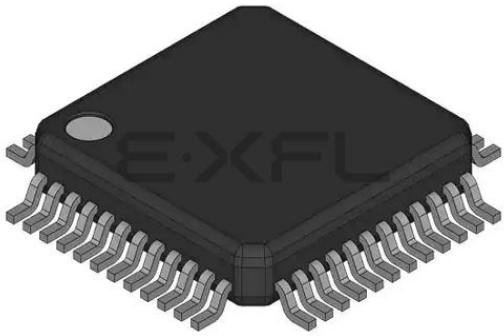

Welcome to E-XFL.COM

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	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	31
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12c32vfae25

- Operating frequency:
  - 32MHz equivalent to 16MHz bus speed for single chip
  - 32MHz equivalent to 16MHz bus speed in expanded bus modes
  - Option of 9S12C Family: 50MHz equivalent to 25MHz bus speed
  - All 9S12GC Family members allow a 50MHz operating frequency.
- Internal 2.5V regulator:
  - Supports an input voltage range from 2.97V to 5.5V
  - Low power mode capability
  - Includes low voltage reset (LVR) circuitry
  - Includes low voltage interrupt (LVI) circuitry
- 48-pin LQFP, 52-pin LQFP, or 80-pin QFP package:
  - Up to 58 I/O lines with 5V input and drive capability (80-pin package)
  - Up to 2 dedicated 5V input only lines (IRQ, XIRQ)
  - 5V 8 A/D converter inputs and 5V I/O
- Development support:
  - Single-wire background debug™ mode (BDM)
  - On-chip hardware breakpoints
  - Enhanced DBG12 debug features

1.1.2 Modes of Operation

User modes (expanded modes are only available in the 80-pin package version).

- Normal and emulation operating modes:
  - Normal single-chip mode
  - Normal expanded wide mode
  - Normal expanded narrow mode
  - Emulation expanded wide mode
  - Emulation expanded narrow mode
- Special operating modes:
  - Special single-chip mode with active background debug mode
  - Special test mode **(Freescale use only)**
  - Special peripheral mode **(Freescale use only)**
- Low power modes:
  - Stop mode
  - Pseudo stop mode
  - Wait mode

Table 1-9. Interrupt Vector Locations (continued)

Vector Address	Interrupt Source	CCR Mask	Local Enable	HPRIO Value to Elevate
0xFFDE, 0xFFDF	Standard timer overflow	I bit	TMSK2 (TOI)	0x00DE
0xFFDC, 0xFFDD	Pulse accumulator A overflow	I bit	PACTL (PAOVI)	0x00DC
0xFFDA, 0xFFDB	Pulse accumulator input edge	I bit	PACTL (PAI)	0x00DA
0xFFD8, 0xFFD9	SPI	I bit	SPICR1 (SPIE, SPTIE)	0x00D8
0xFFD6, 0xFFD7	SCI	I bit	SCICR2 (TIE, TCIE, RIE, ILIE)	0x00D6
0xFFD4, 0xFFD5	Reserved			
0xFFD2, 0xFFD3	ATD	I bit	ATDCTL2 (ASCIE)	0x00D2
0xFFD0, 0xFFD1	Reserved			
0xFFCE, 0xFFCF	Port J	I bit	PIEP (PIEP7-6)	0x00CE
0xFFCC, 0xFFCD	Reserved			
0xFFCA, 0xFFCB	Reserved			
0xFFC8, 0xFFC9	Reserved			
0xFFC6, 0xFFC7	CRG PLL lock	I bit	PLLCR (LOCKIE)	0x00C6
0xFFC4, 0xFFC5	CRG self clock mode	I bit	PLLCR (SCMIE)	0x00C4
0xFFBA to 0xFFC3	Reserved			
0xFFB8, 0xFFB9	FLASH	I bit	FCNFG (CCIE, CBEIE)	0x00B8
0xFFB6, 0xFFB7	CAN wake-up(1)	I bit	CANRIER (WUPIE)	0x00B6
0xFFB4, 0xFFB5	CAN errors[1]	I bit	CANRIER (CSCIE, OVRIE)	0x00B4
0xFFB2, 0xFFB3	CAN receive[1]	I bit	CANRIER (RXFIE)	0x00B2
0xFFB0, 0xFFB1	CAN transmit[1]	I bit	CANTIER (TXEIE[2:0])	0x00B0
0xFF90 to 0xFFAF	Reserved			
0xFF8E, 0xFF8F	Port P	I bit	PIEP (PIEP7-0)	0x008E
0xFF8C, 0xFF8D	Reserved			
0xFF8C, 0xFF8D	PWM Emergency Shutdown	I bit	PWMSDN(PWMIE)	0x008C
0xFF8A, 0xFF8B	VREG LVI	I bit	CTRL0 (LVIE)	0x008A
0xFF80 to 0xFF89	Reserved			

1. Not available on MC9S12GC Family members

2.3.2.1.2 Port T Input Register (PTIT)

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	PTIT7	PTIT6	PTIT5	PTIT4	PTIT3	PTIT2	PTIT1	PTIT0
W								
Reset	—	—	—	—	—	—	—	—

= Unimplemented or Reserved

Figure 2-4. Port T Input Register (PTIT)

Read: Anytime.

