**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 | 50MHz |
| Connectivity | CANbus, EBI/EMI, I²C, IrDA, SCI, SPI |
| Peripherals | LVD, POR, PWM, WDT |
| Number of I/O | 119 |
| Program Memory Size | 256KB (256K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | 4K x 8 |
| RAM Size | 16K x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.72V ~ 5.5V |
| Data Converters | A/D 24x12b |
| Oscillator Type | External |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 144-LQFP |
| Supplier Device Package | 144-LQFP (20x20) |
| Purchase URL | https://www.e-xfl.com/pro/item?MUrl=&PartUrl=s912xet256j2mag |

2.3.87 Port R Data Direction Register (DDRR) . . . 165
2.3.88 Port R Reduced Drive Register (RDRR) . . . 165
2.3.89 Port R Pull Device Enable Register (PERR) . . . 166
2.3.90 Port R Polarity Select Register (PPSR) . . . 166
2.3.91 PIM Reserved Register . . . 167
2.3.92 Port R Routing Register (PTRRR) . . . 167
2.3.93 Port L Data Register (PTL) . . . 168
2.3.94 Port L Input Register (PTIL) . . . 170
2.3.95 Port L Data Direction Register (DDRL) . . . 170
2.3.96 Port L Reduced Drive Register (RDRL) . . . 171
2.3.97 Port L Pull Device Enable Register (PERL) . . . 171
2.3.98 Port L Polarity Select Register (PPSL) . . . 172
2.3.99 Port L Wired-Or Mode Register (WOML) . . . 172
2.3.100 Port L Routing Register (PTLRR) . . . 173
2.3.101 Port F Data Register (PTF) . . . 173
2.3.102 Port F Input Register (PTIF) . . . 175
2.3.103 Port F Data Direction Register (DDRF) . . . 175
2.3.104 Port F Reduced Drive Register (RDRF) . . . 176
2.3.105 Port F Pull Device Enable Register (PERF) . . . 176
2.3.106 Port F Polarity Select Register (PPSF) . . . 177
2.3.107 PIM Reserved Register . . . 177
2.3.108 Port F Routing Register (PTFRR) . . . 177
2.4 Functional Description . . . 178
2.4.1 General . . . 178
2.4.2 Registers . . . 178
2.4.3 Pins and Ports . . . 181
2.4.4 Pin interrupts . . . 185
2.5 Initialization Information . . . 186
2.5.1 Port Data and Data Direction Register writes . . . 186

## Chapter 3
## Memory Mapping Control (S12XMMCV4)

3.1 Introduction . . . 187
3.1.1 Terminology . . . 188
3.1.2 Features . . . 188
3.1.3 S12X Memory Mapping . . . 189
3.1.4 Modes of Operation . . . 189
3.1.5 Block Diagram . . . 190
3.2 External Signal Description . . . 190
3.3 Memory Map and Registers . . . 192
3.3.1 Module Memory Map . . . 192
3.3.2 Register Descriptions . . . 193
3.4 Functional Description . . . 204
3.4.1 MCU Operating Mode . . . 204
3.4.2 Memory Map Scheme . . . 205

**Table 1-5. Derivative Dependent Flash Block Mapping (continued)**

| Device | 0x70_0000 | 0x74_0000 | 0x78_0000 | 0x7A_0000 | 0x7C_0000 | 0x7E_0000 |
|---|---|---|---|---|---|---|
| 9S12XET256<br>9S12XEA256 (1) | — | — | B1S | — | — | B0(128K) |
| 9S12XEG128<br>9S12XEA128[1] | — | — | B1S (64K) | — | — | B0 (64K) |

1. The 9S12XEA devices are special bondouts for access to extra ADC channels in 80QFP. Available in 80QFP only. WARNING: NOT PIN-COMPATIBLE WITH REST OF FAMILY.

Block B1 is divided into two 128K blocks. The XGATE is always mapped to block B1S.

On the 9S12XEG128 the flash is divided into two 64K blocks B0 and B1S, the B1S range extending from 0x78_0000 to 0x78_FFFF, the B0 range extending from 0x7F_0000 to 0x7F_FFFF.

The block B0 is a reduced size 128K block on the 256K derivative. On the larger derivatives B0 is a 256K block. The block B0 is a reduced size 64K block on the 128K derivative.

