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	59
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	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12xs128j1maar

Appendix A
Electrical Characteristics

Appendix B
Package Information

Appendix C
PCB Layout Guidelines

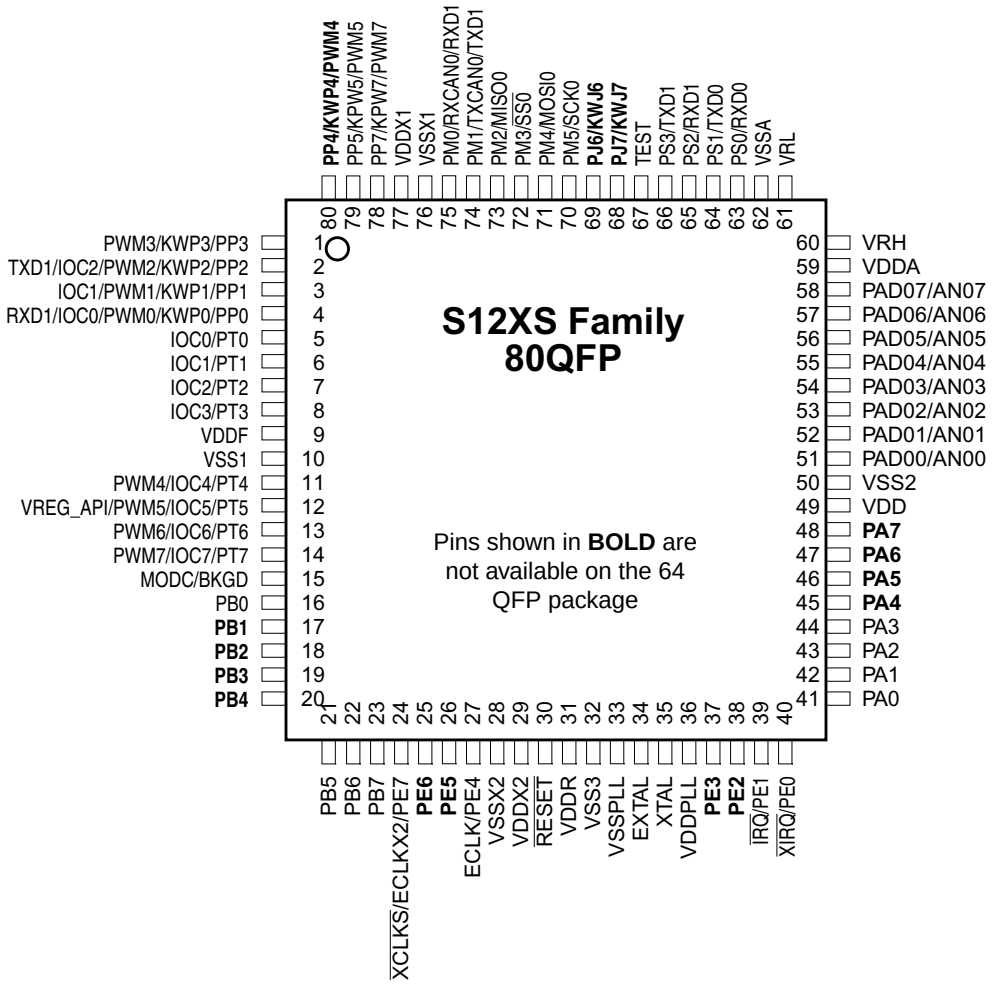

Figure 1-4. S12XS Family Pin Assignments 80-pin QFP Package

Table 1-6. Pin-Out Summary[1] (continued)

