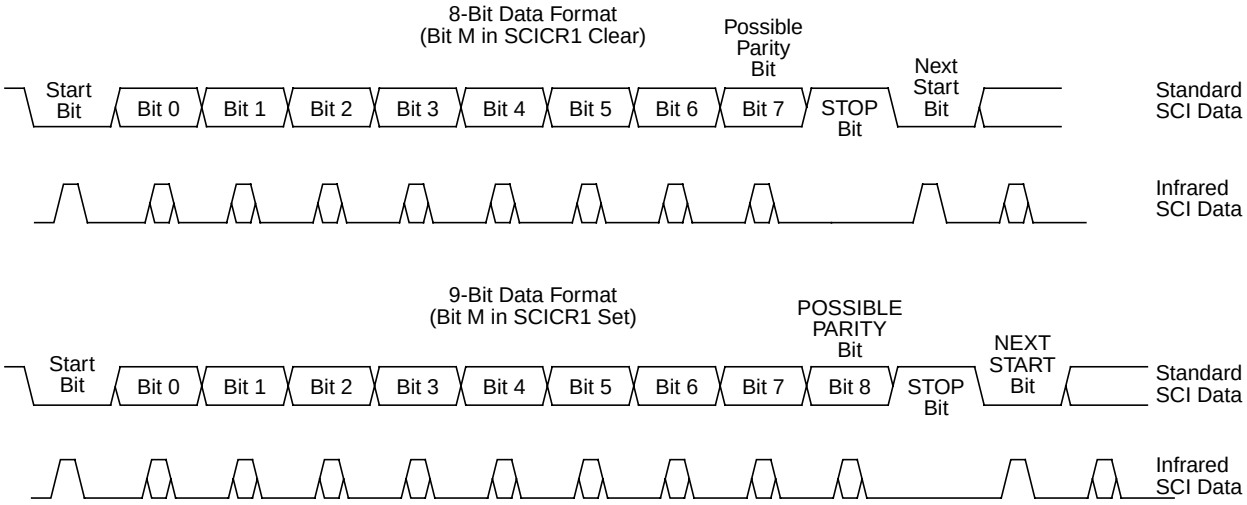


Figure 14-15. SCI Data Formats

Each data character is contained in a frame that includes a start bit, eight or nine data bits, and a stop bit. Clearing the M bit in SCI control register 1 configures the SCI for 8-bit data characters. A frame with eight data bits has a total of 10 bits. Setting the M bit configures the SCI for nine-bit data characters. A frame with nine data bits has a total of 11 bits.

Table 14-14. Example of 8-Bit Data Formats

Start Bit	Data Bits	Address Bits	Parity Bits	Stop Bit
1	8	0	0	1
1	7	0	1	1
1	7	1[1]	0	1

1 The address bit identifies the frame as an address character. See Section 14.4.6.6, “Receiver Wakeup”.

When the SCI is configured for 9-bit data characters, the ninth data bit is the T8 bit in SCI data register high (SCIDRH). It remains unchanged after transmission and can be used repeatedly without rewriting it. A frame with nine data bits has a total of 11 bits.

Table 14-15. Example of 9-Bit Data Formats

Start Bit	Data Bits	Address Bits	Parity Bits	Stop Bit
1	9	0	0	1
1	8	0	1	1
1	8	1[1]	0	1

