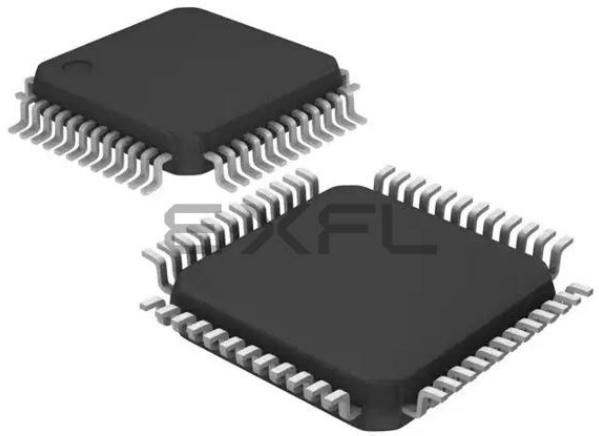


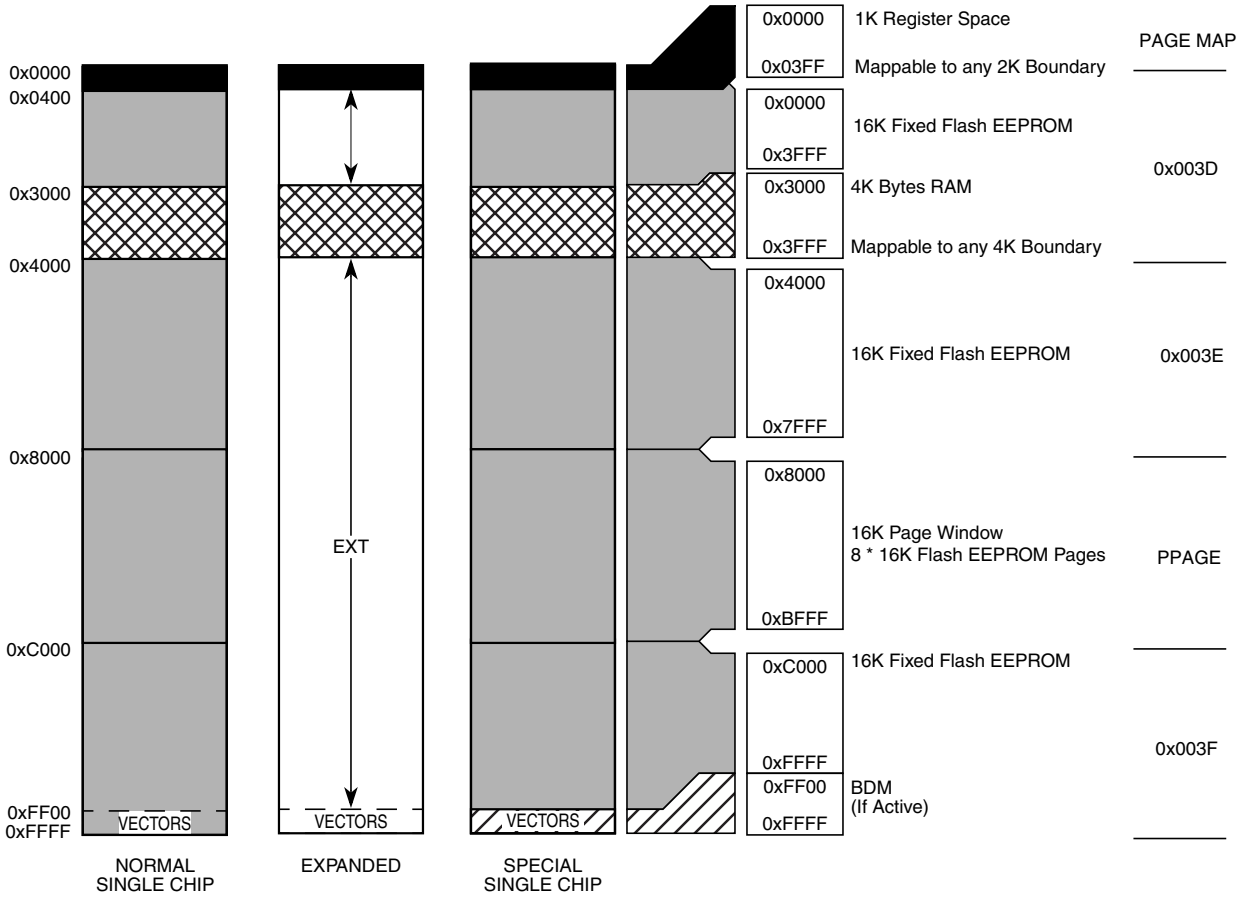
**Welcome to E-XFL.COM**

## 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 | Not For New Designs |
| 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 | 64KB (64K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 4K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.35V ~ 5.5V |
| Data Converters | A/D 8x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 48-LQFP |
| Supplier Device Package | 48-LQFP (7x7) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12c64j2cfae |