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

Table 2-4. PTIT Field Descriptions

Field	Description
7–0 PTIT[7:0]	**Port T Input Register** — This register always reads back the status of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

2.3.2.1.3 Port T Data Direction Register (DDRT)

Module Base + 0x0002

	7	6	5	4	3	2	1	0
R W	DDRT7	DDRT6	DDRT5	DDRT4	DDRT3	DDRT2	DDRT1	DDRT0
Reset	0	0	0	0	0	0	0	0

Figure 2-5. Port T Data Direction Register (DDRT)

Read: Anytime.

Write: Anytime.

Table 2-5. DDRT Field Descriptions

Field	Description
7–0 DDRT[7:0]	**Data Direction Port T** — This register configures each port T pin as either input or output. The standard TIM / PWM modules forces the I/O state to be an output for each standard TIM / PWM module port associated with an enabled output compare. In these cases the data direction bits will not change. The DDRT bits revert to controlling the I/O direction of a pin when the associated timer output compare is disabled. The timer input capture always monitors the state of the pin. 0 Associated pin is configured as input. 1 Associated pin is configured as output. **Note:** Due to internal synchronization circuits, it can take up to 2 bus cycles until the correct value is read on PTT or PTIT registers, when changing the DDRT register.

2.3.2.3.4 Port M Reduced Drive Register (RDRM)

Module Base + 0x0013

	7	6	5	4	3	2	1	0
R	0	0	RDRM5	RDRM4	RDRM3	RDRM2	RDRM1	RDRM0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 2-20. Port M Reduced Drive Register (RDRM)

Read: Anytime.

Write: Anytime.

Table 2-18. RDRM Field Descriptions

Field	Description
5–0 RDRM[5:0]	**Reduced Drive Port M** — This register configures the drive strength of each port M output pin as either full or reduced. If the port is used as input this bit is ignored. 0 Full drive strength at output. 1 Associated pin drives at about 1/3 of the full drive strength.

2.3.2.3.5 Port M Pull Device Enable Register (PERM)

Module Base + 0x0014

	7	6	5	4	3	2	1	0
R	0	0	PERM5	PERM4	PERM3	PERM2	PERM1	PERM0
W								
Reset	0	0	1	1	1	1	1	1

= Unimplemented or Reserved

Figure 2-21. Port M Pull Device Enable Register (PERM)

Read: Anytime.

Write: Anytime.

Table 2-19. PERM Field Descriptions

Field	Description
5–0 PERM[5:0]	**Pull Device Enable Port M** — This register configures whether a pull-up or a pull-down device is activated, if the port is used as input or as output in wired-or (open drain) mode. This bit has no effect if the port is used as push-pull output. Out of reset a pull-up device is enabled. 0 Pull-up or pull-down device is disabled. 1 Either a pull-up or pull-down device is enabled.

3.3.2 Register Descriptions

Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0010 INITRM	R	RAM15	RAM14	RAM13	RAM12	RAM11	0	0	RAMHAL
	W								
0x0011 INITRG	R	0	REG14	REG13	REG12	REG11	0	0	0
	W								
0x0012 INITEE	R	EE15	EE14	EE13	EE12	EE11	0	0	EEON
	W								
0x0013 MISC	R	0	0	0	0	EXSTR1	EXSTR0	ROMHM	ROMON
	W								
0x0014 MTSTO	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0017 MTST1	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x001C MEMSIZ0	R	REG_SW0	0	EEP_SW1	EEP_SW0	0	RAM_SW2	RAM_SW1	RAM_SW0
	W								
0x001D MEMSIZ1	R	ROM_SW1	ROM_SW0	0	0	0	0	PAG_SW1	PAG_SW0
	W								
0x0030 PPAGE	R	0	0	PIX5	PIX4	PIX3	PIX2	PIX1	PIX0
	W								
0x0031 Reserved	R	0	0	0	0	0	0	0	0
	W								

= Unimplemented

Figure 3-2. MMC Register Summary

firmware. The standard BDM firmware watches for serial commands and executes them as they are received.

The firmware commands are shown in Table 6-6.

Table 6-6. Firmware Commands

Command[1]	Opcode (hex)	Data	Description
READ_NEXT	62	16-bit data out	Increment X by 2 (X = X + 2), then read word X points to.
READ_PC	63	16-bit data out	Read program counter.
READ_D	64	16-bit data out	Read D accumulator.
READ_X	65	16-bit data out	Read X index register.
READ_Y	66	16-bit data out	Read Y index register.
READ_SP	67	16-bit data out	Read stack pointer.
WRITE_NEXT	42	16-bit data in	Increment X by 2 (X = X + 2), then write word to location pointed to by X.
WRITE_PC	43	16-bit data in	Write program counter.
WRITE_D	44	16-bit data in	Write D accumulator.
WRITE_X	45	16-bit data in	Write X index register.
WRITE_Y	46	16-bit data in	Write Y index register.
WRITE_SP	47	16-bit data in	Write stack pointer.
GO	08	None	Go to user program. If enabled, ACK will occur when leaving active background mode.
GO_UNTIL[2]	0C	None	Go to user program. If enabled, ACK will occur upon returning to active background mode.
TRACE1	10	None	Execute one user instruction then return to active BDM. If enabled, ACK will occur upon returning to active background mode.
TAGGO	18	None	Enable tagging and go to user program. There is no ACK pulse related to this command.