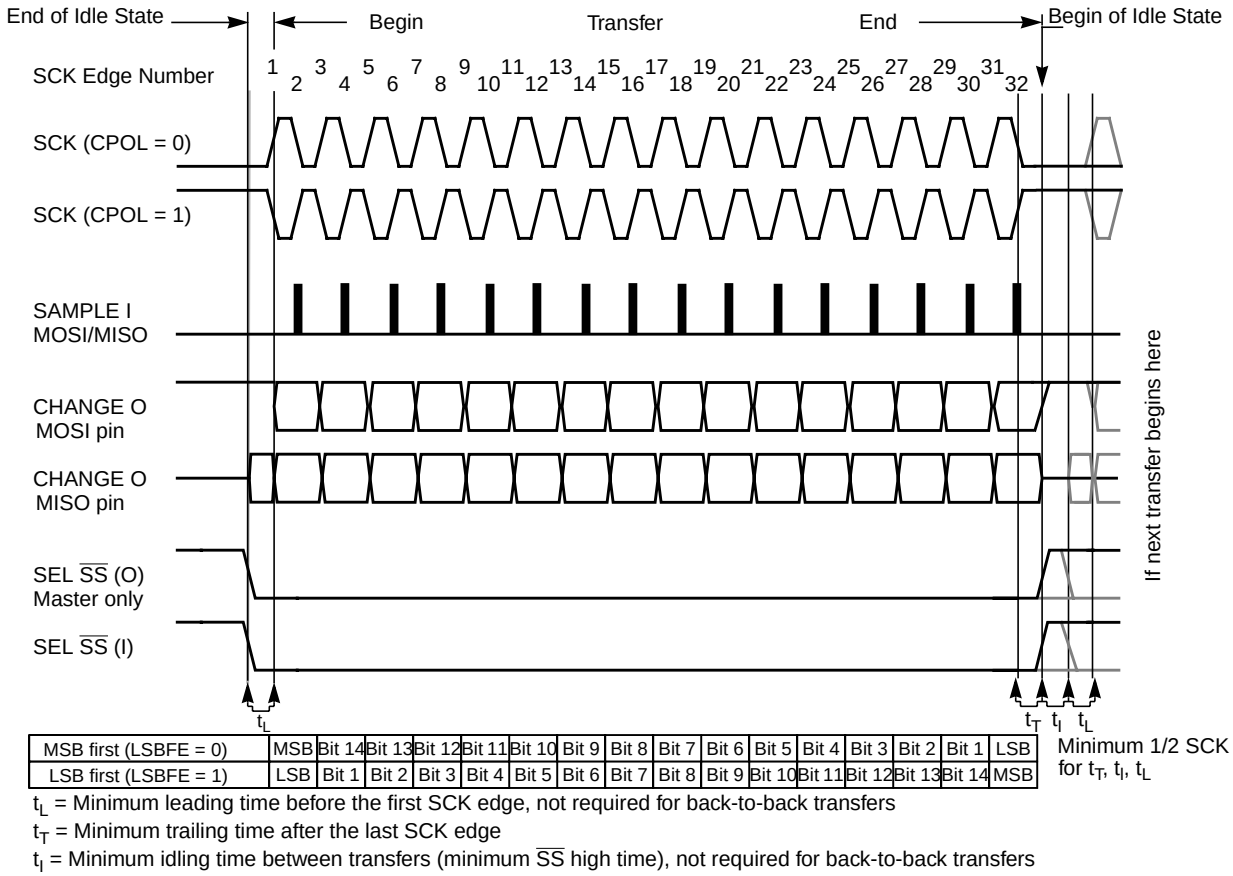


Figure 15-15. SPI Clock Format 1 (CPHA = 1), with 16-Bit Transfer Width selected (XFRW = 1)

The $\overline{SS}$ line can remain active low between successive transfers (can be tied low at all times). This format is sometimes preferred in systems having a single fixed master and a single slave that drive the MISO data line.

- Back-to-back transfers in master mode

  In master mode, if a transmission has completed and new data is available in the SPI data register, this data is sent out immediately without a trailing and minimum idle time.

The SPI interrupt request flag (SPIF) is common to both the master and slave modes. SPIF gets set one half SCK cycle after the last SCK edge.

15.4.4 SPI Baud Rate Generation

Baud rate generation consists of a series of divider stages. Six bits in the SPI baud rate register (SPPR2, SPPR1, SPPR0, SPR2, SPR1, and SPR0) determine the divisor to the SPI module clock which results in the SPI baud rate.

The SPI clock rate is determined by the product of the value in the baud rate preselection bits (SPPR2–SPPR0) and the value in the baud rate selection bits (SPR2–SPR0). The module clock divisor equation is shown in Equation 15-3.

$$\text{BaudRateDivisor} = (\text{SPPR} + 1) \bullet 2^{(\text{SPR} + 1)} \qquad \textit{Eqn. 15-3}$$

Read: Anytime

Write: Anytime.

Table 16-11. TCTL3/TCTL4 Field Descriptions

Field	Description
7:0 EDGnB EDGnA	**Input Capture Edge Control** — These eight pairs of control bits configure the input capture edge detector circuits.

Table 16-12. Edge Detector Circuit Configuration

EDGnB	EDGnA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge (rising or falling)

16.3.2.10 Timer Interrupt Enable Register (TIE)

Module Base + 0x000C

	7	6	5	4	3	2	1	0
R W	C7I	C6I	C5I	C4I	C3I	C2I	C1I	C0I
Reset	0	0	0	0	0	0	0	0

Figure 16-18. Timer Interrupt Enable Register (TIE)

Read: Anytime

Write: Anytime.

Table 16-13. TIE Field Descriptions

Field	Description
7:0 C7I:C0I	**Input Capture/Output Compare "x" Interrupt Enable —** The bits in TIE correspond bit-for-bit with the bits in the TFLG1 status register. If cleared, the corresponding flag is disabled from causing a hardware interrupt. If set, the corresponding flag is enabled to cause a interrupt.