The figure shows a useful map, which is not the map out of reset. After reset the map is:

0x0000–0x03FF: Register Space
0x0000–0x0FFF: 4K RAM (only 3K visible 0x0400–0x0FFF)

Flash erase sector size is 1024 bytes

**Figure 1-2. MC9S12C128 and MC9S12GC128 User Configurable Memory Map**

**Table 3-20. 48K Byte Physical FLASH/ROM Allocated**

| Address Space | Page Window Access | ROMHM | ECS | XAB19:14 |
|---|---|---|---|---|
| 0x0000–0x3FFF | N/A | N/A | 1 | 0x3D |
| 0x4000–0x7FFF | N/A | 0 | 0 | 0x3E |
| | N/A | 1 | 1 | |
| 0x8000–0xBFFF | External | N/A | 1 | PIX[5:0] |
| | Internal | N/A | 0 | |
| 0xC000–0xFFFF | N/A | N/A | 0 | 0x3F |

**Table 3-21. 64K Byte Physical FLASH/ROM Allocated**

| Address Space | Page Window Access | ROMHM | ECS | XAB19:14 |
|---|---|---|---|---|
| 0x0000–0x3FFF | N/A | 0 | 0 | 0x3D |
| | N/A | 1 | 1 | |
| 0x4000–0x7FFF | N/A | 0 | 0 | 0x3E |
| | N/A | 1 | 1 | |
| 0x8000–0xBFFF | External | N/A | 1 | PIX[5:0] |
| | Internal | N/A | 0 | |
| 0xC000–0xFFFF | N/A | N/A | 0 | 0x3F |

### 4.3.2.5 Reserved Registers

Module Base + 0x0004
Starting address location affected by INITRG register setting.

| | 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 4-6. Reserved Register**

Module Base + 0x0005
Starting address location affected by INITRG register setting.

| | 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 4-7. Reserved Register**

Module Base + 0x0006
Starting address location affected by INITRG register setting.

| | 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 4-8. Reserved Register**

Module Base + 0x0007
Starting address location affected by INITRG register setting.

| | 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 4-9. Reserved Register**

## 6.3 Memory Map and Register Definition

A summary of the registers associated with the BDM is shown in Figure 6-2. Registers are accessed by host-driven communications to the BDM hardware using READ_BD and WRITE_BD commands. Detailed descriptions of the registers and associated bits are given in the subsections that follow.

### 6.3.1 Module Memory Map

**Table 6-1. INT Memory Map**

| Register Address | Use | Access |
|---|---|---|
| 0xFF00 | Reserved | — |
| 0xFF01 | BDM Status Register (BDMSTS) | R/W |
| 0xFF02–0xFF05 | Reserved | — |
| 0xFF06 | BDM CCR Holding Register (BDMCCR) | R/W |
| 0xFF07 | BDM Internal Register Position (BDMINR) | R |
| 0xFF08–0xFF0B | Reserved | — |

**Table 8-8. Clock Prescaler Values**

| Prescale Value | Total Divisor Value | Maximum Bus Clock[1] | Minimum Bus Clock[2] |
|---|---|---|---|
| 00000 | Divide by 2 | 4 MHz | 1 MHz |
| 00001 | Divide by 4 | 8 MHz | 2 MHz |
| 00010 | Divide by 6 | 12 MHz | 3 MHz |
| 00011 | Divide by 8 | 16 MHz | 4 MHz |
| 00100 | Divide by 10 | 20 MHz | 5 MHz |
| 00101 | Divide by 12 | 24 MHz | 6 MHz |
| 00110 | Divide by 14 | 28 MHz | 7 MHz |
| 00111 | Divide by 16 | 32 MHz | 8 MHz |
| 01000 | Divide by 18 | 36 MHz | 9 MHz |
| 01001 | Divide by 20 | 40 MHz | 10 MHz |
| 01010 | Divide by 22 | 44 MHz | 11 MHz |
| 01011 | Divide by 24 | 48 MHz | 12 MHz |
| 01100 | Divide by 26 | 52 MHz | 13 MHz |
| 01101 | Divide by 28 | 56 MHz | 14 MHz |
| 01110 | Divide by 30 | 60 MHz | 15 MHz |
| 01111 | Divide by 32 | 64 MHz | 16 MHz |
| 10000 | Divide by 34 | 68 MHz | 17 MHz |
| 10001 | Divide by 36 | 72 MHz | 18 MHz |
| 10010 | Divide by 38 | 76 MHz | 19 MHz |
| 10011 | Divide by 40 | 80 MHz | 20 MHz |
| 10100 | Divide by 42 | 84 MHz | 21 MHz |
| 10101 | Divide by 44 | 88 MHz | 22 MHz |
| 10110 | Divide by 46 | 92 MHz | 23 MHz |
| 10111 | Divide by 48 | 96 MHz | 24 MHz |
| 11000 | Divide by 50 | 100 MHz | 25 MHz |
| 11001 | Divide by 52 | 104 MHz | 26 MHz |
| 11010 | Divide by 54 | 108 MHz | 27 MHz |
| 11011 | Divide by 56 | 112 MHz | 28 MHz |
| 11100 | Divide by 58 | 116 MHz | 29 MHz |
| 11101 | Divide by 60 | 120 MHz | 30 MHz |
| 11110 | Divide by 62 | 124 MHz | 31 MHz |
| 11111 | Divide by 64 | 128 MHz | 32 MHz |