## 2.3.29 Port S Data Register (PTS)

Address 0x0248 Access: User read/write(1)

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R<br>W | PTS7 | PTST6 | PTS5 | PTS4 | PTS3 | PTS2 | PTS1 | PTS0 |
| Altern. Function | $\overline{SS0}$ | SCK0 | MOSI0 | MISO0 | TXD1 | RXD1 | TXD0 | RXD0 |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 2-27. Port S Data Register (PTS)**

1. Read: Anytime.
   Write: Anytime.

**Table 2-26. PTS Register Field Descriptions**

| Field | Description |
|---|---|
| 7<br>PTS | **Port S general purpose input/output data**—Data Register<br>Port S pin 7 is associated with the $\overline{SS}$ signal of the SPI0 module.<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. |
| 6<br>PTS | **Port S general purpose input/output data**—Data Register<br>Port S pin 6 is associated with the SCK signal of the SPI0 module.<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. |
| 5<br>PTS | **Port S general purpose input/output data**—Data Register<br>Port S pin 5 is associated with the MOSI signal of the SPI0 module.<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. |
| 4<br>PTS | **Port S general purpose input/output data**—Data Register<br>Port S pin 4 is associated with the MISO signal of the SPI0 module.<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. |
| 3<br>PTS | **Port S general purpose input/output data**—Data Register<br>Port S pin 3 is associated with the TXD signal of the SCI1 module.<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. |
| 2<br>PTS | **Port S general purpose input/output data**—Data Register<br>Port S bits 2 is associated with the RXD signal of the SCI1 module.<br>When not used with the alternative function, this pin can be used as general purpose I/O.<br>If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. |