Table 16-19. Pin Action

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

NOTE

If the timer is not active (TEN = 0 in TSCR), there is no divide-by-64 because the ÷64 clock is generated by the timer prescaler.

Table 16-20. Timer Clock Selection

CLK1	CLK0	Timer Clock
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

For the description of PACLK please refer Figure 16-30**.**

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.

16.3.2.16 Pulse Accumulator Flag Register (PAFLG)

Module Base + 0x0021

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PAOVF	PAIF
W								
Reset	0	0	0	0	0	0	0	0

Unimplemented or Reserved

Figure 16-25. Pulse Accumulator Flag Register (PAFLG)

Read: Anytime

Write: Anytime

When the TFFCA bit in the TSCR register is set, any access to the PACNT register will clear all the flags in the PAFLG register. Timer module or Pulse Accumulator must stay enabled (TEN=1 or PAEN=1) while clearing these bits.

Chapter 18
256 KByte Flash Module (S12XFTMR256K1V1)

Table 18-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V01.04	03 Jan 2008		- Cosmetic changes
V01.05	19 Dec 2008	18.1/18-507 18.4.2.4/18-542 18.4.2.6/18-544 18.4.2.11/18-547 18.4.2.11/18-547 18.4.2.11/18-547	- Clarify single bit fault correction for P-Flash phrase - Add statement concerning code runaway when executing Read Once, Program Once, and Verify Backdoor Access Key commands from Flash block containing associated fields - Relate Key 0 to associated Backdoor Comparison Key address - Change “power down reset” to “reset” in Section 18.4.2.11
V01.06	25 Sep 2009	18.3.2/18-514 18.3.2.1/18-516 18.4.1.2/18-536 18.6/18-556	The following changes were made to clarify module behavior related to Flash register access during reset sequence and while Flash commands are active: - Add caution concerning register writes while command is active - Writes to FCLKDIV are allowed during reset sequence while CCIF is clear - Add caution concerning register writes while command is active - Writes to FCCOBIX, FCCOBHI, FCCOBLO registers are ignored during reset sequence

18.1 Introduction

The FTMR256K1 module implements the following:

- 256 Kbytes of P-Flash (Program Flash) memory
- 8 Kbytes of D-Flash (Data Flash) memory

The Flash memory is ideal for single-supply applications allowing for field reprogramming without requiring external high voltage sources for program or erase operations. The Flash module includes a memory controller that executes commands to modify Flash memory contents. The user interface to the memory controller consists of the indexed Flash Common Command Object (FCCOB) register which is written to with the command, global address, data, and any required command parameters. The memory controller must complete the execution of a command before the FCCOB register can be written to with a new command.

CAUTION

A Flash word or phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash word or phrase is not allowed.

The Flash memory may be read as bytes, aligned words, or misaligned words. Read access time is one bus cycle for bytes and aligned words, and two bus cycles for misaligned words. For Flash memory, an erased bit reads 1 and a programmed bit reads 0.

It is not possible to read from a Flash block while any command is executing on that specific Flash block. It is possible to read from a Flash block while a command is executing on a different Flash block.

Both P-Flash and D-Flash memories are implemented with Error Correction Codes (ECC) that can resolve single bit faults and detect double bit faults. For P-Flash memory, the ECC implementation requires that programming be done on an aligned 8 byte basis (a Flash phrase). Since P-Flash memory is always read by phrase, only one single bit fault in the phrase containing the byte or word accessed will be corrected.

19.1.1 Glossary

Command Write Sequence — An MCU instruction sequence to execute built-in algorithms (including program and erase) on the Flash memory.

D-Flash Memory — The D-Flash memory constitutes the nonvolatile memory store for data.

D-Flash Sector — The D-Flash sector is the smallest portion of the D-Flash memory that can be erased. The D-Flash sector consists of four 64 byte rows for a total of 256 bytes.

NVM Command Mode — An NVM mode using the CPU to setup the FCCOB register to pass parameters required for Flash command execution.

Phrase — An aligned group of four 16-bit words within the P-Flash memory. Each phrase includes eight ECC bits for single bit fault correction and double bit fault detection within the phrase.

P-Flash Memory — The P-Flash memory constitutes the main nonvolatile memory store for applications.

P-Flash Sector — The P-Flash sector is the smallest portion of the P-Flash memory that can be erased. Each P-Flash sector contains 1024 bytes.