1. Maximum ATD conversion clock frequency is 2 MHz. The maximum allowed bus clock frequency is shown in this column.
2. Minimum ATD conversion clock frequency is 500 kHz. The minimum allowed bus clock frequency is shown in this column.

## 8.7 Interrupts

The interrupt requested by the ATD10B8C is listed in Table 8-20. Refer to MCU specification for related vector address and priority.

**Table 8-20. ATD10B8C Interrupt Vectors**

| Interrupt Source | CCR Mask | Local Enable |
|---|---|---|
| Sequence complete interrupt | I bit | ASCIE in ATDCTL2 |

See Section 8.3.2, “Register Descriptions” for further details.

of 4 OSCCLK plus 4 PLLCLK cycles to make the transition. During the transition, all clocks freeze and CPU activity ceases.

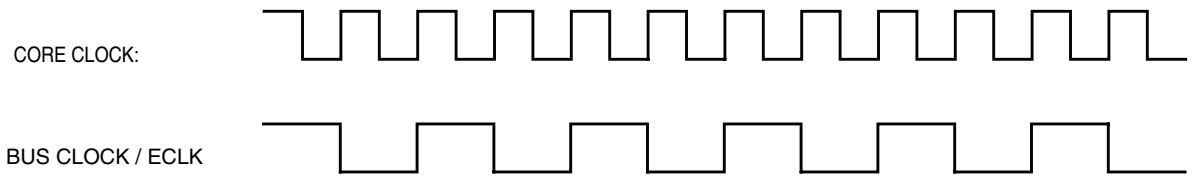



**Figure 9-18. Core Clock and Bus Clock Relationship**

## 9.4.3 Clock Monitor (CM)

If no OSCCLK edges are detected within a certain time, the clock monitor within the oscillator block generates a clock monitor fail event. The CRGV4 then asserts self-clock mode or generates a system reset depending on the state of SCME bit. If the clock monitor is disabled or the presence of clocks is detected no failure is indicated by the oscillator block.The clock monitor function is enabled/disabled by the CME control bit.

## 9.4.4 Clock Quality Checker

The clock monitor performs a coarse check on the incoming clock signal. The clock quality checker provides a more accurate check in addition to the clock monitor.

A clock quality check is triggered by any of the following events:

- Power-on reset (POR)
- Low voltage reset (LVR)
- Wake-up from full stop mode (exit full stop)
- Clock monitor fail indication (CM fail)

A time window of 50000 VCO clock cycles[1] is called *check window*.

A number greater equal than 4096 rising OSCCLK edges within a *check window* is called *osc ok*. Note that *osc ok* immediately terminates the current *check window*. See Figure 9-19 as an example.

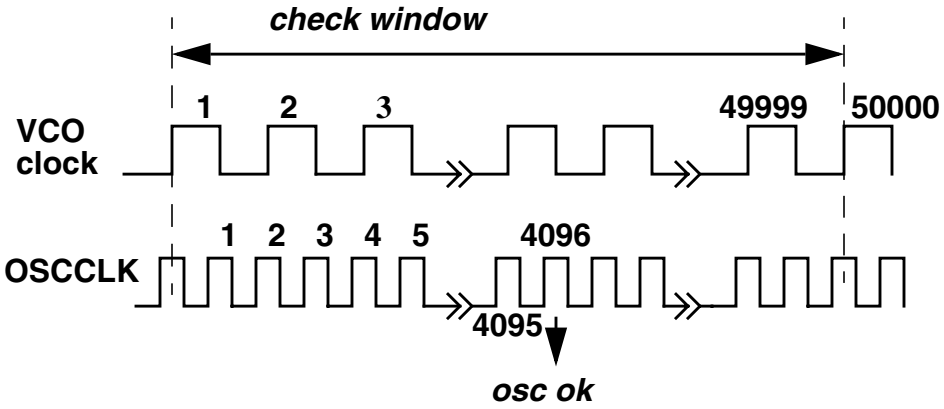



**Figure 9-19. Check Window Example**

1. VCO clock cycles are generated by the PLL when running at minimum frequency $f_{SCM}$.

- The channel is disabled

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

**NOTE**

Reads of this register return the most recent value written. Reads do not necessarily return the value of the currently active period due to the double buffering scheme.