1. If enabled, ACK will occur when data is ready for transmission for all BDM READ commands and will occur after the write is complete for all BDM WRITE commands.
2. Both WAIT (with clocks to the S12 CPU core disabled) and STOP disable the ACK function. The GO_UNTIL command will not get an Acknowledge if one of these two CPU instructions occurs before the “UNTIL” instruction. This can be a problem for any instruction that uses ACK, but GO_UNTIL is a lot more difficult for the development tool to time-out.

6.4.5 BDM Command Structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.

NOTE

8-bit reads return 16-bits of data, of which, only one byte will contain valid data. If reading an even address, the valid data will appear in the MSB. If reading an odd address, the valid data will appear in the LSB.

Chapter 7
Debug Module (DBGV1) Block Description

7.1 Introduction

This section describes the functionality of the debug (DBG) sub-block of the HCS12 core platform.

The DBG module is designed to be fully compatible with the existing BKP_HCS12_A module (BKP mode) and furthermore provides an on-chip trace buffer with flexible triggering capability (DBG mode). The DBG module provides for non-intrusive debug of application software. The DBG module is optimized for the HCS12 16-bit architecture.

7.1.1 Features

The DBG module in BKP mode includes these distinctive features:

- Full or dual breakpoint mode
  - Compare on address and data (full)
  - Compare on either of two addresses (dual)
- BDM or SWI breakpoint
  - Enter BDM on breakpoint (BDM)
  - Execute SWI on breakpoint (SWI)
- Tagged or forced breakpoint
  - Break just before a specific instruction will begin execution (TAG)
  - Break on the first instruction boundary after a match occurs (Force)
- Single, range, or page address compares
  - Compare on address (single)
  - Compare on address 256 byte (range)
  - Compare on any 16K page (page)
- At forced breakpoints compare address on read or write
- High and/or low byte data compares
- Comparator C can provide an additional tag or force breakpoint (enhancement for BKP mode)

Table 7-14. DBGC2 Field Descriptions (continued)

Field	Description
4 TAGAB	**Comparator A/B Tag Select** — This bit controls whether the breakpoint will cause a break on the next instruction boundary (force) or on a match that will be an executable opcode (tagged). Non-executed opcodes cannot cause a tagged breakpoint. 0 On match, break at the next instruction boundary (force) 1 On match, break if/when the instruction is about to be executed (tagged)
3 BKCEN	**Breakpoint Comparator C Enable Bit** — This bit enables the breakpoint capability using comparator C. 0 Comparator C disabled for breakpoint 1 Comparator C enabled for breakpoint **Note:** This bit will be cleared automatically when the DBG module is armed in loop1 mode.
2 TAGC	**Comparator C Tag Select** — This bit controls whether the breakpoint will cause a break on the next instruction boundary (force) or on a match that will be an executable opcode (tagged). Non-executed opcodes cannot cause a tagged breakpoint. 0 On match, break at the next instruction boundary (force) 1 On match, break if/when the instruction is about to be executed (tagged)
1 RWCEN	**Read/Write Comparator C Enable Bit** — The RWCEN bit controls whether read or write comparison is enabled for comparator C. RWCEN is not useful for tagged breakpoints. 0 Read/Write is not used in comparison 1 Read/Write is used in comparison
0 RWC	**Read/Write Comparator C Value Bit** — The RWC bit controls whether read or write is used in compare for comparator C. The RWC bit is not used if RWCEN = 0. 0 Write cycle will be matched 1 Read cycle will be matched

7.3.2.8 Debug Control Register 3 (DBGC3)

Module Base + 0x0029
Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R W	BKAMBH(1)	BKAMBL1	BKBMBH(2)	BKBMBL2	RWAEN	RWA	RWBEN	RWB
Reset	0	0	0	0	0	0	0	0

1. In DBG mode, BKAMBH:BKAMBL has no meaning and are forced to 0's.
2. In DBG mode, BKBMBH:BKBMBL are used in full mode to qualify data.