Program IFR — Nonvolatile information register located in the P-Flash block that contains the Device ID, Version ID, and the Program Once field. The Program IFR is visible in the global memory map by setting the PGMIFRON bit in the MMCCTL1 register.

19.1.2 Features

19.1.2.1 P-Flash Features

- 128 Kbytes of P-Flash memory composed of one 128 Kbyte Flash block divided into 128 sectors of 1024 bytes
- Single bit fault correction and double bit fault detection within a 64-bit phrase during read operations

Table 19-40. Program P-Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 101 at command launch
		Set if command not available in current mode (see Table 19-28)
		Set if an invalid global address [22:0] is supplied
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
	FPVIOL	Set if the global address [22:0] points to a protected area
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

19.4.2.6 Program Once Command

The Program Once command restricts programming to a reserved 64 byte field (8 phrases) in the nonvolatile information register located in P-Flash block 0. The Program Once reserved field can be read using the Read Once command as described in Section 19.4.2.4. The Program Once command must only be issued once since the nonvolatile information register in P-Flash block 0 cannot be erased. The Program Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 19-41. Program Once Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x07	Not Required
001	Program Once phrase index (0x0000 - 0x0007)	
010	Program Once word 0 value	
011	Program Once word 1 value	
100	Program Once word 2 value	
101	Program Once word 3 value	

Upon clearing CCIF to launch the Program Once command, the Memory Controller first verifies that the selected phrase is erased. If erased, then the selected phrase will be programmed and then verified with read back. The CCIF flag will remain clear, setting only after the Program Once operation has completed.

The reserved nonvolatile information register accessed by the Program Once command cannot be erased and any attempt to program one of these phrases a second time will not be allowed. Valid phrase index values for the Program Once command range from 0x0000 to 0x0007. During execution of the Program Once command, any attempt to read addresses within P-Flash block 0 will return invalid data.

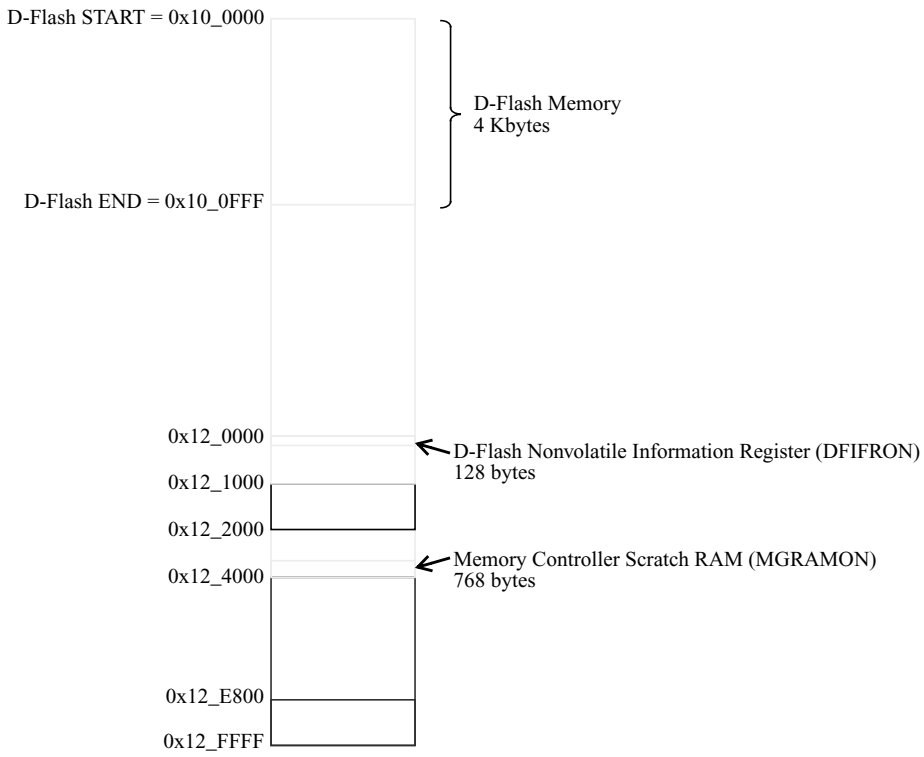


Figure 20-3. D-Flash and Memory Controller Resource Memory Map

20.3.2 Register Descriptions

The Flash module contains a set of 20 control and status registers located between Flash module base + 0x0000 and 0x0013. A summary of the Flash module registers is given in Figure 20-4 with detailed descriptions in the following subsections.

