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	91
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 16x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	112-LQFP
Supplier Device Package	112-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912xet256j2calr

Unimplemented RAM pages are mapped externally in expanded modes. Accessing unimplemented RAM pages in single chip modes causes an illegal address reset if the MPU is not configured to flag an MPU protection error in that range.

Accessing unimplemented FLASH pages in single chip modes causes an illegal address reset if the MPU is not configured to flag an MPU protection error in that range.

The PARTID value should be referenced regarding the specific memory map for any given device. For devices sharing the same part ID, the memory regions which are implemented on the larger device but not supported on the smaller device are implemented but untested on that smaller device. These regions do not appear as unimplemented in the memory map and do not result in an illegal address reset if erroneously accessed.

Table 1-2. Unimplemented Range Mapping to Part ID

Part ID	RAM_LOW	EE_LOW	Flash Blocks	Registers
0xCC8x	0x0F_0000	0x13_F000	B3, B2, B1S, B1N, B0	2K
0xCC9x	0x0F_0000	0x13_F000	B3, B2, B1S, B1N, B0	2K
0xC48x	0x0F_8000	0x13_F000	B1N, B1S, B0	2K
0xC08x	0x0F_C000	0x13_F000	B1S, B0(128K)	2K

From the above the following examples can be derived.

The 9S12XEP768 is currently only available as a 9S12XEP100 die, thus the unimplemented FLASH pages are those of the 9S12XEP100 device map.

The 9S12XEQ384, 9S12XEG384, 9S12XES384 are currently only available as a 9S12XEQ512 die, thus the unimplemented FLASH pages are those of the 9S12XEQ512 device map.

The 9S12XEG128 is currently only available as a 9S12XET256 die, thus the unimplemented FLASH pages are those of the 9S12XET256 device map.

The range between 0x10_0000 and 0x13_FFFF is mapped to EEPROM resources. The actual EEPROM and dataflash block sizes are listed in Table 1-4. Within EEPROM resource range an address range exists which is neither used by EEPROM resources nor remapped to external resources via chip selects (see the FTM/MMC descriptions for details). These ranges do not constitute unimplemented areas.

Accessing reserved registers within the 2K register space does not generate an illegal address reset.

The fixed 8K RAM default location in the global map is 0x0F_E000- 0x0F_FFFF. This is subject to remapping when configuring the local address map for a larger RAM access range.

The ECLKCTL register is used to control the availability of the free-running clocks and the free-running clock divider.

Table 2-16. ECLKCTL Register Field Descriptions

Field	Description
7 NECLK	**No ECLK**—Disable ECLK output This bit controls the availability of a free-running clock on the ECLK pin. Clock output is always active in emulation modes and if enabled in all other operating modes. 1 ECLK disabled 0 ECLK enabled
6 NCLKX2	**No ECLKX2**—Disable ECLKX2 output This bit controls the availability of a free-running clock on the ECLKX2 pin. This clock has a fixed rate of twice the internal Bus Clock. Clock output is always active in emulation modes and if enabled in all other operating modes. 1 ECLKX2 disabled 0 ECLKX2 enabled
5 DIV16	**Free-running ECLK predivider**—Divide by 16 This bit enables a divide-by-16 stage on the selected EDIV rate. 1 Divider enabled: ECLK rate = EDIV rate divided by 16 0 Divider disabled: ECLK rate = EDIV rate
4-0 EDIV	**Free-running ECLK Divider**—Configure ECLK rate These bits determine the rate of the free-running clock on the ECLK pin. Divider is always disabled in emulation modes and active as programmed in all other operating modes. 00000 ECLK rate = Bus Clock rate 00001 ECLK rate = Bus Clock rate divided by 2 00010 ECLK rate = Bus Clock rate divided by 3, ... 11111 ECLK rate = Bus Clock rate divided by 32

2.3.16 PIM Reserved Register

Address 0x001D (PRR) Access: User read(1)

	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 2-14. PIM Reserved Register