Reference Section 12.4.2.3, “PWM Period and Duty,” for more information.

To calculate the output period, take the selected clock source period for the channel of interest (A, B, SA, or SB) and multiply it by the value in the period register for that channel:

- Left aligned output (CAEx = 0)
- PWMx period = channel clock period * PWMPERx center aligned output (CAEx = 1)
- PWMx period = channel clock period * (2 * PWMPERx)

For boundary case programming values, please refer to Section 12.4.2.8, “PWM Boundary Cases.”

Module Base + 0x0012

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 12-21. PWM Channel Period Registers (PWMPER0)**

Module Base + 0x0013

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 12-22. PWM Channel Period Registers (PWMPER1)**

Module Base + 0x0014

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 12-23. PWM Channel Period Registers (PWMPER2)**

# Chapter 13
# Serial Communications Interface (S12SCIV2) Block Description

## 13.1 Introduction

This block guide provide an overview of serial communication interface (SCI) module. The SCI allows asynchronous serial communications with peripheral devices and other CPUs.

### 13.1.1 Glossary

IRQ — Interrupt Request

LSB — Least Significant Bit

MSB — Most Significant Bit

NRZ — Non-Return-to-Zero

RZI — Return-to-Zero-Inverted

RXD — Receive Pin

SCI — Serial Communication Interface

TXD — Transmit Pin

### 13.1.2 Features

The SCI includes these distinctive features:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- 13-bit baud rate selection
- Programmable 8-bit or 9-bit data format
- Separately enabled transmitter and receiver
- Programmable transmitter output parity
- Two receiver wake up methods:
  - Idle line wake-up
  - Address mark wake-up
- Interrupt-driven operation with eight flags:
  - Transmitter empty

**Table 15-2. TIM16B8CV1 Memory Map**

| Address Offset | Use | Access |
|---|---|---|
| 0x0000 | Timer Input Capture/Output Compare Select (TIOS) | R/W |
| 0x0001 | Timer Compare Force Register (CFORC) | R/W(1) |
| 0x0002 | Output Compare 7 Mask Register (OC7M) | R/W |
| 0x0003 | Output Compare 7 Data Register (OC7D) | R/W |
| 0x0004 | Timer Count Register (TCNT(hi)) | R/W(2) |
| 0x0005 | Timer Count Register (TCNT(lo)) | R/W2 |
| 0x0006 | Timer System Control Register1 (TSCR1) | R/W |
| 0x0007 | Timer Toggle Overflow Register (TTOV) | R/W |
| 0x0008 | Timer Control Register1 (TCTL1) | R/W |
| 0x0009 | Timer Control Register2 (TCTL2) | R/W |
| 0x000A | Timer Control Register3 (TCTL3) | R/W |
| 0x000B | Timer Control Register4 (TCTL4) | R/W |
| 0x000C | Timer Interrupt Enable Register (TIE) | R/W |
| 0x000D | Timer System Control Register2 (TSCR2) | R/W |
| 0x000E | Main Timer Interrupt Flag1 (TFLG1) | R/W |
| 0x000F | Main Timer Interrupt Flag2 (TFLG2) | R/W |
| 0x0010 | Timer Input Capture/Output Compare Register 0 (TC0(hi)) | R/W(3) |
| 0x0011 | Timer Input Capture/Output Compare Register 0 (TC0(lo)) | R/W3 |
| 0x0012 | Timer Input Capture/Output Compare Register 1 (TC1(hi)) | R/W3 |
| 0x0013 | Timer Input Capture/Output Compare Register 1 (TC1(lo)) | R/W3 |
| 0x0014 | Timer Input Capture/Output Compare Register 2 (TC2(hi)) | R/W3 |
| 0x0015 | Timer Input Capture/Output Compare Register 2 (TC2(lo)) | R/W3 |
| 0x0016 | Timer Input Capture/Output Compare Register 3 (TC3(hi)) | R/W3 |
| 0x0017 | Timer Input Capture/Output Compare Register 3 (TC3(lo)) | R/W3 |
| 0x0018 | Timer Input Capture/Output Compare Register4 (TC4(hi)) | R/W3 |
| 0x0019 | Timer Input Capture/Output Compare Register 4 (TC4(lo)) | R/W3 |
| 0x001A | Timer Input Capture/Output Compare Register 5 (TC5(hi)) | R/W3 |
| 0x001B | Timer Input Capture/Output Compare Register 5 (TC5(lo)) | R/W3 |
| 0x001C | Timer Input Capture/Output Compare Register 6 (TC6(hi)) | R/W3 |
| 0x001D | Timer Input Capture/Output Compare Register 6 (TC6(lo)) | R/W3 |
| 0x001E | Timer Input Capture/Output Compare Register 7 (TC7(hi)) | R/W3 |
| 0x001F | Timer Input Capture/Output Compare Register 7 (TC7(lo)) | R/W3 |
| 0x0020 | 16-Bit Pulse Accumulator Control Register (PACTL) | R/W |
| 0x0021 | Pulse Accumulator Flag Register (PAFLG) | R/W |
| 0x0022 | Pulse Accumulator Count Register (PACNT(hi)) | R/W |
| 0x0023 | Pulse Accumulator Count Register (PACNT(lo)) | R/W |
| 0x0024 – 0x002C | Reserved | — (4) |
| 0x002D | Timer Test Register (TIMTST) | R/W2 |
| 0x002E – 0x002F | Reserved | — 4 |