## 3.3 Memory Map and Registers

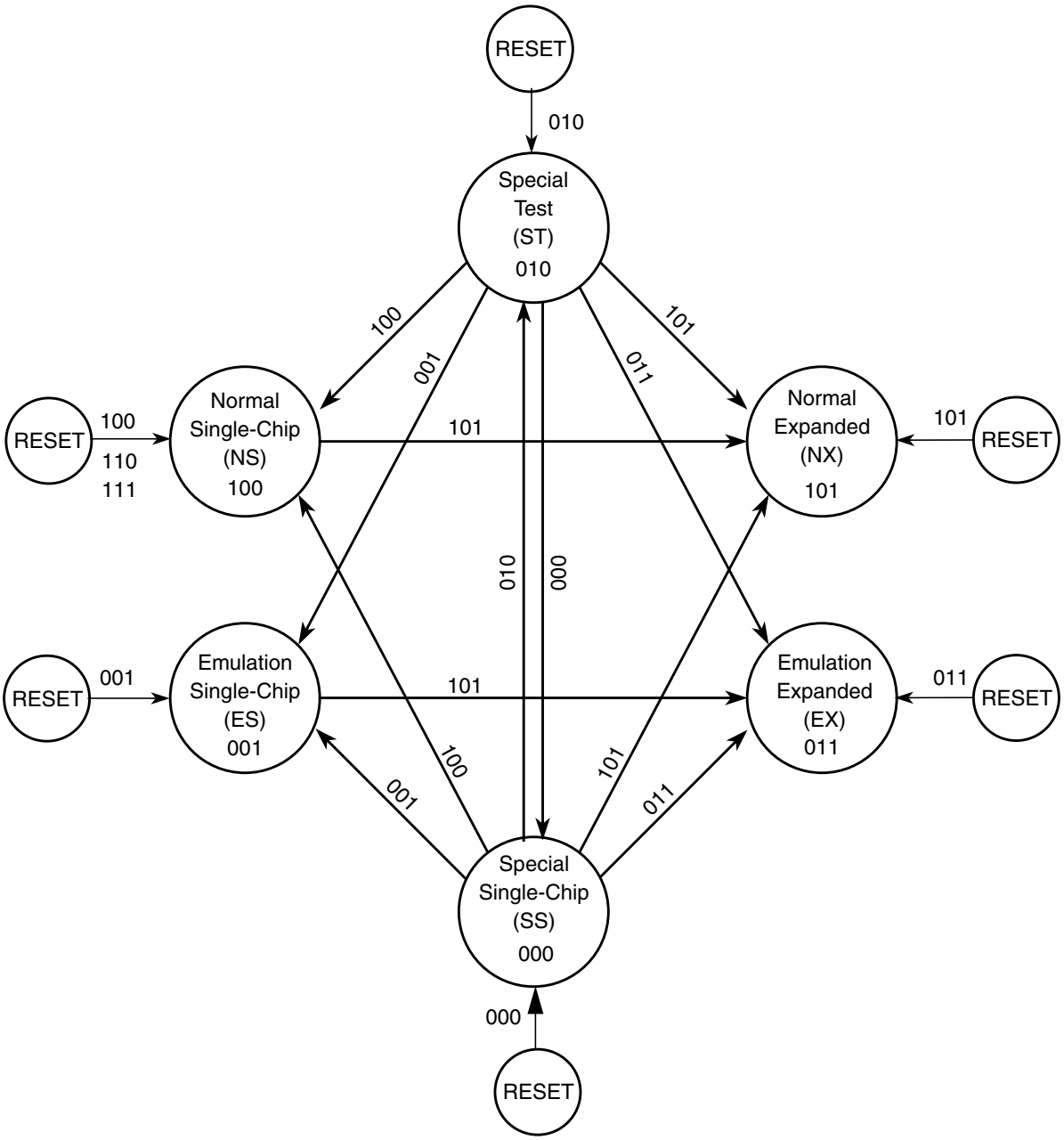
### 3.3.1 Module Memory Map

A summary of the registers associated with the MMC block is shown in Figure 3-2. Detailed descriptions of the registers and bits are given in the subsections that follow.

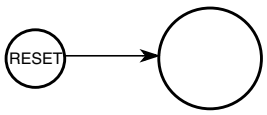
| Address | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x000A | MMCCTL0 | R | CS3E1 | CS3E0 | CS2E1 | CS2E0 | CS1E1 | CS1E0 | CS0E1 | CS0E0 |
| | | W | | | | | | | | |
| 0x000B | MODE | R | MODC | MODB | MODA | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0010 | GPAGE | R | 0 | GP6 | GP5 | GP4 | GP3 | GP2 | GP1 | GP0 |
| | | W | | | | | | | | |
| 0x0011 | DIRECT | R | DP15 | DP14 | DP13 | DP12 | DP11 | DP10 | DP9 | DP8 |
| | | W | | | | | | | | |
| 0x0012 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0013 | MMCCTL1 | R | TGMRAMON | 0 | EEEIFRON | PGMIFRON | RAMHM | EROMON | ROMHM | ROMON |
| | | W | | | | | | | | |
| 0x0014 | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x0015 | PPAGE | R | PIX7 | PIX6 | PIX5 | PIX4 | PIX3 | PIX2 | PIX1 | PIX0 |
| | | W | | | | | | | | |
| 0x0016 | RPAGE | R | RP7 | RP6 | RP5 | RP4 | RP3 | RP2 | RP1 | RP0 |
| | | W | | | | | | | | |
| 0x0017 | EPAGE | R | EP7 | EP6 | EP5 | EP4 | EP3 | EP2 | EP1 | EP0 |
| | | W | | | | | | | | |

= Unimplemented or Reserved

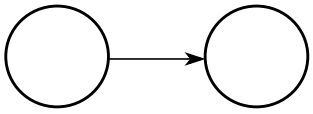
**Figure 3-2. MMC Register Summary**




Transition done by external pins (MODC, MODB, MODA)




Transition done by write access to the MODE register



110
111

Illegal (MODC, MODB, MODA) pin values.

Do not use. (Reserved for future use).

**Figure 3-5. Mode Transition Diagram when MCU is Unsecured**

internal RAM and misaligned XGATE PRR accesses in emulation modes are the only type of access that are able to produce $\overline{LSTRB}$ = ADDR0 = 1. This is summarized in Table 5-20.

**Table 5-20. Access in Emulation Modes and Special Test Mode**

| Access | $\overline{RW}$ | $\overline{LSTRB}$ | ADDR0 | DATA[15:8] | | DATA[7:0] | |
|---|---|---|---|---|---|---|---|
| | | | | I/O | data(addr) | I/O | data(addr) |
| Word write of data on DATA[15:0] at an even and even+1 address | 0 | 0 | 0 | Out | data(even) | Out | data(odd) |
| Byte write of data on DATA[7:0] at an odd address | 0 | 0 | 1 | In | x | Out | data(odd) |
| Byte write of data on DATA[15:8] at an even address | 0 | 1 | 0 | Out | data(odd) | In | x |
| Word write at an odd and odd+1 internal RAM address (misaligned — only in emulation modes) | 0 | 1 | 1 | Out | data(odd+1) | Out | data(odd) |
| Word read of data on DATA[15:0] at an even and even+1 address | 1 | 0 | 0 | In | data(even) | In | data(even+1) |
| Byte read of data on DATA[7:0] at an odd address | 1 | 0 | 1 | In | x | In | data(odd) |
| Byte read of data on DATA[15:8] at an even address | 1 | 1 | 0 | In | data(even) | In | x |
| Word read at an odd and odd+1 internal RAM address (misaligned - only in emulation modes) | 1 | 1 | 1 | In | data(odd+1) | In | data(odd) |

- Destination address of RTI, RTS, and RTC instructions.
- Vector address of interrupts, except for SWI and BDM vectors

LBRA, BRA, BSR, BGND as well as non-indexed JMP, JSR, and CALL instructions are not classified as change of flow and are not stored in the trace buffer.