Figure 7-14. Debug Control Register 3 (DBGC3)

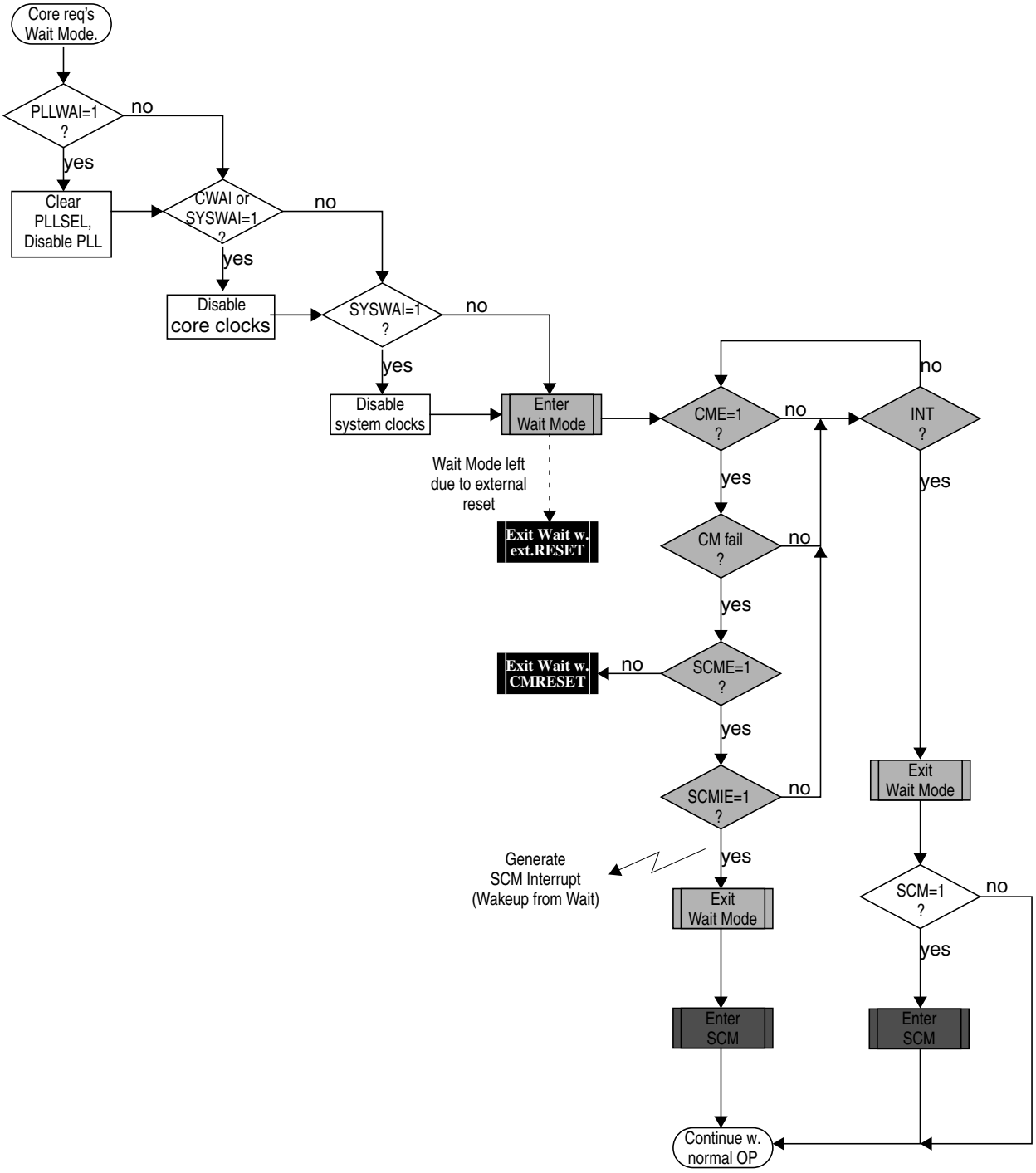

Figure 9-23. Wait Mode Entry/Exit Sequence

10.1.4 Modes of Operation

The following modes of operation are specific to the MSCAN. See Section 10.4, "Functional Description," for details.

- Listen-Only Mode
- MSCAN Sleep Mode
- MSCAN Initialization Mode
- MSCAN Power Down Mode

10.2 External Signal Description

The MSCAN uses two external pins:

10.2.1 RXCAN — CAN Receiver Input Pin

RXCAN is the MSCAN receiver input pin.

10.2.2 TXCAN — CAN Transmitter Output Pin

TXCAN is the MSCAN transmitter output pin. The TXCAN output pin represents the logic level on the CAN bus:

0 = Dominant state
1 = Recessive state

10.2.3 CAN System

A typical CAN system with MSCAN is shown in Figure 10-2. Each CAN station is connected physically to the CAN bus lines through a transceiver device. The transceiver is capable of driving the large current needed for the CAN bus and has current protection against defective CAN or defective stations.