1. Read: Always reads 0x00
   Write: Unimplemented

Table 2-49. PTH Register Field Descriptions (continued)

Field	Description
1 PTH	**Port H general purpose input/output data**—Data Register Port H pin 1 is associated with the TXD signal of the SCI6 module and the MOSI signal of the routed SPI1. The routed SPI1 function takes precedence over the SCI6 and the general purpose I/O function if the routed SPI1 module is enabled. The SCI6 function takes precedence over the general purpose I/O function if the SCI6 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. 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.
0 PTH	**Port H general purpose input/output data**—Data Register Port H pin 0 is associated with the RXD signal of the SCI6 module and the MISO signal of the routed SPI1. The routed SPI1 function takes precedence over the SCI6 and the general purpose I/O function if the routed SPI1 module is enabled. The SCI6 function takes precedence over the general purpose I/O function if the SCI6 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. 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.

2.3.54 Port H Input Register (PTIH)

Address 0x0261 Access: User read(1)

	7	6	5	4	3	2	1	0
R	PTIH7	PTIH6	PTIH5	PTIH4	PTIH3	PTIH2	PTIH1	PTIH0
W								
Reset	u	u	u	u	u	u	u	u

= Unimplemented or Reserved u = Unaffected by reset

Figure 2-52. Port H Input Register (PTIH)

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

Table 2-50. PTIH Register Field Descriptions

Field	Description
7-0 PTIH	**Port H input data**— This register always reads back the buffered state of the associated pins. This can also be used to detect overload or short circuit conditions on output pins.

2.3.55 Port H Data Direction Register (DDRH)

Address 0x0262 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	DDRH7	DDRH6	DDRH5	DDRH4	DDRH3	DDRH2	DDRH1	DDRH0
Reset	0	0	0	0	0	0	0	0

Figure 2-53. Port H Data Direction Register (DDRH)

3.1.1 Terminology

Table 3-2. Acronyms and Abbreviations

Logic level “1”	Voltage that corresponds to Boolean true state
Logic level “0”	Voltage that corresponds to Boolean false state
0x	Represents hexadecimal number
x	Represents logic level ’don’t care’
byte	8-bit data
word	16-bit data
local address	based on the 64 KBytes Memory Space (16-bit address)
global address	based on the 8 MBytes Memory Space (23-bit address)
Aligned address	Address on even boundary
Mis-aligned address	Address on odd boundary
Bus Clock	System Clock. Refer to CRG Block Guide.
expanded modes	Normal Expanded Mode Emulation Single-Chip Mode Emulation Expanded Mode Special Test Mode
single-chip modes	Normal Single-Chip Mode Special Single-Chip Mode
emulation modes	Emulation Single-Chip Mode Emulation Expanded Mode
normal modes	Normal Single-Chip Mode Normal Expanded Mode
special modes	Special Single-Chip Mode Special Test Mode
NS	Normal Single-Chip Mode
SS	Special Single-Chip Mode
NX	Normal Expanded Mode
ES	Emulation Single-Chip Mode
EX	Emulation Expanded Mode
ST	Special Test Mode
Unimplemented areas	Areas which are accessible by the pages (RPAGE,PPAGE,EPAGE) and not implemented
External Space	Area which is accessible in the global address range 14_0000 to 3F_FFFF
external resource	Resources (Emulator, Application) connected to the MCU via the external bus on expanded modes (Unimplemented areas and External Space)
PRR	Port Replacement Registers
PRU	Port Replacement Unit located on the emulator side
MCU	MicroController Unit
NVM	Non-volatile Memory; Flash EEPROM or ROM

3.1.2 Features

The main features of this block are:

- Paging capability to support a global 8 Mbytes memory address space
- Bus arbitration between the masters CPU, BDM and XGATE

16-bit misaligned reads and writes are generally not allowed. If attempted by BDM hardware command, the BDM will ignore the least significant bit of the address and will assume an even address from the remaining bits.