COF addresses are defined as follows for the XGATE:

- Source address of taken conditional branches
- Destination address of indexed JAL instructions.
- First XGATE code address in a thread

Change-of-flow addresses stored include the full 23-bit address bus of CPU12X, the 16-bit address bus for the XGATE module and an information byte, which contains a source/destination bit to indicate whether the stored address was a source address or destination address.

**NOTE**

When an CPU12X COF instruction with destination address is executed, the destination address is stored to the trace buffer on instruction completion, indicating the COF has taken place. If an interrupt occurs simultaneously then the next instruction carried out is actually from the interrupt service routine. The instruction at the destination address of the original program flow gets exectuted after the interrupt service routine.

In the following example an IRQ interrupt occurs during execution of the indexed JMP at address MARK1. The BRN at the destination (SUB_1) is not executed until after the IRQ service routine but the destination address is entered into the trace buffer to indicate that the indexed JMP COF has taken place.

```
         LDX     #SUB_1
MARK1    JMP     0,X                    ; IRQ interrupt occurs during execution of this
MARK2    NOP                            ;

SUB_1    BRN     *                      ; JMP Destination address TRACE BUFFER ENTRY 1
                                        ; RTI Destination address TRACE BUFFER ENTRY 3
         NOP                            ;
ADDR1    DBNE    A,PART5                ; Source address TRACE BUFFER ENTRY 4

IRQ_ISR  LDAB    #$F0                   ; IRQ Vector $FFF2 = TRACE BUFFER ENTRY 2
         STAB    VAR_C1
         RTI                            ;
```

The execution flow taking into account the IRQ is as follows

```
         LDX     #SUB_1
MARK1    JMP     0,X                    ;
IRQ_ISR  LDAB    #$F0                   ;
         STAB    VAR_C1
         RTI                            ;
SUB_1    BRN     *
         NOP                            ;
ADDR1    DBNE    A,PART5                ;
```

# BGE Branch if Greater than or Equal to Zero BGE

## Operation

If N ^ V = 0, then PC + $0002 + (REL9 << 1) ⇒ PC

Branch instruction to compare signed numbers.

Branch if RS1 ≥ RS2:

```
SUB     R0,RS1,RS2
BGE     REL9
```

## CCR Effects

| N | Z | V | C |
|---|---|---|---|
| — | — | — | — |

N: Not affected.
Z: Not affected.
V: Not affected.
C: Not affected.

## Code and CPU Cycles

| Source Form | Address Mode | Machine Code | | | | | | | | Cycles |
|---|---|---|---|---|---|---|---|---|---|---|
| BGE REL9 | REL9 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | REL9 | PP/P |

## 16.3 Memory Map and Register Definition

This section provides a detailed description of all registers accessible in the MSCAN.

### 16.3.1 Module Memory Map

Figure 16-3 gives an overview on all registers and their individual bits in the MSCAN memory map. The *register address* results from the addition of *base address* and *address offset*. The *base address* is determined at the MCU level and can be found in the MCU memory map description. The *address offset* is defined at the module level.

The MSCAN occupies 64 bytes in the memory space. The base address of the MSCAN module is determined at the MCU level when the MCU is defined. The register decode map is fixed and begins at the first address of the module address offset.

The detailed register descriptions follow in the order they appear in the register map.

# Chapter 20
# Serial Communication Interface (S12SCIV5)

**Table 20-1. Revision History**

| Version Number | Revision Date | Effective Date | Author | Description of Changes |
|---|---|---|---|---|
| 05.03 | 12/25/2008 | | | remove redundancy comments in Figure1-2 |
| 05.04 | 08/05/2009 | | | fix typo, SCIBDL reset value be 0x04, not 0x00 |
| 05.05 | 06/03/2010 | | | fix typo, Table 20-4,SCICR1 Even parity should be PT=0<br>fix typo, on page 20-745,should be BKDIF,not BLDIF |

## 20.1 Introduction

This block guide provides an overview of the serial communication interface (SCI) module.

The SCI allows asynchronous serial communications with peripheral devices and other CPUs.