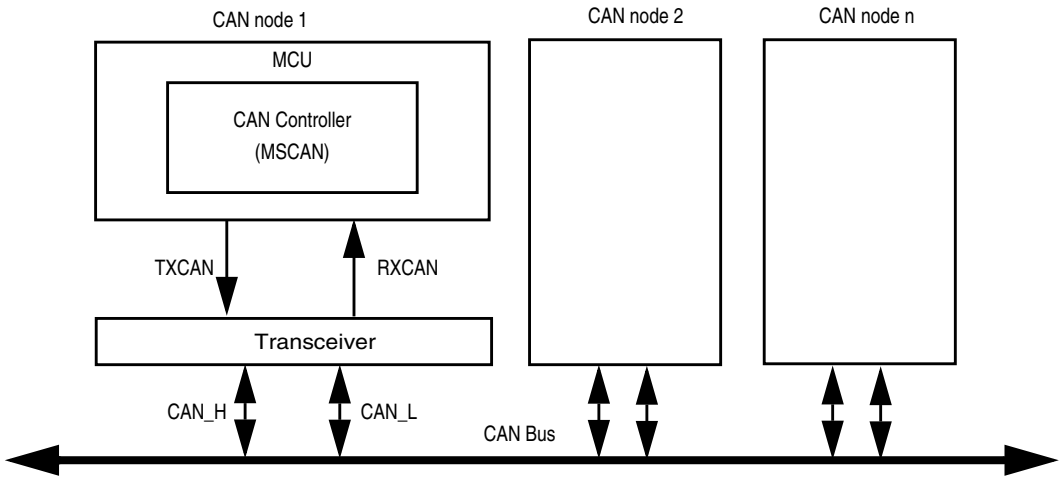

Figure 10-2. CAN System

Module Base + 0x00X3

	7	6	5	4	3	2	1	0
R W								
Reset:	x	x	x	x	x	x	x	x

= Unused; always read 'x'

Figure 10-32. Identifier Register 3 — Standard Mapping

10.3.3.2 Data Segment Registers (DSR0-7)

The eight data segment registers, each with bits DB[7:0], contain the data to be transmitted or received. The number of bytes to be transmitted or received is determined by the data length code in the corresponding DLR register.

Module Base + 0x0004 (DSR0)
0x0005 (DSR1)
0x0006 (DSR2)
0x0007 (DSR3)
0x0008 (DSR4)
0x0009 (DSR5)
0x000A (DSR6)
0x000B (DSR7)

	7	6	5	4	3	2	1	0
R W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Reset:	x	x	x	x	x	x	x	x

Figure 10-33. Data Segment Registers (DSR0–DSR7) — Extended Identifier Mapping

Table 10-30. DSR0–DSR7 Register Field Descriptions

Field	Description
7:0 DB[7:0]	**Data bits 7:0**

13.4.4.4 Framing Errors

If the data recovery logic does not detect a logic 1 where the stop bit should be in an incoming frame, it sets the framing error flag, FE, in SCI status register 1 (SCISR1). A break character also sets the FE flag because a break character has no stop bit. The FE flag is set at the same time that the RDRF flag is set.

13.4.4.5 Baud Rate Tolerance

A transmitting device may be operating at a baud rate below or above the receiver baud rate. Accumulated bit time misalignment can cause one of the three stop bit data samples (RT8, RT9, and RT10) to fall outside the actual stop bit.A noise error will occur if the RT8, RT9, and RT10 samples are not all the same logical values. A framing error will occur if the receiver clock is misaligned in such a way that the majority of the RT8, RT9, and RT10 stop bit samples are a logic zero.

As the receiver samples an incoming frame, it re-synchronizes the RT clock on any valid falling edge within the frame. Re synchronization within frames will correct a misalignment between transmitter bit times and receiver bit times.

13.4.4.5.1 Slow Data Tolerance

Figure 13-20 shows how much a slow received frame can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.

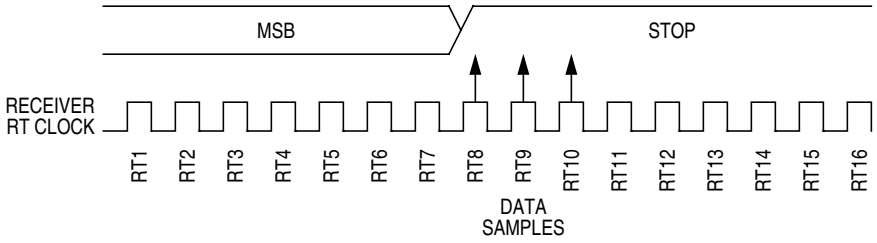

Figure 13-20. Slow Data

Let’s take RTr as receiver RT clock and RTt as transmitter RT clock.

For an 8-bit data character, it takes the receiver 9 bit times x 16 RTr cycles +7 RTr cycles =151 RTr cycles to start data sampling of the stop bit.

With the misaligned character shown in Figure 13-20, the receiver counts 151 RTr cycles at the point when the count of the transmitting device is 9 bit times x 16 RTt cycles = 144 RTt cycles.

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit data character with no errors is:

$$((151 - 144) / 151) \times 100 = 4.63\%$$

For a 9-bit data character, it takes the receiver 10 bit times x 16 RTr cycles + 7 RTr cycles = 167 RTr cycles to start data sampling of the stop bit.

15.1.2 Modes of Operation

Stop: Timer is off because clocks are stopped.