Package Terminal			Function					Power Supply	Internal Pull Resistor		Description
LQFP 112	QFP 80	LQFP 64	Pin	2nd Func.	3rd Func.	4th Func.	5th Func.		CTRL	Reset State	
55	39	31	PE1	$\overline{IRQ}$	—	—	—	V_{DDX}	PUCR	Up	Port E Input, maskable interrupt
56	40	32	PE0	$\overline{XIRQ}$	—	—	—	V_{DDX}	PUCR	Up	Port E Input, non-maskable interrupt
57	41	33	PA0	—	—	—	—	V_{DDX}	PUCR	Disabled	Port A I/O
58	42	34	PA1	—	—	—	—	V_{DDX}	PUCR	Disabled	Port A I/O
59	43	35	PA2	—	—	—	—	V_{DDX}	PUCR	Disabled	Port A I/O
60	44	36	PA3	—	—	—	—	V_{DDX}	PUCR	Disabled	Port A I/O
61	45	-	PA4	—	—	—	—	V_{DDX}	PUCR	Disabled	Port A I/O
62	46	-	PA5	—	—	—	—	V_{DDX}	PUCR	Disabled	Port A I/O
63	47	-	PA6	—	—	—	—	V_{DDX}	PUCR	Disabled	Port A I/O
64	48	-	PA7	—	—	—	—	V_{DDX}	PUCR	Disabled	Port A I/O
65	49	37	VDD	—	—	—	—	—	—	—	—
66	50	38	VSS2	—	—	—	—	—	—	—	—
67	51	39	PAD00	AN00	—	—	—	V_{DDA}	PER1AD	Disabled	Port AD I/O, analog input of ATD
68	-	-	PAD08	AN08	—	—	—	V_{DDA}	PER0AD	Disabled	Port AD I/O, analog input of ATD
69	52	40	PAD01	AN01	—	—	—	V_{DDA}	PER1AD	Disabled	Port AD I/O, analog input of ATD
70	-	-	PAD09	AN09	—	—	—	V_{DDA}	PER0AD	Disabled	Port AD I/O, analog input of ATD
71	53	41	PAD02	AN02	—	—	—	V_{DDA}	PER1AD	Disabled	Port AD I/O, analog input of ATD

1.9 BDM Clock Configuration

The BDM alternate clock source is the oscillator clock.

1.10 Oscillator Configuration

The $\overline{\text{XCLKS}}$ is an input signal which controls whether a crystal in combination with the internal loop controlled (low power) Pierce oscillator is used or whether full swing Pierce oscillator/external clock circuitry is used.

The $\overline{\text{XCLKS}}$ signal selects the oscillator configuration during reset low phase while a clock quality check is ongoing. This is the case for:

- Power on reset or low-voltage reset
- Clock monitor reset
- Any reset while in self-clock mode or full stop mode

2.3.3 Port A Data Register (PORTA)

Address 0x0000 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Reset	0	0	0	0	0	0	0	0

Figure 2-1. Port A Data Register (PORTA)

1 Read: Anytime, the data source depends on the data direction value
Write: Anytime

Table 2-4. PORTA Register Field Descriptions

Field	Description
7-0 PA	**Port A general purpose input/output data**—Data Register The associated pin can be used as general purpose I/O. In general purpose output mode the register bit value is driven to the pin. If the associated data direction bit is set to 1, a read returns the value of the port register bit, otherwise the buffered pin input state is read.

2.3.4 Port B Data Register (PORTB)

Address 0x0001 (PRR) Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Reset	0	0	0	0	0	0	0	0

Figure 2-2. Port B Data Register (PORTB)

1 Read: Anytime, the data source depends on the data direction value
Write: Anytime

Table 2-5. PORTB Register Field Descriptions

Field	Description
7-0 PB	**Port B general purpose input/output data**—Data Register The associated pin can be used as general purpose I/O. In general purpose output mode the register bit value is driven to the pin. If the associated data direction bit is set to 1, a read returns the value of the port register bit, otherwise the buffered pin input state is read.

Table 2-22. PTTRR Register Field Descriptions

Field	Description
7 PTTRR	**Port T peripheral routing**— This register controls the routing of PWM channel 7. 1 PWM7 routed to PT7 0 PWM7 routed to PP7
6 PTTRR	**Port T peripheral routing**— This register controls the routing of PWM channel 6. 1 PWM6 routed to PT6 0 PWM6 routed to PP6
5 PTTRR	**Port T peripheral routing**— This register controls the routing of PWM channel 5. 1 PWM5 routed to PT5 0 PWM5 routed to PP5
4 PTTRR	**Port T peripheral routing**— This register controls the routing of PWM channel 4. 1 PWM4 routed to PT4 0 PWM4 routed to PP4
2 PTTRR	**Port T peripheral routing**— This register controls the routing of TIM channel 2. 1 IOC2 routed to PP2 0 IOC2 routed to PT2
1 PTTRR	**Port T peripheral routing**— This register controls the routing of TIM channel 1. 1 IOC1 routed to PP1 0 IOC1 routed to PT1
0 PTTRR	**Port T peripheral routing**— This register controls the routing of TIM channel 0. 1 IOC0 routed to PP0 0 IOC0 routed to PT0