### 20.1.1 Glossary

IR: InfraRed

IrDA: Infrared Design Associate

IRQ: Interrupt Request

LIN: Local Interconnect Network

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

**Table 24-16. FERCNFG Field Descriptions (continued)**

| Field | Description |
|---|---|
| 3<br>ERSVIE1 | **EEE Error Type 1 Interrupt Enable** — The ERSVIE1 bit controls interrupt generation when a change state error is detected during an EEE operation.<br>0 ERSVIF1 interrupt disabled<br>1 An interrupt will be requested whenever the ERSVIF1 flag is set (see Section 24.3.2.8) |
| 2<br>ERSVIE0 | **EEE Error Type 0 Interrupt Enable** — The ERSVIE0 bit controls interrupt generation when a sector format error is detected during an EEE operation.<br>0 ERSVIF0 interrupt disabled<br>1 An interrupt will be requested whenever the ERSVIF0 flag is set (see Section 24.3.2.8) |
| 1<br>DFDIE | **Double Bit Fault Detect Interrupt Enable** — The DFDIE bit controls interrupt generation when a double bit fault is detected during a Flash block read operation.<br>0 DFDIF interrupt disabled<br>1 An interrupt will be requested whenever the DFDIF flag is set (see Section 24.3.2.8) |
| 0<br>SFDIE | **Single Bit Fault Detect Interrupt Enable** — The SFDIE bit controls interrupt generation when a single bit fault is detected during a Flash block read operation.<br>0 SFDIF interrupt disabled whenever the SFDIF flag is set (see Section 24.3.2.8)<br>1 An interrupt will be requested whenever the SFDIF flag is set (see Section 24.3.2.8) |

### 24.3.2.7 Flash Status Register (FSTAT)

The FSTAT register reports the operational status of the Flash module.

Offset Module Base + 0x0006

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | CCIF | 0 | ACCERR | FPVIOL | MGBUSY | RSVD | MGSTAT[1:0] | |
| W | | | | | | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0(1) | 0[1] |

= Unimplemented or Reserved

**Figure 24-11. Flash Status Register (FSTAT)**

1. Reset value can deviate from the value shown if a double bit fault is detected during the reset sequence (see Section 24.6).

CCIF, ACCERR, and FPVIOL bits are readable and writable, MGBUSY and MGSTAT bits are readable but not writable, while remaining bits read 0 and are not writable.

**Table 26-56. Verify Backdoor Access Key Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 100 at command launch |
| | | Set if a Load Data Field command sequence is currently active |
| | | Set if an incorrect backdoor key is supplied |
| | | Set if backdoor key access has not been enabled (KEYEN[1:0] != 10, see Section 26.3.2.2) |
| | | Set if the backdoor key has mismatched since the last reset |
| | FPVIOL | None |
| | MGSTAT1 | None |
| | MGSTAT0 | None |
| FERSTAT | EPVIOLIF | None |

### 26.4.2.13 Set User Margin Level Command

The Set User Margin Level command causes the Memory Controller to set the margin level for future read operations of a specific P-Flash or D-Flash block.

**Table 26-57. Set User Margin Level Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x0D | Global address [22:16] to identify the Flash block |
| 001 | Margin level setting | |

Upon clearing CCIF to launch the Set User Margin Level command, the Memory Controller will set the user margin level for the targeted block and then set the CCIF flag.

Valid margin level settings for the Set User Margin Level command are defined in Table 26-58.

**Table 26-58. Valid Set User Margin Level Settings**

| CCOB (CCOBIX=001) | Level Description |
|---|---|
| 0x0000 | Return to Normal Level |
| 0x0001 | User Margin-1 Level(1) |
| 0x0002 | User Margin-0 Level(2) |

1. Read margin to the erased state
2. Read margin to the programmed state

**Table 27-26. FCCOB - NVM Command Mode (Typical Usage)**

| CCOBIX[2:0] | Byte | FCCOB Parameter Fields (NVM Command Mode) |
|---|---|---|
| 011 | HI | Data 1 [15:8] |
| | LO | Data 1 [7:0] |
| 100 | HI | Data 2 [15:8] |
| | LO | Data 2 [7:0] |
| 101 | HI | Data 3 [15:8] |
| | LO | Data 3 [7:0] |

### 27.3.2.12 EEE Tag Counter Register (ETAG)

The ETAG register contains the number of outstanding words in the buffer RAM EEE partition that need to be programmed into the D-Flash EEE partition. The ETAG register is decremented prior to the related tagged word being programmed into the D-Flash EEE partition. All tagged words have been programmed into the D-Flash EEE partition once all bits in the ETAG register read 0 and the MGBUSY flag in the FSTAT register reads 0.

Offset Module Base + 0x000C

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | ETAG[15:8] | | | | | | | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 27-18. EEE Tag Counter High Register (ETAGHI)**

Offset Module Base + 0x000D

| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| R | ETAG[7:0] | | | | | | | |
| W | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented or Reserved

**Figure 27-19. EEE Tag Counter Low Register (ETAGLO)**

All ETAG bits are readable but not writable and are cleared by the Memory Controller.