Freeze: Timer counter keep on running, unless TSFRZ in TSCR (0x0006) is set to 1.

Wait: Counters keep on running, unless TSWAI in TSCR (0x0006) is set to 1.

Normal: Timer counter keep on running, unless TEN in TSCR (0x0006) is cleared to 0.

15.1.3 Block Diagrams

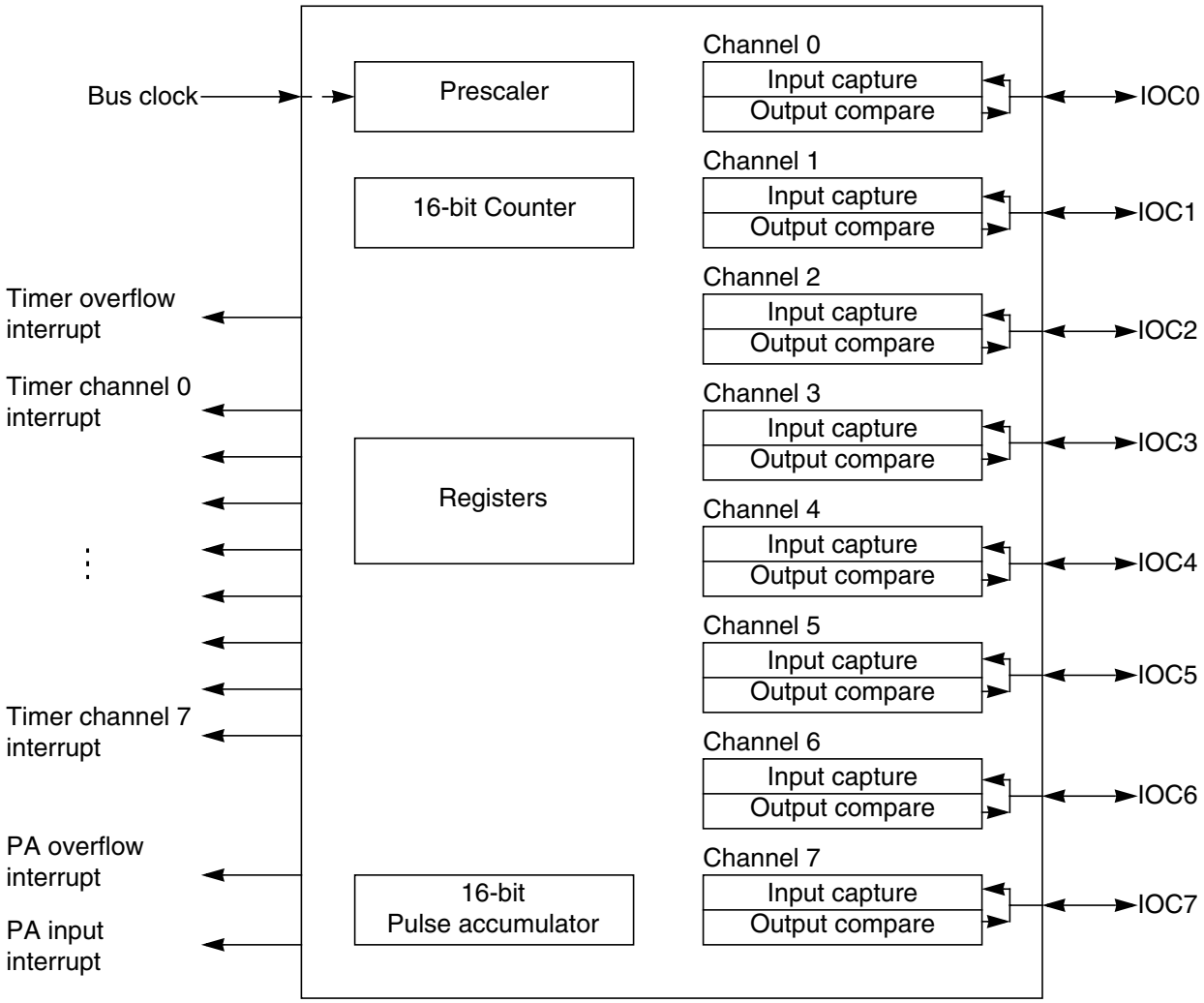

Figure 15-1. TIM16B8CV1 Block Diagram

Chapter 17
16 Kbyte Flash Module (S12FTS16KV1)

17.1 Introduction

The FTS16K module implements a 16 Kbyte Flash (nonvolatile) memory. The Flash memory contains one array of 16 Kbytes organized as 256 rows of 64 bytes with an erase sector size of eight rows (512 bytes). The Flash array may be read as either bytes, aligned words, or misaligned words. Read access time is one bus cycle for byte and aligned word, and two bus cycles for misaligned words.

The Flash array is ideal for program and data storage for single-supply applications allowing for field reprogramming without requiring external voltage sources for program or erase. Program and erase functions are controlled by a command driven interface. The Flash module supports both mass erase and sector erase. An erased bit reads 1 and a programmed bit reads 0. The high voltage required to program and erase is generated internally. It is not possible to read from a Flash array while it is being erased or programmed.