For devices with external bus:

The following cycle count information is only valid when the external wait function is not used (see wait bit of EBI sub-block). During an external wait the BDM can not steal a cycle. Hence be careful with the external wait function if the BDM serial interface is much faster than the bus, because of the BDM soft-reset after time-out (see Section 7.4.11, “Serial Communication Time Out”).

For hardware data read commands, the external host must wait at least 150 bus clock cycles after sending the address before attempting to obtain the read data. This is to be certain that valid data is available in the BDM shift register, ready to be shifted out. For hardware write commands, the external host must wait 150 bus clock cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed. The 150 bus clock cycle delay in both cases includes the maximum 128 cycle delay that can be incurred as the BDM waits for a free cycle before stealing a cycle.

For firmware read commands, the external host should wait at least 48 bus clock cycles after sending the command opcode and before attempting to obtain the read data. This includes the potential of extra cycles when the access is external and stretched (+1 to maximum +7 cycles) or to registers of the PRU (port replacement unit) in emulation modes (if modes available). The 48 cycle wait allows enough time for the requested data to be made available in the BDM shift register, ready to be shifted out.

NOTE

This timing has increased from previous BDM modules due to the new capability in which the BDM serial interface can potentially run faster than the bus. On previous BDM modules this extra time could be hidden within the serial time.

For firmware write commands, the external host must wait 36 bus clock cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed.

The external host should wait at least for 76 bus clock cycles after a TRACE1 or GO command before starting any new serial command. This is to allow the CPU to exit gracefully from the standard BDM firmware lookup table and resume execution of the user code. Disturbing the BDM shift register prematurely may adversely affect the exit from the standard BDM firmware lookup table.

NOTE

If the bus rate of the target processor is unknown or could be changing or the external wait function is used, it is recommended that the ACK (acknowledge function) is used to indicate when an operation is complete. When using ACK, the delay times are automated.

Figure 7-7 represents the BDM command structure. The command blocks illustrate a series of eight bit times starting with a falling edge. The bar across the top of the blocks indicates that the BKGD line idles in the high state. The time for an 8-bit command is 8×16 target clock cycles.[1]

```
                        MOVW  2,X+, 2,Y+
                        MOVW  2,X+, 2,Y+
                        MOVW  2,X+, 2,Y+
                        CPX   #XGATE_CODE_FLASH_END
                        BLS   COPY_XGATE_CODE_LOOP

                        ;#########################################
                        ;#               START XGATE             #
                        ;#########################################
START_XGATE             MOVW  #XGMCTL_ENABLE, XGMCTL          ;enable XGATE
                        BRA   *

                        ;#########################################
                        ;#     DUMMY INTERRUPT SERVICE ROUTINE     #
                        ;#########################################
DUMMY_ISR               RTI

                        CPU   XGATE
                        ;#########################################
                        ;#                XGATE DATA               #
                        ;#########################################
                        ALIGN 1
XGATE_DATA_FLASH        EQU   *
XGATE_DATA_SCI          EQU   *-XGATE_DATA_FLASH
                        DW    SCI_REGS                        ;pointer to SCI register space
XGATE_DATA_IDX          EQU   *-XGATE_DATA_FLASH
                        DB    XGATE_DATA_MSG                  ;string pointer
XGATE_DATA_MSG          EQU   *-XGATE_DATA_FLASH
                        FCC   "Hello World!                   ;ASCII string
                        DB    $0D                             ;CR

                        ;#########################################
                        ;#               XGATE CODE              #
                        ;#########################################
                        ALIGN 1
XGATE_CODE_FLASH        LDW   R2,(R1,#XGATE_DATA_SCI)         ;SCI -> R2
                        LDB   R3,(R1,#XGATE_DATA_IDX)         ;msg -> R3
                        LDB   R4,(R1,R3+)                     ;curr. char -> R4
                        STB   R3,(R1,#XGATE_DATA_IDX)         ;R3 -> idx
                        LDB   R0,(R2,#(SCISR1-SCI_REGS))      ;initiate SCI transmit
                        STB   R4,(R2,#(SCIDRL-SCI_REGS))      ;initiate SCI transmit
                        CMPL  R4,#$0D
                        BEQ   XGATE_CODE_DONE
                        RTS
XGATE_CODE_DONE         LDL   R4,#$00                         ;disable SCI interrupts
                        STB   R4,(R2,#(SCICR2-SCI_REGS))
                        LDL   R3,#XGATE_DATA_MSG;reset R3
                        STB   R3,(R1,#XGATE_DATA_IDX)
XGATE_CODE_FLASH_END    RTS
XGATE_DUMMY_ISR_XG      EQU   (XGATE_CODE_FLASH_END-XGATE_CODE_FLASH)+XGATE_CODE_XG
```