### 27.3.2.13 Flash ECC Error Results Register (FECCR)

The FECCR registers contain the result of a detected ECC fault for both single bit and double bit faults. The FECCR register provides access to several ECC related fields as defined by the ECCRIX index bits in the FECCRIX register (see Section 27.3.2.4). Once ECC fault information has been stored, no other

**Table 27-70. Erase D-Flash Sector Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 001 at command launch |
| | | Set if a Load Data Field command sequence is currently active |
| | | Set if command not available in current mode (see Table 27-30) |
| | | Set if an invalid global address [22:0] is supplied |
| | | Set if a misaligned word address is supplied (global address [0] != 0) |
| | | Set if the global address [22:0] points to the D-Flash EEE partition |
| | FPVIOL | None |
| | 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 |
| FERSTAT | EPVIOLIF | None |

### 27.4.2.19 Enable EEPROM Emulation Command

The Enable EEPROM Emulation command causes the Memory Controller to enable EEE activity. EEE activity is disabled after any reset.

**Table 27-71. Enable EEPROM Emulation Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x13 | Not required |

Upon clearing CCIF to launch the Enable EEPROM Emulation command, the CCIF flag will set after the Memory Controller enables EEE operations using the contents of the EEE tag RAM and tag counter. The Full Partition D-Flash or the Partition D-Flash command must be run prior to launching the Enable EEPROM Emulation command.

**Table 27-72. Enable EEPROM Emulation Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 000 at command launch |
| | | Set if a Load Data Field command sequence is currently active |
| | | Set if Full Partition D-Flash or Partition D-Flash command not previously run |
| | FPVIOL | None |
| | MGSTAT1 | None |
| | MGSTAT0 | None |
| FERSTAT | EPVIOLIF | None |

#### 28.4.1.2.1 Define FCCOB Contents

The FCCOB parameter fields must be loaded with all required parameters for the Flash command being executed. Access to the FCCOB parameter fields is controlled via the CCOBIX bits in the FCCOBIX register (see Section 28.3.2.3).

The contents of the FCCOB parameter fields are transferred to the Memory Controller when the user clears the CCIF command completion flag in the FSTAT register (writing 1 clears the CCIF to 0). The CCIF flag will remain clear until the Flash command has completed. Upon completion, the Memory Controller will return CCIF to 1 and the FCCOB register will be used to communicate any results. The flow for a generic command write sequence is shown in Figure 28-26.

**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 required for EEE. Memory space in the D-Flash memory not required for EEE can be partitioned to provide nonvolatile memory space for applications.

**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.

**EEE (Emulated EEPROM)** — A method to emulate the small sector size features and endurance characteristics associated with an EEPROM.

**EEE IFR** — Nonvolatile information register located in the D-Flash block that contains data required to partition the D-Flash memory and buffer RAM for EEE. The EEE IFR is visible in the global memory map by setting the EEEIFRON bit in the MMCCTL1 register.

**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.

## 29.1.2 Features

### 29.1.2.1 P-Flash Features

- 1024 Kbytes of P-Flash memory composed of three 256 Kbyte Flash blocks and two 128 Kbyte Flash blocks. The 256 Kbyte Flash block consists of two 128 Kbyte sections each divided into 128 sectors of 1024 bytes. The 128 Kbyte Flash blocks are each divided into 128 sectors of 1024 bytes.
- Single bit fault correction and double bit fault detection within a 64-bit phrase during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and phrase program operation
- Ability to program up to one phrase in each P-Flash block simultaneously
- Flexible protection scheme to prevent accidental program or erase of P-Flash memory