3.3.2.7 Data FLASH Page Index Register (EPAGE)

Address: 0x0017

	7	6	5	4	3	2	1	0
R W	EP7	EP6	EP5	EP4	EP3	EP2	EP1	EP0
Reset	1	1	1	1	1	1	1	0

Figure 3-15. Data FLASH Page Index Register (EPAGE)

Read: Anytime

Write: Anytime

These eight index bits are used to page 1KB blocks into the Data FLASH page window located in the local (CPU or BDM) memory map from address 0x0800 to address 0x0BFF (see Figure 3-16). This supports accessing up to 256KB of Data FLASH (in the Global map) within the 64KB Local map. The Data FLASH page index register is effectively used to construct paged Data FLASH addresses in the Local map format.

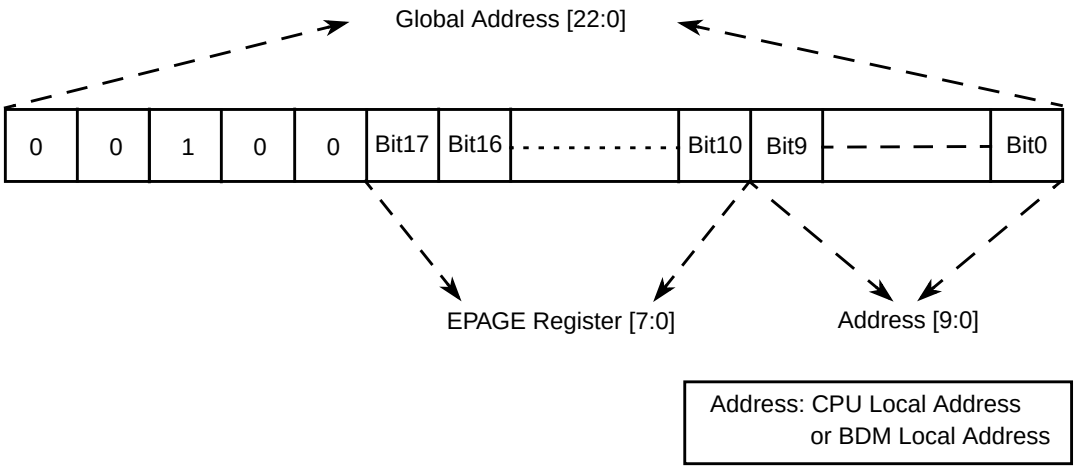

Figure 3-16. EPAGE Address Mapping

Table 3-9. EPAGE Field Descriptions

Field	Description
7–0 EP[7:0]	**Data FLASH Page Index Bits 7–0** — These page index bits are used to select which of the 256 Data FLASH array pages is to be accessed in the Data FLASH Page Window.

Table 6-22. State2 —Sequencer Next State Selection (continued)

SC[3:0]	Description
0101	Match3 triggers to Final State..... Other matches have no effect
0110	Match0 triggers to State1..... Match1 triggers to State3..... Other matches have no effect
0111	Match1 triggers to State3..... Match0 triggers Final State..... Other matches have no effect
1000	Match0 triggers to State1..... Match2 triggers to State3..... Other matches have no effect
1001	Match2 triggers to State3..... Match0 triggers Final State..... Other matches have no effect
1010	Match1 triggers to State1..... Match3 triggers to State3..... Other matches have no effect
1011	Match3 triggers to State3..... Match1 triggers Final State..... Other matches have no effect
1100	Match2 triggers to State1..... Match3 trigger to Final State
1101	Match2 has no affect, all other matches (M0,M1,M3) trigger to Final State
1110	Reserved. (No match triggers state sequencer transition)
1111	Reserved. (No match triggers state sequencer transition)

The trigger priorities described in Table 6-39 dictate that in the case of simultaneous matches, the match on the lower channel number (0,1,2,3) has priority. The SC[3:0] encoding ensures that a match leading to final state has priority over all other matches.