CAUTION

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

17.1.1 Glossary

Command Write Sequence — A three-step MCU instruction sequence to program, erase, or erase verify the Flash array memory.

17.1.2 Features

- 16 Kbytes of Flash memory comprised of one 16 Kbyte array divided into 32 sectors of 512 bytes
- Automated program and erase algorithm
- Interrupts on Flash command completion and command buffer empty
- Fast sector erase and word program operation
- 2-stage command pipeline for faster multi-word program times
- Flexible protection scheme to prevent accidental program or erase
- Single power supply for Flash program and erase operations
- Security feature to prevent unauthorized access to the Flash array memory

then FCLKDIV register bits PRDIV8 and FDIV[5:0] are to be set as described in Figure 17-21.

For example, if the oscillator clock frequency is 950 kHz and the bus clock is 10 MHz, FCLKDIV bits FDIV[5:0] should be set to 4 (000100) and bit PRDIV8 set to 0. The resulting FCLK is then 190 kHz. As a result, the Flash algorithm timings are increased over optimum target by:

$$(200 - 190) / 200 \times 100 \; = \; 5\%$$

Command execution time will increase proportionally with the period of FCLK.

CAUTION

Because of the impact of clock synchronization on the accuracy of the functional timings, programming or erasing the Flash array cannot be performed if the bus clock runs at less than 1 MHz. Programming or erasing the Flash array with an input clock < 150 kHz should be avoided. Setting FCLKDIV to a value such that FCLK < 150 kHz can destroy the Flash array due to overstress. Setting FCLKDIV to a value such that (1/FCLK + Tbus) < 5µs can result in incomplete programming or erasure of the Flash array cells.

If the FCLKDIV register is written, the bit FDIVLD is set automatically. If the FDIVLD bit is 0, the FCLKDIV register has not been written since the last reset. If the FCLKDIV register has not been written to, the Flash command loaded during a command write sequence will not execute and the ACCERR flag in the FSTAT register will set.

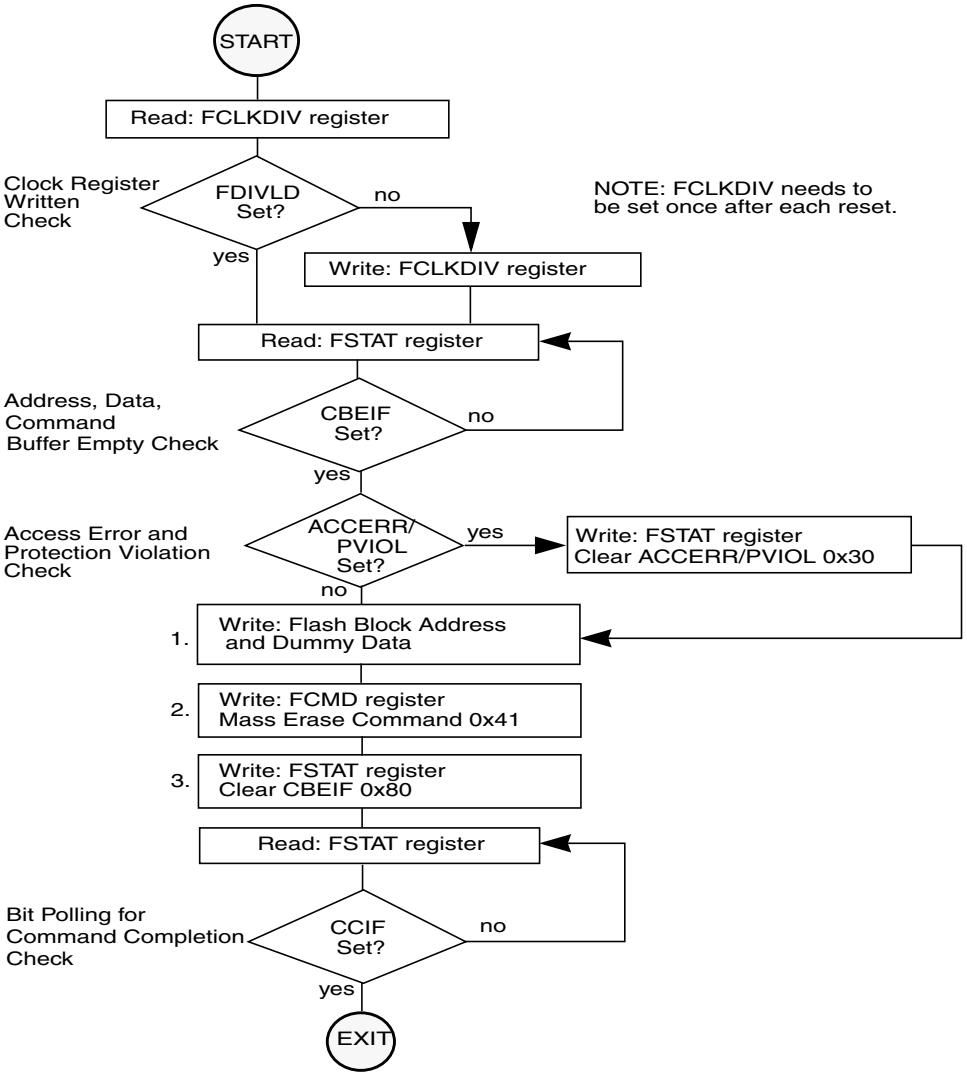

Figure 17-25. Example Mass Erase Command Flow

Chapter 19
64 Kbyte Flash Module (S12FTS64KV4)