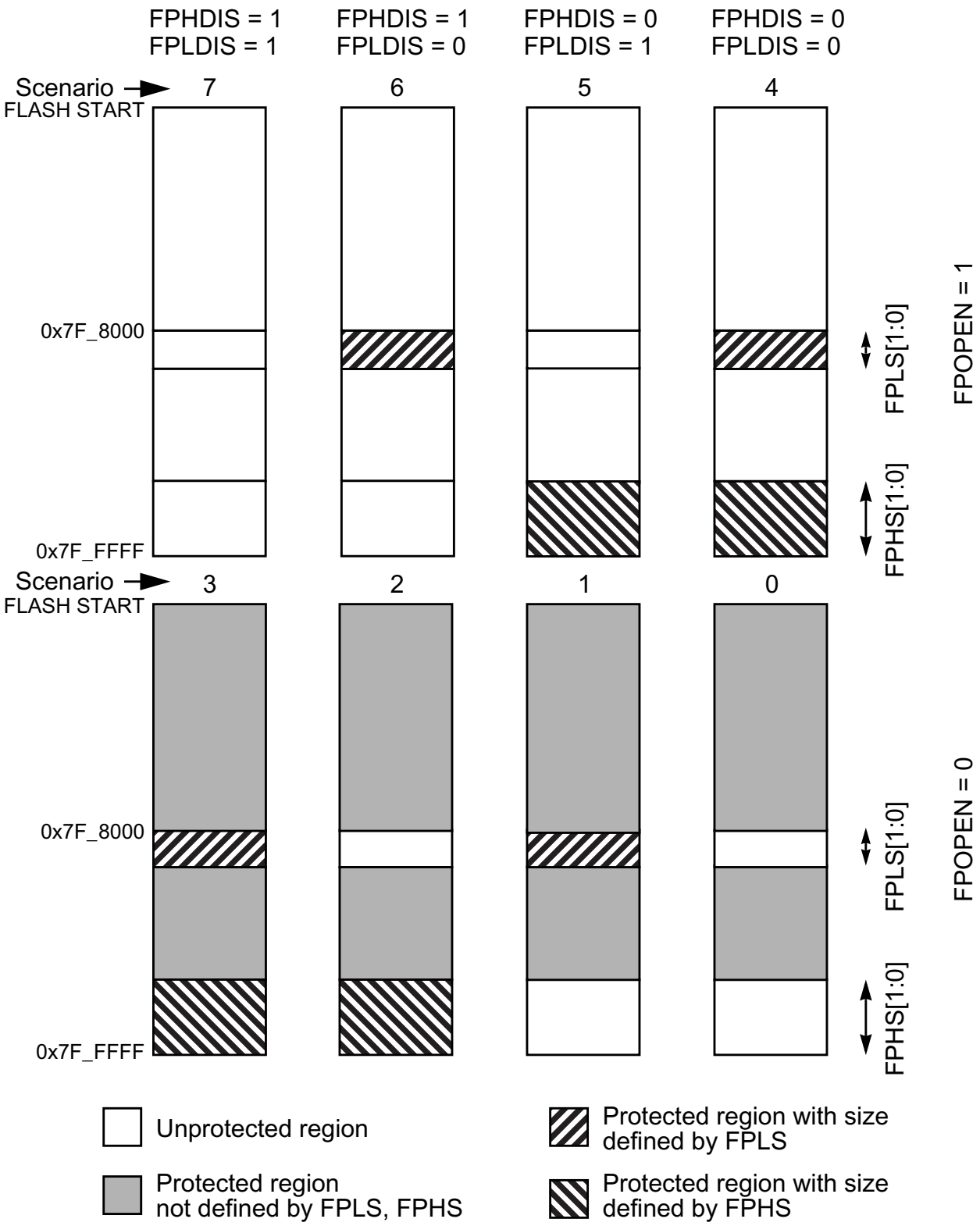



**Figure 29-14. P-Flash Protection Scenarios**

**Table 29-36. Erase Verify Block Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 000 at command launch |
| | | Set if a Load Data Field command sequence is currently active |
| | | Set if an invalid global address [22:16] is supplied |
| | FPVIOL | None |
| | MGSTAT1 | Set if any errors have been encountered during the read |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the read |
| FERSTAT | EPVIOLIF | None |

### 29.4.2.3 Erase Verify P-Flash Section Command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases. The section to be verified cannot cross a 256 Kbyte boundary in the P-Flash memory space.

**Table 29-37. Erase Verify P-Flash Section Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x03 | Global address [22:16] of a P-Flash block |
| 001 | Global address [15:0] of the first phrase to be verified | |
| 010 | Number of phrases to be verified | |

Upon clearing CCIF to launch the Erase Verify P-Flash Section command, the Memory Controller will verify the selected section of Flash memory is erased. The CCIF flag will set after the Erase Verify P-Flash Section operation has completed.

**Table 29-38. Erase Verify P-Flash Section Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 010 at command launch |
| | | Set if a Load Data Field command sequence is currently active |
| | | Set if command not available in current mode (see Table 29-30) |
| | | Set if an invalid global address [22:0] is supplied |
| | | Set if a misaligned phrase address is supplied (global address [2:0] != 000) |
| | | Set if the requested section crosses a 256 Kbyte boundary |
| | FPVIOL | None |
| | MGSTAT1 | Set if any errors have been encountered during the read |
| | MGSTAT0 | Set if any non-correctable errors have been encountered during the read |

### 29.4.2.20 Disable EEPROM Emulation Command

The Disable EEPROM Emulation command causes the Memory Controller to suspend current EEE activity.