6.3.2.7.3 Debug State Control Register 3 (DBGSCR3)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	SC3	SC2	SC1	SC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 6-11. Debug State Control Register 3 (DBGSCR3)

Read: If COMRV[1:0] = 10

Write: If COMRV[1:0] = 10 and S12XDBG is not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 10. The state control register three selects the targeted next state whilst in State3. The matches refer to the match channels of the comparator match control logic as depicted in Figure 6-1 and described in Section 6.3.2.8.1. Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 6-23. DBGSCR3 Field Descriptions

Field	Description
3–0 SC[3:0]	These bits select the targeted next state whilst in State3, based upon the match event.

Table 6-24. State3 — Sequencer Next State Selection

SC[3:0]	Description
0000	Any match triggers to state1
0001	Any match triggers to state2

unimplemented bus, thus preventing proper operation.

The DBGC1_COMRV bits determine which comparator control, address, data and datamask registers are visible in the 8-byte window from 0x0028 to 0x002F as shown in Section Table 6-26.

Table 6-26. Comparator Address Register Visibility

COMRV	Visible Comparator
00	DBGACTL, DBGAAH ,DBGAAM, DBGAAL, DBGADH, DBGADL, DBGADHM, DBGADLM
01	DBGBCTL, DBGBAH, DBGBAM, DBGBAL
10	DBGCCTL, DBGCAH, DBGCAM, DBGCAL, DBGCDH, DBGCDL, DBGCDHM, DBGCDLM
11	DBGDCTL, DBGDAH, DBGDAM, DBGDAL

Table 6-27. DBGXCTL Field Descriptions

Field	Description
7 SZE (Comparators B and D)	**Size Comparator Enable Bit** — The SZE bit controls whether access size comparison is enabled for the associated comparator. This bit is ignored if the TAG bit in the same register is set. 0 Word/Byte access size is not used in comparison 1 Word/Byte access size is used in comparison
6 NDB (Comparators A and C	**Not Data Bus** — The NDB bit controls whether the match occurs when the data bus matches the comparator register value or when the data bus differs from the register value. Furthermore data bus bits can be individually masked using the comparator data mask registers. This bit is only available for comparators A and C. This bit is ignored if the TAG bit in the same register is set. This bit position has an SZ functionality for comparators B and D. 0 Match on data bus equivalence to comparator register contents 1 Match on data bus difference to comparator register contents
6 SZ (Comparators B and D)	**Size Comparator Value Bit** — The SZ bit selects either word or byte access size in comparison for the associated comparator. This bit is ignored if the SZE bit is cleared or if the TAG bit in the same register is set. This bit position has NDB functionality for comparators A and C 0 Word access size will be compared 1 Byte access size will be compared
5 TAG	**Tag Select** — This bit controls whether the comparator match will cause a trigger or tag the opcode at the matched address. Tagged opcodes trigger only if they reach the execution stage of the instruction queue. 0 Trigger immediately on match 1 On match, tag the opcode. If the opcode is about to be executed a trigger is generated
4 BRK	**Break** — This bit controls whether a channel match terminates a debug session immediately, independent of state sequencer state. To generate an immediate breakpoint the module breakpoints must be enabled using DBGBRK. 0 The debug session termination is dependent upon the state sequencer and trigger conditions. 1 A match on this channel terminates the debug session immediately; breakpoints if active are generated, tracing, if active, is terminated and the module disarmed.
3 RW	**Read/Write Comparator Value Bit** — The RW bit controls whether read or write is used in compare for the associated comparator . The RW bit is not used if RWE = 0. 0 Write cycle will be matched 1 Read cycle will be matched
2 RWE	**Read/Write Enable Bit** — The RWE bit controls whether read or write comparison is enabled for the associated comparator. This bit is not used for tagged operations. 0 Read/Write is not used in comparison 1 Read/Write is used in comparison

Chapter 11
Freescale's Scalable Controller Area Network (S12MSCANV3)

Table 11-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V03.11	31 Mar 2009		• Orthographic corrections
V03.12	09 Aug 2010	Table 11-37	• Added 'Bosch CAN 2.0A/B' to bit time settings table
V03.13	03 Mar 2011	Figure 11-4 Table 11-3	• Corrected CANE write restrictions • Removed footnote from RXFRM bit

11.1 Introduction

Freescale's scalable controller area network (S12MSCANV3) definition is based on the MSCAN12 definition, which is the specific implementation of the MSCAN concept targeted for the M68HC12 microcontroller family.