1. Always read 0x0000.
2. Only writable in special modes (test_mode = 1).
3. Write to these registers have no meaning or effect during input capture.
4. Write has no effect; return 0 on read

**Table 15-15. TSCR2 Field Descriptions**

| Field | Description |
|---|---|
| 7<br>TOI | **Timer Overflow Interrupt Enable**<br>0 Interrupt inhibited.<br>1 Hardware interrupt requested when TOF flag set. |
| 3<br>TCRE | **Timer Counter Reset Enable** — This bit allows the timer counter to be reset by a successful output compare 7 event. This mode of operation is similar to an up-counting modulus counter.<br>0 Counter reset inhibited and counter free runs.<br>1 Counter reset by a successful output compare 7.<br>**Note:** If TC7 = 0x0000 and TCRE = 1, TCNT will stay at 0x0000 continuously. If TC7 = 0xFFFF and TCRE = 1, TOF will never be set when TCNT is reset from 0xFFFF to 0x0000.<br>**Note:** TCRE=1 and TC7!=0, the TCNT cycle period will be TC7 x "prescaler counter width" + "1 Bus Clock", for a more detail explanation please refer to Section 15.4.3, “Output Compare |
| 2<br>PR[2:0] | **Timer Prescaler Select** — These three bits select the frequency of the timer prescaler clock derived from the Bus Clock as shown in Table 15-16. |

**Table 15-16. Timer Clock Selection**

| PR2 | PR1 | PR0 | Timer Clock |
|---|---|---|---|
| 0 | 0 | 0 | Bus Clock / 1 |
| 0 | 0 | 1 | Bus Clock / 2 |
| 0 | 1 | 0 | Bus Clock / 4 |
| 0 | 1 | 1 | Bus Clock / 8 |
| 1 | 0 | 0 | Bus Clock / 16 |
| 1 | 0 | 1 | Bus Clock / 32 |
| 1 | 1 | 0 | Bus Clock / 64 |
| 1 | 1 | 1 | Bus Clock / 128 |

**NOTE**

The newly selected prescale factor will not take effect until the next synchronized edge where all prescale counter stages equal zero.

### 15.3.2.12 Main Timer Interrupt Flag 1 (TFLG1)

Module Base + 0x000E

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | C7F | C6F | C5F | C4F | C3F | C2F | C1F | C0F |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 15-20. Main Timer Interrupt Flag 1 (TFLG1)**

Read: Anytime

# 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

To change the Flash protection that will be loaded on reset, the upper sector of the Flash array must be unprotected, then the Flash protection byte located at Flash address 0xFF0D must be written to.

A protected Flash sector is disabled by FPHDIS while the size of the protected sector is defined by FPHS[1:0] in the FPROT register.

Trying to alter any of the protected areas will result in a protect violation error and the PVIOL flag will be set in the FSTAT register (see Section 17.3.2.6). A mass erase of the whole Flash array is only possible when protection is fully disabled by setting the FPOPEN and FPHDIS bits. An attempt to mass erase a Flash array while protection is enabled will set the PVIOL flag in the FSTAT register.