10.9.3 Stack Support

To simplify the implementation of a program stack the XGATE can be configured to set RISC core register R7 to the beginning of a stack region before executing a thread. Two separate stack regions can be defined: One for threads of priority level 7 to 4 (refer to Section 10.3.1.5, “XGATE Initial Stack Pointer for

13.1.2 Modes of Operation

13.1.2.1 Conversion Modes

There is software programmable selection between performing **single** or **continuous conversion** on a **single channel** or **multiple channels**.

13.1.2.2 MCU Operating Modes

- **Stop Mode**
  - **ICLKSTP=0 (in ATDCTL2 register)**
    Entering Stop Mode aborts any conversion sequence in progress and if a sequence was aborted restarts it after exiting stop mode. This has the same effect/consequences as starting a conversion sequence with write to ATDCTL5. So after exiting from stop mode with a previously aborted sequence all flags are cleared etc.
  - **ICLKSTP=1 (in ATDCTL2 register)**
    A/D conversion sequence seamless continues in Stop Mode based on the internally generated clock ICLK as ATD clock. For conversions during transition from Run to Stop Mode or vice versa the result is not written to the results register, no CCF flag is set and no compare is done. When converting in Stop Mode (ICLKSTP=1) an ATD Stop Recovery time $t_{ATDSTPRCV}$ is required to switch back to bus clock based ATDCLK when leaving Stop Mode. Do not access ATD registers during this time.
- **Wait Mode**
  ADC12B16C behaves same in Run and Wait Mode. For reduced power consumption continuos conversions should be aborted before entering Wait mode.
- **Freeze Mode**
  In Freeze Mode the ADC12B16C will either continue or finish or stop converting according to the FRZ1 and FRZ0 bits. This is useful for debugging and emulation.

13.4.2.2 General-Purpose Digital Port Operation

The input channel pins can be multiplexed between analog and digital data. As analog inputs, they are multiplexed and sampled as analog channels to the A/D converter. The analog/digital multiplex operation is performed in the input pads. The input pad is always connected to the analog input channels of the ADC12B16C. The input pad signal is buffered to the digital port registers. This buffer can be turned on or off with the ATDDIEN register. This is important so that the buffer does not draw excess current when analog potentials are presented at its input.

13.5 Resets

At reset the ADC12B16C is in a power down state. The reset state of each individual bit is listed within the Register Description section (see Section 13.3.2, “Register Descriptions”) which details the registers and their bit-field.

13.6 Interrupts

The interrupts requested by the ADC12B16C are listed in Table 13-24. Refer to MCU specification for related vector address and priority.

Table 13-24. ATD Interrupt Vectors

Interrupt Source	CCR Mask	Local Enable
Sequence Complete Interrupt	I bit	ASCIE in ATDCTL2
Compare Interrupt	I bit	ACMPIE in ATDCTL2

See Section 13.3.2, “Register Descriptions” for further details.

Chapter 15
Inter-Integrated Circuit (IICV3) Block Description

Table 15-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V01.03	28 Jul 2006	15.7.1.7/15-601	- Update flow-chart of interrupt routine for 10-bit address
V01.04	17 Nov 2006	15.3.1.2/15-582	- Revise Table1-5
V01.05	14 Aug 2007	15.3.1.1/15-581	- Backward compatible for IBAD bit name

15.1 Introduction

The inter-IC bus (IIC) is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange between devices. Being a two-wire device, the IIC bus minimizes the need for large numbers of connections between devices, and eliminates the need for an address decoder.