19.1 Introduction

The FTS128K1FTS64K module implements a 12864 Kbyte Flash (nonvolatile) memory. The Flash memory contains one array of 12864 Kbytes organized as 1024512 rows of 128128 bytes with an erase sector size of eight rows (10241024 bytes). The Flash array may be read as either bytes, aligned words, or misaligned words. Read access time is one bus cycle for byte and aligned word, and two bus cycles for misaligned words.

The Flash array is ideal for program and data storage for single-supply applications allowing for field reprogramming without requiring external voltage sources for program or erase. Program and erase functions are controlled by a command driven interface. The Flash module supports both mass erase and sector erase. An erased bit reads 1 and a programmed bit reads 0. The high voltage required to program and erase is generated internally. It is not possible to read from a Flash array while it is being erased or programmed.

CAUTION

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

19.1.1 Glossary

Command Write Sequence — A three-step MCU instruction sequence to program, erase, or erase verify the Flash array memory.

19.1.2 Features

- 12864 Kbytes of Flash memory comprised of one 12864 Kbyte array divided into 12864 sectors of 10241024 bytes
- Automated program and erase algorithm
- Interrupts on Flash command completion and command buffer empty
- Fast sector erase and word program operation
- 2-stage command pipeline for faster multi-word program times
- Flexible protection scheme to prevent accidental program or erase
- Single power supply for Flash program and erase operations
- Security feature to prevent unauthorized access to the Flash array memory

19.3.2.11 RESERVED3

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x000C

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

= Unimplemented or Reserved

Figure 19-20. RESERVED3

All bits read 0 and are not writable.

19.3.2.12 RESERVED4

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x000D

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

= Unimplemented or Reserved

Figure 19-21. RESERVED4

All bits read 0 and are not writable.

19.3.2.13 RESERVED5

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x000E

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

= Unimplemented or Reserved

Figure 19-22. RESERVED5

All bits read 0 and are not writable.

20.4.2 Operating Modes

20.4.2.1 Wait Mode

If the MCU enters wait mode while a Flash command is active (CCIF = 0), that command and any buffered command will be completed.

The Flash module can recover the MCU from wait mode if the interrupts are enabled (see Section 20.4.5).

20.4.2.2 Stop Mode

If the MCU enters stop mode while a Flash command is active (CCIF = 0), that command will be aborted and the data being programmed or erased is lost. The high voltage circuitry to the Flash array will be switched off when entering stop mode. CCIF and ACCERR flags will be set. Upon exit from stop mode, the CBEIF flag will be set and any buffered command will not be executed. The ACCERR flag must be cleared before returning to normal operation.

NOTE

As active Flash commands are immediately aborted when the MCU enters stop mode, it is strongly recommended that the user does not use the STOP instruction during program and erase execution.

20.4.2.3 Background Debug Mode

In background debug mode (BDM), the FPROT register is writable. If the MCU is unsecured, then all Flash commands listed in Table 20-17 can be executed. If the MCU is secured and is in special single chip mode, the only possible command to execute is mass erase.

20.4.3 Flash Module Security

The Flash module provides the necessary security information to the MCU. After each reset, the Flash module determines the security state of the MCU as defined in Section 20.3.2.2, “Flash Security Register (FSEC)”.

The contents of the Flash security/options byte at address 0xFF0F in the Flash configuration field must be changed directly by programming address 0xFF0F when the device is unsecured and the higher address sector is unprotected. If the Flash security/options byte is left in the secure state, any reset will cause the MCU to return to the secure operating mode.

20.4.3.1 Unsecuring the MCU using Backdoor Key Access

The MCU may only be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor key (four 16-bit words programmed at addresses 0xFF00–0xFF07). If KEYEN[1:0] = 1:0 and the KEYACC bit is set, a write to a backdoor key address in the Flash array triggers a comparison between the written data and the backdoor key data stored in the Flash array. If all four words of data are written to the correct addresses in the correct order and the data matches the backdoor key stored in the Flash array, the MCU will be unsecured. The data must be written to the backdoor key

How to Reach Us:

Home Page:
www.freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
1-800-521-6274 or 480-768-2130

Europe, Middle East, and Africa:
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)

Japan:
Freescale Semiconductor Japan Ltd.
Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-0047, Japan
0120-191014 or +81-3-3440-3569

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
852-26668334

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150

MC9S12C128
Rev 01.24

05/2010