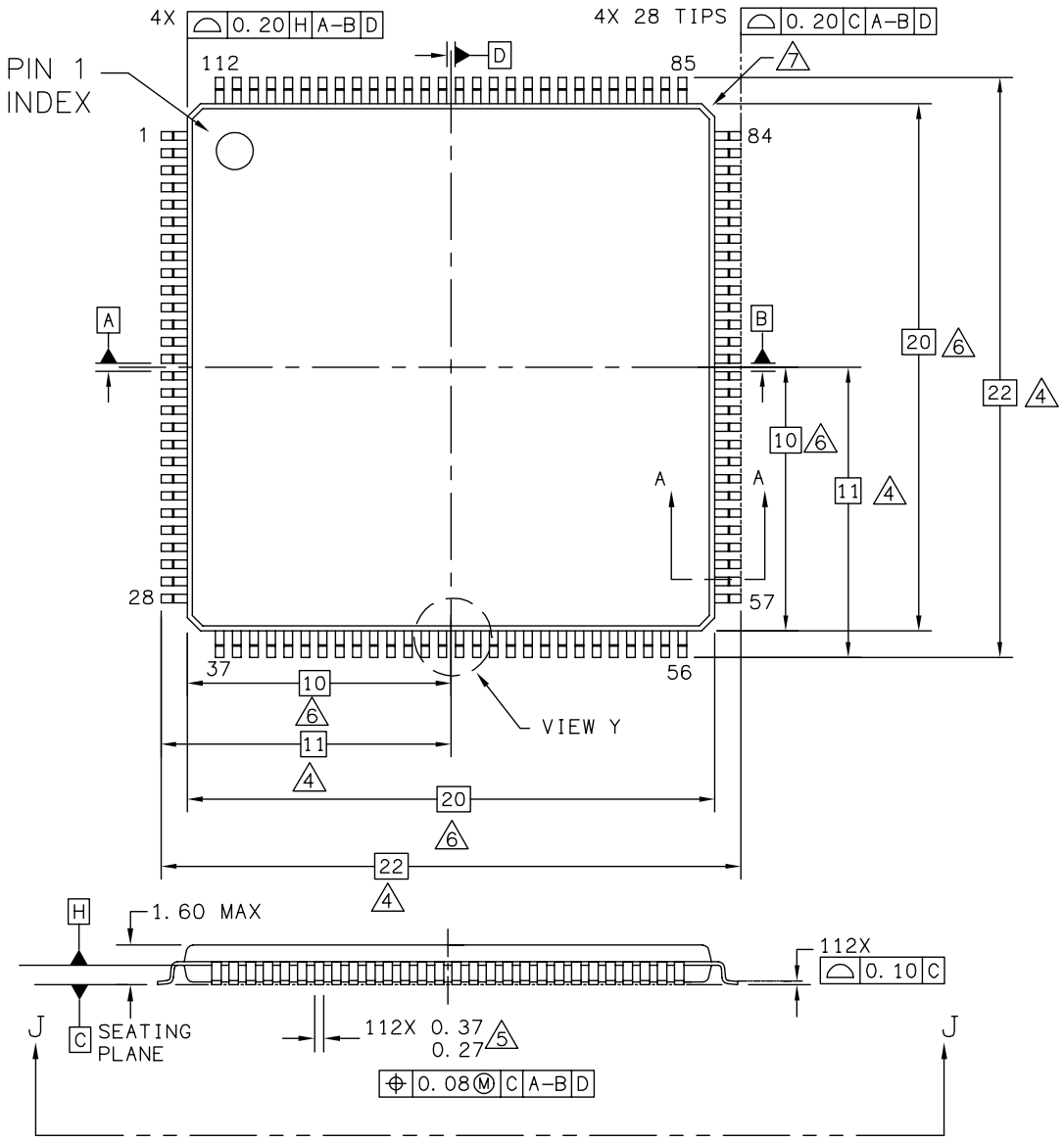
CAUTION

Writes to any Flash register must be avoided while a Flash command is active (CCIF=0) to prevent corruption of Flash register contents and Memory Controller behavior.

Address & Name		7	6	5	4	3	2	1	0
0x0000 FCLKDIV	R	FDIVLD	FDIV6	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
	W								
0x0001 FSEC	R	KEYEN1	KEYEN0	RNV5	RNV4	RNV3	RNV2	SEC1	SEC0
	W								

Figure 20-4. FTMR64K1 Register Summary

B.1 112-pin LQFP Mechanical Dimensions



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: 112LD LQFP 20 X 20 X 1.4 0.65 PITCH	DOCUMENT NO: 98ASS23330W	REV: F
	CASE NUMBER: 987-03	15 DEC 2006
	STANDARD: JEDEC MS-026 BFA	

Figure B-1. 112-pin LQFP (case no. 987) - page 1