The module is a communication controller implementing the CAN 2.0A/B protocol as defined in the Bosch specification dated September 1991. For users to fully understand the MSCAN specification, it is recommended that the Bosch specification be read first to familiarize the reader with the terms and concepts contained within this document.

Though not exclusively intended for automotive applications, CAN protocol is designed to meet the specific requirements of a vehicle serial data bus: real-time processing, reliable operation in the EMI environment of a vehicle, cost-effectiveness, and required bandwidth.

MSCAN uses an advanced buffer arrangement resulting in predictable real-time behavior and simplified application software.

11.4.3.1 Protocol Violation Protection

The MSCAN protects the user from accidentally violating the CAN protocol through programming errors. The protection logic implements the following features:

- The receive and transmit error counters cannot be written or otherwise manipulated.
- All registers which control the configuration of the MSCAN cannot be modified while the MSCAN is on-line. The MSCAN has to be in Initialization Mode. The corresponding INITRQ/INITAK handshake bits in the CANCTL0/CANCTL1 registers (see Section 11.3.2.1, "MSCAN Control Register 0 (CANCTL0)") serve as a lock to protect the following registers:
  - MSCAN control 1 register (CANCTL1)
  - MSCAN bus timing registers 0 and 1 (CANBTR0, CANBTR1)
  - MSCAN identifier acceptance control register (CANIDAC)
  - MSCAN identifier acceptance registers (CANIDAR0–CANIDAR7)
  - MSCAN identifier mask registers (CANIDMR0–CANIDMR7)
- The TXCAN is immediately forced to a recessive state when the MSCAN goes into the power down mode or initialization mode (see Section 11.4.5.6, "MSCAN Power Down Mode," and Section 11.4.4.5, "MSCAN Initialization Mode").
- The MSCAN enable bit (CANE) is writable only once in normal system operation modes, which provides further protection against inadvertently disabling the MSCAN.

11.4.3.2 Clock System

Figure 11-43 shows the structure of the MSCAN clock generation circuitry.

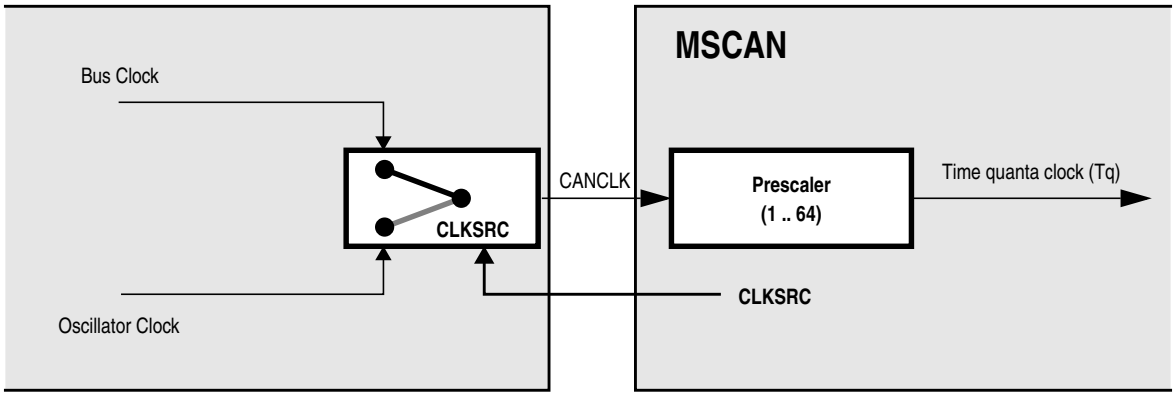

Figure 11-43. MSCAN Clocking Scheme

The clock source bit (CLKSRC) in the CANCTL1 register (11.3.2.2/11-303) defines whether the internal CANCLK is connected to the output of a crystal oscillator (oscillator clock) or to the bus clock.

The clock source has to be chosen such that the tight oscillator tolerance requirements (up to 0.4%) of the CAN protocol are met. Additionally, for high CAN bus rates (1 Mbps), a 45% to 55% duty cycle of the clock is required.