**Table 17-8. FPROT Field Descriptions**

| Field | Description |
|---|---|
| 7<br>FPOPEN | **Protection Function for Program or Erase** — The FPOPEN bit is used to either select an address range to be protected using the FPHDIS and FPHS[1:0] bits or to select the same address range to be unprotected as shown in Table 17-9.<br>0 The FPHDIS bit allows a Flash address range to be unprotected<br>1 The FPHDIS bit allows a Flash address range to be protected |
| 6<br>NV6 | **Nonvolatile Flag Bit** — The NV6 bit should remain in the erased state for future enhancements. |
| 5<br>FPHDIS | **Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in the higher space of the Flash address map.<br>0 Protection/unprotection enabled<br>1 Protection/unprotection disabled |
| 4–3<br>FPHS[1:0] | **Flash Protection Higher Address Size** — The FPHS[1:0] bits determine the size of the protected/unprotected sector as shown in Table 17-10. The FPHS[1:0] bits can only be written to while the FPHDIS bit is set. |
| 2–0<br>NV[2:0] | **Nonvolatile Flag Bits** — The NV[2:0] bits should remain in the erased state for future enhancements. |

**Table 17-9. Flash Protection Function**

| FPOPEN | FPHDIS | FPHS1 | FPHS0 | Function(1) |
|---|---|---|---|---|
| 1 | 1 | x | x | No protection |
| 1 | 0 | x | x | Protect high range |
| 0 | 1 | x | x | Full Flash array protected |
| 0 | 0 | x | x | Unprotected high range |

1. For range sizes refer to Table 17-10.

**Table 17-10. Flash Protection Higher Address Range**

| FPHS[1:0] | Address Range | Range Size |
|---|---|---|
| 00 | 0xF800–0xFFFF | 2 Kbytes |
| 01 | 0xF000–0xFFFF | 4 Kbytes |
| 10 | 0xE000–0xFFFF | 8 Kbytes |
| 11 | 0xC000–0xFFFF | 16 Kbytes |

**Table 20-15. FCMD Field Descriptions**

| Field | Description |
|---|---|
| 6, 5, 2, 0<br>CMDB[6:5]<br>CMDB[2]<br>CMDB[0] | Valid Flash commands are shown in Table 20-16. An attempt to execute any command other than those listed in Table 20-16 will set the ACCERR bit in the FSTAT register (see Section 20.3.2.6). |

**Table 20-16. Valid Flash Command List**

| CMDB | NVM Command |
|---|---|
| 0x05 | Erase verify |
| 0x20 | Word program |
| 0x40 | Sector erase |
| 0x41 | Mass erase |

### 20.3.2.8 RESERVED2

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

Module Base + 0x0007

| | 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 20-14. RESERVED2**

All bits read 0 and are not writable.

### 20.3.2.9 Flash Address Register (FADDR)

FADDRHI and FADDRLO are the Flash address registers.

Module Base + 0x0008

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | FABHI | | | | | | | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 20-15. Flash Address High Register (FADDRHI)**

#### 20.4.1.2 Command Write Sequence

The Flash command controller is used to supervise the command write sequence to execute program, erase, and erase verify algorithms.

Before starting a command write sequence, the ACCERR and PVIOL flags in the FSTAT register must be clear and the CBEIF flag should be tested to determine the state of the address, data, and command buffers. If the CBEIF flag is set, indicating the buffers are empty, a new command write sequence can be started. If the CBEIF flag is clear, indicating the buffers are not available, a new command write sequence will overwrite the contents of the address, data, and command buffers.

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

1. Write to a valid address in the Flash array memory.
2. Write a valid command to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the command.

The address written in step 1 will be stored in the FADDR registers and the data will be stored in the FDATA registers. When the CBEIF flag is cleared in step 3, the CCIF flag is cleared by the Flash command controller indicating that the command was successfully launched. For all command write sequences, the CBEIF flag will set after the CCIF flag is cleared indicating that the address, data, and command buffers are ready for a new command write sequence to begin. A buffered command will wait for the active operation to be completed before being launched. Once a command is launched, the completion of the command operation is indicated by the setting of the CCIF flag in the FSTAT register. The CCIF flag will set upon completion of all active and buffered commands.

#### 20.4.1.3.3 Sector Erase Command

The sector erase operation will erase all addresses in a 1024 byte sector of the Flash array using an embedded algorithm.

An example flow to execute the sector erase operation is shown in Figure 20-26. The sector erase command write sequence is as follows:

1. Write to a Flash array address to start the command write sequence for the sector erase command. The Flash address written determines the sector to be erased while MCU address bits [9:0] and the data written are ignored.
2. Write the sector erase command, 0x40, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the sector erase command.

If a Flash sector to be erased is in a protected area of the Flash array, the PVIOL flag in the FSTAT register will set and the sector erase command will not launch. Once the sector erase command has successfully launched, the CCIF flag in the FSTAT register will set after the sector erase operation has completed unless a new command write sequence has been buffered.