This bus is suitable for applications requiring occasional communications over a short distance between a number of devices. It also provides flexibility, allowing additional devices to be connected to the bus for further expansion and system development.

The interface is designed to operate up to 100 kbps with maximum bus loading and timing. The device is capable of operating at higher baud rates, up to a maximum of clock/20, with reduced bus loading. The maximum communication length and the number of devices that can be connected are limited by a maximum bus capacitance of 400 pF.

15.1.1 Features

The IIC module has the following key features:

- Compatible with I2C bus standard
- Multi-master operation
- Software programmable for one of 256 different serial clock frequencies
- Software selectable acknowledge bit
- Interrupt driven byte-by-byte data transfer
- Arbitration lost interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- Start and stop signal generation/detection
- Repeated start signal generation
- Acknowledge bit generation/detection
- Bus busy detection

Chapter 16
Freescale's Scalable Controller Area Network (S12MSCANV3)

Table 16-1. Revision History

Revision Number	Revision Date	Sections Affected	Description of Changes
V03.13	03 Mar 2011	Figure 16-4 Table 16-3	• Corrected CANE write restrictions • Removed footnote from RXFRM bit
V03.14	12 Nov 2012	Table 16-11	• Corrected RxWRN and TxWRN threshold values
V03.15	12 Jan 2013	Table 16-3 Table 16-26 Figure 16-37 16.1/16-605 16.3.2.15/16-626	• Updated TIME bit description • Added register names to buffer map • Updated TSRH and TSRL read conditions • Updated introduction • Updated CANTXERR and CANRXERR register notes

16.1 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 S12, S12X and S12Z microcontroller families.

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.

NOTE

Changing the PWM output mode from left aligned to center aligned output (or vice versa) while channels are operating can cause irregularities in the PWM output. It is recommended to program the output mode before enabling the PWM channel.

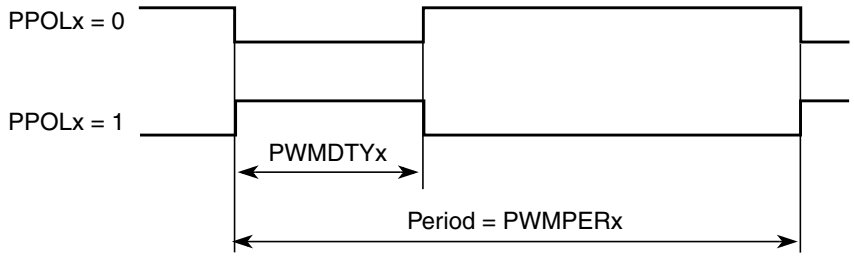

Figure 19-20. PWM Left Aligned Output Waveform

To calculate the output frequency in left aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB) and divide it by the value in the period register for that channel.

- PWMx Frequency = Clock (A, B, SA, or SB) / PWMPERx
- PWMx Duty Cycle (high time as a% of period):
  - Polarity = 0 (PPOLx = 0)
- Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
  - Polarity = 1 (PPOLx = 1)

Duty Cycle = [PWMDTYx / PWMPERx] * 100%

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

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

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10 MHz/4 = 2.5 MHz

PWMx Period = 400 ns

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

The output waveform generated is shown in Figure 19-21.

Table 24-45. Erase All Blocks Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x08	Not required

Upon clearing CCIF to launch the Erase All Blocks command, the Memory Controller will erase the entire 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. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag will set after the Erase All Blocks operation has completed.

Table 24-46. Erase All Blocks 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 24-30)
	FPVIOL	Set if any area of the P-Flash memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation(1)
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation[1]
FERSTAT	EPVIOLIF	Set if any area of the buffer RAM EEE partition is protected

1. As found in the memory map for FTM256K2.

24.4.2.8 Erase P-Flash Block Command

The Erase P-Flash Block operation will erase all addresses in a P-Flash block.