If the bus clock is generated from a PLL, it is recommended to select the oscillator clock rather than the bus clock due to jitter considerations, especially at the faster CAN bus rates.

12.3.0.1 PIT Control and Force Load Micro Timer Register (PITCFLMT)

Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	PITE	PITSWAI	PITFRZ	0	0	0	0	0
W							PFLMT1	PFLMT0
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 12-3. PIT Control and Force Load Micro Timer Register (PITCFLMT)

Read: Anytime

Write: Anytime; writes to the reserved bits have no effect

Table 12-2. PITCFLMT Field Descriptions

Field	Description
7 PITE	**PIT Module Enable Bit** — This bit enables the PIT module. If PITE is cleared, the PIT module is disabled and flag bits in the PITTF register are cleared. When PITE is set, individually enabled timers (PCE set) start down-counting with the corresponding load register values. 0 PIT disabled (lower power consumption). 1 PIT is enabled.
6 PITSWAI	**PIT Stop in Wait Mode Bit** — This bit is used for power conservation while in wait mode. 0 PIT operates normally in wait mode 1 PIT clock generation stops and freezes the PIT module when in wait mode
5 PITFRZ	**PIT Counter Freeze while in Freeze Mode Bit** — When during debugging a breakpoint (freeze mode) is encountered it is useful in many cases to freeze the PIT counters to avoid e.g. interrupt generation. The PITFRZ bit controls the PIT operation while in freeze mode. 0 PIT operates normally in freeze mode 1 PIT counters are stalled when in freeze mode
1:0 PFLMT[1:0]	**PIT Force Load Bits for Micro Timer 1:0** — These bits have only an effect if the corresponding micro timer is active and if the PIT module is enabled (PITE set). Writing a one into a PFLMT bit loads the corresponding 8-bit micro timer load register into the 8-bit micro timer down-counter. Writing a zero has no effect. Reading these bits will always return zero. **Note:** A micro timer force load affects all timer channels that use the corresponding micro time base.

drive the MOSI and SCK lines. In this case, the SPI immediately switches to slave mode, by clearing the MSTR bit and also disables the slave output buffer MISO (or SISO in bidirectional mode). So the result is that all outputs are disabled and SCK, MOSI, and MISO are inputs. If a transmission is in progress when the mode fault occurs, the transmission is aborted and the SPI is forced into idle state.

This mode fault error also sets the mode fault (MODF) flag in the SPI status register (SPISR). If the SPI interrupt enable bit (SPIE) is set when the MODF flag becomes set, then an SPI interrupt sequence is also requested.

When a write to the SPI data register in the master occurs, there is a half SCK-cycle delay. After the delay, SCK is started within the master. The rest of the transfer operation differs slightly, depending on the clock format specified by the SPI clock phase bit, CPHA, in SPI control register 1 (see Section 15.4.3, “Transmission Formats”).

NOTE

A change of the bits CPOL, CPHA, SSOE, LSBFE, XFRW, MODFEN, SPC0, or BIDIROE with SPC0 set, SPPR2-SPPR0 and SPR2-SPR0 in master mode will abort a transmission in progress and force the SPI into idle state. The remote slave cannot detect this, therefore the master must ensure that the remote slave is returned to idle state.

15.4.2 Slave Mode

The SPI operates in slave mode when the MSTR bit in SPI control register 1 is clear.

- Serial clock

  In slave mode, SCK is the SPI clock input from the master.

- MISO, MOSI pin

  In slave mode, the function of the serial data output pin (MISO) and serial data input pin (MOSI) is determined by the SPC0 bit and BIDIROE bit in SPI control register 2.

- $\overline{SS}$ pin

  The $\overline{SS}$ pin is the slave select input. Before a data transmission occurs, the $\overline{SS}$ pin of the slave SPI must be low. $\overline{SS}$ must remain low until the transmission is complete. If $\overline{SS}$ goes high, the SPI is forced into idle state.

  The $\overline{SS}$ input also controls the serial data output pin, if $\overline{SS}$ is high (not selected), the serial data output pin is high impedance, and, if $\overline{SS}$ is low, the first bit in the SPI data register is driven out of the serial data output pin. Also, if the slave is not selected ($\overline{SS}$ is high), then the SCK input is ignored and no internal shifting of the SPI shift register occurs.