**Table 21-13. FSTAT Field Descriptions**

| Field | Description |
|---|---|
| 5<br>PVIOL | **Protection Violation** — The PVIOL flag indicates an attempt was made to program or erase an address in a protected Flash array memory area. The PVIOL flag is cleared by writing a 1 to PVIOL. Writing a 0 to the PVIOL flag has no effect on PVIOL. While PVIOL is set, it is not possible to launch another command.<br>0 No protection violation detected<br>1 Protection violation has occurred |
| 4<br>ACCERR | **Access Error** — The ACCERR flag indicates an illegal access to the Flash array caused by either a violation of the command write sequence, issuing an illegal command (illegal combination of the CMDBx bits in the FCMD register) or the execution of a CPU STOP instruction while a command is executing (CCIF=0). The ACCERR flag is cleared by writing a 1 to ACCERR. Writing a 0 to the ACCERR flag has no effect on ACCERR. While ACCERR is set, it is not possible to launch another command.<br>0 No access error detected<br>1 Access error has occurred |
| 2<br>BLANK | **Flash Array Has Been Verified as Erased** — The BLANK flag indicates that an erase verify command has checked the Flash array and found it to be erased. The BLANK flag is cleared by hardware when CBEIF is cleared as part of a new valid command write sequence. Writing to the BLANK flag has no effect on BLANK.<br>0 If an erase verify command has been requested, and the CCIF flag is set, then a 0 in BLANK indicates the array is not erased<br>1 Flash array verifies as erased |
| 1<br>FAIL | **Flag Indicating a Failed Flash Operation** — In special modes, the FAIL flag will set if the erase verify operation fails (Flash array verified as not erased). Writing a 0 to the FAIL flag has no effect on FAIL. The FAIL flag is cleared by writing a 1 to FAIL. While FAIL is set, it is not possible to launch another command.<br>0 Flash operation completed without error<br>1 Flash operation failed |
| 0<br>DONE | **Flag Indicating a Failed Operation is not Active** — In special modes, the DONE flag will clear if a program, erase, or erase verify operation is active.<br>0 Flash operation is active<br>1 Flash operation is not active |

### 21.3.2.7 Flash Command Register (FCMD)

The FCMD register defines the Flash commands.

Module Base + 0x0006

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | 0 | CMDB6 | CMDB5 | 0 | 0 | CMDB2 | 0 | CMDB0 |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 21-11. Flash Command Register (FCMD)**

Bits CMDB6, CMDB5, CMDB2, and CMDB0 are readable and writable during a command write sequence while bits 7, 4, 3, and 1 read 0 and are not writable.

**Table A-9. Supply Current Characteristics for Other Family Members**

| Conditions are shown in Table A-4 with internal regulator enabled unless otherwise noted | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Num** | **C** | **Rating** | **Symbol** | **Min** | **Typ** | **Max** | **Unit** |
| 1 | P | Run Supply Current Single Chip, | $I_{DD5}$ | — | — | 45 | mA |
| 2 | | Wait Supply current | $I_{DDW}$ | | | | mA |
| | P | All modules enabled | | — | — | 33 | |
| | P | VDDR<4.9V, only RTI enabled[2] | | — | 2.5 | 8 | |
| | C | VDDR>4.9V, only RTI enabled | | — | 3.5 | — | |
| 6 | | Pseudo Stop Current (RTI and COP disabled)[23] | $I_{DDPS}$[(1)] | | | | µA |
| | C | –40°C | | — | 190 | — | |
| | P | 27°C | | — | 200 | 250 | |
| | C | 85°C | | — | 300 | — | |
| | P | "C" Temp Option 100˚C | | — | 400 | 1400 | |
| | C | 105°C | | — | 450 | — | |
| | P | "V" Temp Option 120˚C | | — | 600 | 1900 | |
| | C | 125°C | | — | 650 | — | |
| | P | "M" Temp Option 140°C | | — | 1000 | 4800 | |
| 4 | | Pseudo Stop Current (RTI and COP enabled)[(2) (3)] | $I_{DDPS}$[1] | | | | µA |
| | C | –40°C | | — | 370 | — | |
| | C | 27°C | | — | 500 | — | |
| | C | 85°C | | — | 590 | — | |
| | C | 105°C | | — | 780 | — | |
| | C | 125°C | | — | 1200 | — | |
| 5 | | Stop Current [3] | $I_{DDS}$[1] | | | | µA |
| | C | –40°C | | — | 12 | — | |
| | P | 27°C | | — | 25 | 100 | |
| | C | 85°C | | — | 130 | — | |
| | P | "C" Temp Option 100˚C | | — | 160 | 1200 | |
| | C | 105°C | | — | 200 | — | |
| | P | "V" Temp Option 120˚C | | — | 350 | 1700 | |
| | C | 125°C | | — | 400 | — | |
| | P | "M" Temp Option 140°C | | — | 600 | 4500 | |