**Table 29-73. Disable EEPROM Emulation Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x14 | Not required |

Upon clearing CCIF to launch the Disable EEPROM Emulation command, the Memory Controller will halt EEE operations at the next convenient point without clearing the EEE tag RAM or tag counter before setting the CCIF flag.

**Table 29-74. Disable EEPROM Emulation Command Error Handling**

| Register | Error Bit | Error Condition |
|---|---|---|
| FSTAT | ACCERR | Set if CCOBIX[2:0] != 000 at command launch |
| | | Set if a Load Data Field command sequence is currently active |
| | | Set if Full Partition D-Flash or Partition D-Flash command not previously run |
| | FPVIOL | None |
| | MGSTAT1 | None |
| | MGSTAT0 | None |
| FERSTAT | EPVIOLIF | None |

### 29.4.2.21 EEPROM Emulation Query Command

The EEPROM Emulation Query command returns EEE partition and status variables.

**Table 29-75. EEPROM Emulation Query Command FCCOB Requirements**

| CCOBIX[2:0] | FCCOB Parameters | |
|---|---|---|
| 000 | 0x15 | Not required |
| 001 | Return DFPART | |
| 010 | Return ERPART | |
| 011 | Return ECOUNT(1) | |
| 100 | Return Dead Sector Count | Return Ready Sector Count |

1. Indicates sector erase count

Upon clearing CCIF to launch the EEPROM Emulation Query command, the CCIF flag will set after the EEE partition and status variables are stored in the FCCOBIX register.If the Emulation Query command is executed prior to partitioning (Partition D-Flash Command Section 29.4.2.15), the following reset values are returned: DFPART = 0x_FFFF, ERPART = 0x_FFFF, ECOUNT = 0x_FFFF, Dead Sector Count = 0x_00, Ready Sector Count = 0x_00.

## 0x00D0–0x00D7 Asynchronous Serial Interface (SCI1) Map (continued)

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00D5 | SCI1SR2 | R | AMAP | 0 | 0 | TXPOL | RXPOL | BRK13 | TXDIR | RAF |
| | | W | | | | | | | | |
| 0x00D6 | SCI1DRH | R | R8 | T8 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x00D7 | SCI1DRL | R | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| | | W | T7 | T6 | T5 | T4 | T3 | T2 | T1 | T0 |

1. Those registers are accessible if the AMAP bit in the SCI1SR2 register is set to zero
2. Those registers are accessible if the AMAP bit in the SCI1SR2 register is set to one

## 0x00D8–0x00DF Serial Peripheral Interface (SPI0) Map

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00D8 | SPI0CR1 | R | SPIE | SPE | SPTIE | MSTR | CPOL | CPHA | SSOE | LSBFE |
| | | W | | | | | | | | |
| 0x00D9 | SPI0CR2 | R | 0 | XFRW | 0 | MODFEN | BIDIROE | 0 | SPISWAI | SPC0 |
| | | W | | | | | | | | |
| 0x00DA | SPI0BR | R | 0 | SPPR2 | SPPR1 | SPPR0 | 0 | SPR2 | SPR1 | SPR0 |
| | | W | | | | | | | | |
| 0x00DB | SPI0SR | R | SPIF | 0 | SPTEF | MODF | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x00DC | SPI0DRH | R | R15 | R14 | R13 | R12 | R11 | R10 | R9 | R8 |
| | | W | T15 | T14 | T13 | T12 | T11 | T10 | T9 | T8 |
| 0x00DD | SPI0DRL | R | R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| | | W | T7 | T6 | T5 | T4 | T3 | T2 | T1 | T0 |
| 0x00DE | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |
| 0x00DF | Reserved | R | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | W | | | | | | | | |

## 0x00E0–0x00E7 Inter IC Bus (IIC0) Map

| Address | Name | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00E0 | IBAD | R | ADR7 | ADR6 | ADR5 | ADR4 | ADR3 | ADR2 | ADR1 | 0 |
| | | W | | | | | | | | |
| 0x00E1 | IBFD | R | IBC7 | IBC6 | IBC5 | IBC4 | IBC3 | IBC2 | IBC1 | IBC0 |
| | | W | | | | | | | | |
| 0x00E2 | IBCR | R | IBEN | IBIE | MS/$\overline{SL}$ | TX/$\overline{RX}$ | TXAK | 0 | 0 | IBSWAI |
| | | W | | | | | | RSTA | | |
| 0x00E3 | IBSR | R | TCF | IAAS | IBB | IBAL | 0 | SRW | IBIF | RXAK |
| | | W | | | | | | | | |
| 0x00E4 | IBDR | R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D 0 |
| | | W | | | | | | | | |