  Although the SPI is capable of duplex operation, some SPI peripherals are capable of only receiving SPI data in a slave mode. For these simpler devices, there is no serial data out pin.

NOTE

When peripherals with duplex capability are used, take care not to simultaneously enable two receivers whose serial outputs drive the same system slave’s serial data output line.

15.4.7.5.2 SPIF

SPIF occurs when new data has been received and copied to the SPI data register. After SPIF is set, it does not clear until it is serviced. SPIF has an automatic clearing process, which is described in Section 15.3.2.4, “SPI Status Register (SPISR)”.

15.4.7.5.3 SPTEF

SPTEF occurs when the SPI data register is ready to accept new data. After SPTEF is set, it does not clear until it is serviced. SPTEF has an automatic clearing process, which is described in Section 15.3.2.4, “SPI Status Register (SPISR)”.

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.

Table 19-29. P-Flash Commands

FCMD	Command	Function on P-Flash Memory
0x01	Erase Verify All Blocks	Verify that all P-Flash (and D-Flash) blocks are erased.
0x02	Erase Verify Block	Verify that a P-Flash block is erased.
0x03	Erase Verify P-Flash Section	Verify that a given number of words starting at the address provided are erased.
0x04	Read Once	Read a dedicated 64 byte field in the nonvolatile information register in P-Flash block 0 that was previously programmed using the Program Once command.
0x06	Program P-Flash	Program a phrase in a P-Flash block.
0x07	Program Once	Program a dedicated 64 byte field in the nonvolatile information register in P-Flash block 0 that is allowed to be programmed only once.
0x08	Erase All Blocks	Erase all P-Flash (and D-Flash) blocks. An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command.
0x09	Erase Flash Block	Erase a P-Flash (or D-Flash) block. An erase of the full P-Flash block is only possible when FPLDIS, FPHDIS and FPOPEN bits in the FPROT register are set prior to launching the command.
0x0A	Erase P-Flash Sector	Erase all bytes in a P-Flash sector.
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all P-Flash (and D-Flash) blocks and verifying that all P-Flash (and D-Flash) blocks are erased.
0x0C	Verify Backdoor Access Key	Supports a method of releasing MCU security by verifying a set of security keys.
0x0D	Set User Margin Level	Specifies a user margin read level for all P-Flash blocks.
0x0E	Set Field Margin Level	Specifies a field margin read level for all P-Flash blocks (special modes only).

19.4.1.5 D-Flash Commands

Table 19-30 summarizes the valid D-Flash commands along with the effects of the commands on the D-Flash block.

Table 19-30. D-Flash Commands

FCMD	Command	Function on D-Flash Memory
0x01	Erase Verify All Blocks	Verify that all D-Flash (and P-Flash) blocks are erased.
0x02	Erase Verify Block	Verify that the D-Flash block is erased.
0x08	Erase All Blocks	Erase all D-Flash (and P-Flash) blocks. An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command.
0x09	Erase Flash Block	Erase a D-Flash (or P-Flash) block. An erase of the full D-Flash block is only possible when DPOPEN bit in the DFPROT register is set prior to launching the command.

Table 19-48. Erase P-Flash Sector Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if command not available in current mode (see Table 19-28)
		Set if an invalid global address [22:16] is supplied
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
	FPVIOL	Set if the selected P-Flash sector is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

19.4.2.10 Unsecure Flash Command

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

Table 19-49. Unsecure Flash Command FCCOB Requirements

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

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

Table 19-50. Unsecure Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if command not available in current mode (see Table 19-28)
	FPVIOL	Set if any area of the P-Flash or D-Flash memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

19.4.2.11 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command will only execute if it is enabled by the KEYEN bits in the FSEC register (see Table 19-9). The Verify Backdoor Access Key command releases security if user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see

20.5.1 Unsecuring the MCU using Backdoor Key Access

The MCU may be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor keys (four 16-bit words programmed at addresses 0x7F_FF00–0x7F_FF07). If the KEYEN[1:0] bits are in the enabled state (see Section 20.3.2.2), the Verify Backdoor Access Key command (see Section 20.4.2.11) allows the user to present four prospective keys for comparison to the keys stored in the Flash memory via the Memory Controller. If the keys presented in the Verify Backdoor Access Key command match the backdoor keys stored in the Flash memory, the SEC bits in the FSEC register (see Table 20-10) will be changed to unsecure the MCU. Key values of 0x0000 and 0xFFFF are not permitted as backdoor keys. While the Verify Backdoor Access Key command is active, P-Flash block 0 will not be available for read access and will return invalid data.