1. STOP current measured in production test at increased junction temperature, hence for Temp Option "C" the test is carried out at 100˚C although the Temperature specification is 85˚C. Similarly for "v" and "M" options the temperature used in test lies 15˚C above the temperature option specification.
2. PLL off
3. At those low power dissipation levels $T_J = T_A$ can be assumed

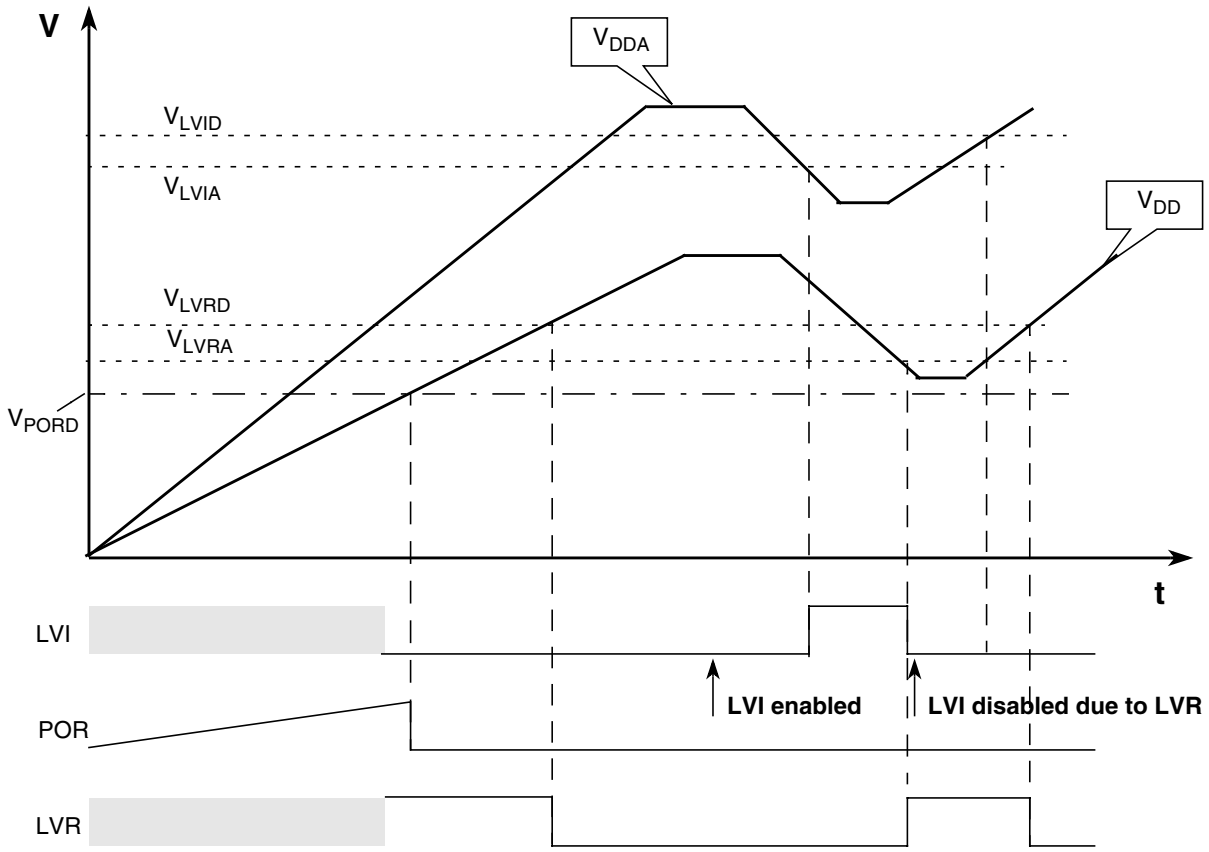



**Figure A-10. Voltage Regulator — Chip Power-up and Voltage Drops (not scaled)**

## A.7.3 Output Loads

### A.7.3.1 Resistive Loads

The on-chip voltage regulator is intended to supply the internal logic and oscillator circuits allows no external DC loads.

### A.7.3.2 Capacitive Loads

The capacitive loads are specified in Table A-24. Ceramic capacitors with X7R dielectricum are required.

**Table A-24. Voltage Regulator — Capacitive Loads**

| Num | Characteristic | Symbol | Min | Typical | Max | Unit |
|---|---|---|---|---|---|---|
| 1 | $V_{DD}$ external capacitive load | $C_{DDext}$ | 400 | 440 | 12000 | nF |
| 2 | $V_{DDPLL}$ external capacitive load | $C_{DDPLLext}$ | 90 | 220 | 5000 | nF |