The user code stored in the P-Flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 20.3.2.2), the MCU can be unsecured by the backdoor key access sequence described below:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Section 20.4.2.11
2. If the Verify Backdoor Access Key command is successful, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 10

The Verify Backdoor Access Key command is monitored by the Memory Controller and an illegal key will prohibit future use of the Verify Backdoor Access Key command. A reset of the MCU is the only method to re-enable the Verify Backdoor Access Key command.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the sector containing the Flash security byte can be erased and the Flash security byte can be reprogrammed to the unsecure state, if desired.

In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0x7F_FF00–0x7F_FF07 in the Flash configuration field.

The security as defined in the Flash security byte (0x7F_FF0F) is not changed by using the Verify Backdoor Access Key command sequence. The backdoor keys stored in addresses 0x7F_FF00–0x7F_FF07 are unaffected by the Verify Backdoor Access Key command sequence. After the next reset of the MCU, the security state of the Flash module is determined by the Flash security byte (0x7F_FF0F). The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the Flash protection register, FPROT.

20.5.2 Unsecuring the MCU in Special Single Chip Mode using BDM

The MCU can be unsecured in special single chip mode by erasing the P-Flash and D-Flash memory by one of the following methods:

- Reset the MCU into special single chip mode, delay while the erase test is performed by the BDM, send BDM commands to disable protection in the P-Flash and D-Flash memory, and execute the Erase All Blocks command write sequence to erase the P-Flash and D-Flash memory.

A.2.3.1 ATD Accuracy Definitions

For the following definitions see also Figure A-1.
Differential non-linearity (DNL) is defined as the difference between two adjacent switching steps.

$$DNL(i) = \frac{V_i - V_{i-1}}{1LSB} - 1$$

The integral non-linearity (INL) is defined as the sum of all DNLs:

$$INL(n) = \sum_{i=1}^{n} DNL(i) = \frac{V_n - V_0}{1LSB} - n$$

Detailed MSCAN Foreground Receive and Transmit Buffer Layout

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0xXXX0	Extended ID	R	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
	Standard ID	R	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
	CANxRIDR0	W								
0xXXX1	Extended ID	R	ID20	ID19	ID18	SRR=1	IDE=1	ID17	ID16	ID15
	Standard ID	R	ID2	ID1	ID0	RTR	IDE=0			
	CANxRIDR1	W								
0xXXX2	Extended ID	R	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
	Standard ID	R								
	CANxRIDR2	W								
0xXXX3	Extended ID	R	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
	Standard ID	R								
	CANxRIDR3	W								
0xXXX4-0xXXXB	CANxRDSR0-CANxRDSR7	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
		W								
0xXXXC	CANRxDLR	R					DLC3	DLC2	DLC1	DLC0
		W								
0xXXXD	Reserved	R								
		W								
0xXXXE	CANxRTSRH	R	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8
		W								
0xXXXF	CANxRTSRL	R	TSR7	TSR6	TSR5	TSR4	TSR3	TSR2	TSR1	TSR0
		W								
0xXX10	Extended ID CANxTIDR0	R	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
		W								
	Standard ID	R	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
		W								
0xXX0x XX10	Extended ID CANxTIDR1	R	ID20	ID19	ID18	SRR=1	IDE=1	ID17	ID16	ID15
		W								
	Standard ID	R	ID2	ID1	ID0	RTR	IDE=0			
		W								
0xXX12	Extended ID CANxTIDR2	R	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
		W								
	Standard ID	R								
		W								
0xXX13	Extended ID CANxTIDR3	R	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
		W								
	Standard ID	R								
		W								
0xXX14-0xXX1B	CANxTDSR0–CANxTDSR7	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
		W								
0xXX1C	CANxTDLR	R					DLC3	DLC2	DLC1	DLC0
		W								
0xXX1D	CANxTTBPR	R	PRIO7	PRIO6	PRIO5	PRIO4	PRIO3	PRIO2	PRIO1	PRIO0
		W								
0xXX1E	CANxTTSRH	R	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8
		W								