Table 24-47. Erase P-Flash Block Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x09	Global address [22:16] to identify P-Flash block
001	Global address [15:0] in P-Flash block to be erased	

Upon clearing CCIF to launch the Erase P-Flash Block command, the Memory Controller will erase the selected P-Flash block and verify that it is erased. The CCIF flag will set after the Erase P-Flash Block operation has completed.

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 26-59. Set User Margin Level 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 26-30)
		Set if an invalid global address [22:16] is supplied(1)
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

1. As defined by the memory map for FTM512K3.

NOTE

User margin levels can be used to check that Flash memory contents have adequate margin for normal level read operations. If unexpected results are encountered when checking Flash memory contents at user margin levels, a potential loss of information has been detected.

26.4.2.14 Set Field Margin Level Command

The Set Field Margin Level command, valid in special modes only, causes the Memory Controller to set the margin level specified for future read operations of a specific P-Flash or D-Flash block.

Table 26-60. Set Field Margin Level Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0E	Global address [22:16] to identify the Flash block
001	Margin level setting	

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

Valid margin level settings for the Set Field Margin Level command are defined in Table 26-61.

Table 26-61. Valid Set Field Margin Level Settings

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

Table 28-59. Set User Margin Level 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 28-30)
		Set if an invalid global address [22:16] is supplied(1)
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

1. As defined by the memory map for FTM1024K5.

NOTE

User margin levels can be used to check that Flash memory contents have adequate margin for normal level read operations. If unexpected results are encountered when checking Flash memory contents at user margin levels, a potential loss of information has been detected.

28.4.2.14 Set Field Margin Level Command

The Set Field Margin Level command, valid in special modes only, causes the Memory Controller to set the margin level specified for future read operations of a specific P-Flash or D-Flash block.

Table 28-60. Set Field Margin Level Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0E	Global address [22:16] to identify the Flash block
001	Margin level setting	

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

Valid margin level settings for the Set Field Margin Level command are defined in Table 28-61.

Table 28-61. Valid Set Field Margin Level Settings

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

Table 28-63. Full Partition D-Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0F	Not required
001	Number of 256 byte sectors for the D-Flash user partition (DFPART)	
010	Number of 256 byte sectors for buffer RAM EEE partition (ERPART)	

Upon clearing CCIF to launch the Full Partition D-Flash command, the following actions are taken to define a partition within the D-Flash block for direct access (DFPART) and a partition within the buffer RAM for EEE use (ERPART):

- Validate the DFPART and ERPART values provided:
  - DFPART <= 128 (maximum number of 256 byte sectors in D-Flash block)
  - ERPART <= 16 (maximum number of 256 byte sectors in buffer RAM)
  - If ERPART > 0, 128 - DFPART >= 12 (minimum number of 256 byte sectors in the D-Flash block required to support EEE)
  - If ERPART > 0, ((128-DFPART)/ERPART) >= 8 (minimum ratio of D-Flash EEE space to buffer RAM EEE space to support EEE)
- Erase the D-Flash block and the EEE nonvolatile information register
- Program DFPART to the EEE nonvolatile information register at global address 0x12_0000 (see Table 28-7)
- Program a duplicate DFPART to the EEE nonvolatile information register at global address 0x12_0002 (see Table 28-7)
- Program ERPART to the EEE nonvolatile information register at global address 0x12_0004 (see Table 28-7)
- Program a duplicate ERPART to the EEE nonvolatile information register at global address 0x12_0006 (see Table 28-7)

The D-Flash user partition will start at global address 0x10_0000. The buffer RAM EEE partition will end at global address 0x13_FFFF. After the Full Partition D-Flash operation has completed, the CCIF flag will set.

Running the Full Partition D-Flash command a second time will result in the previous partition values and the entire D-Flash memory being erased. The data value written corresponds to the number of 256 byte sectors allocated for either direct D-Flash access (DFPART) or buffer RAM EEE access (ERPART).

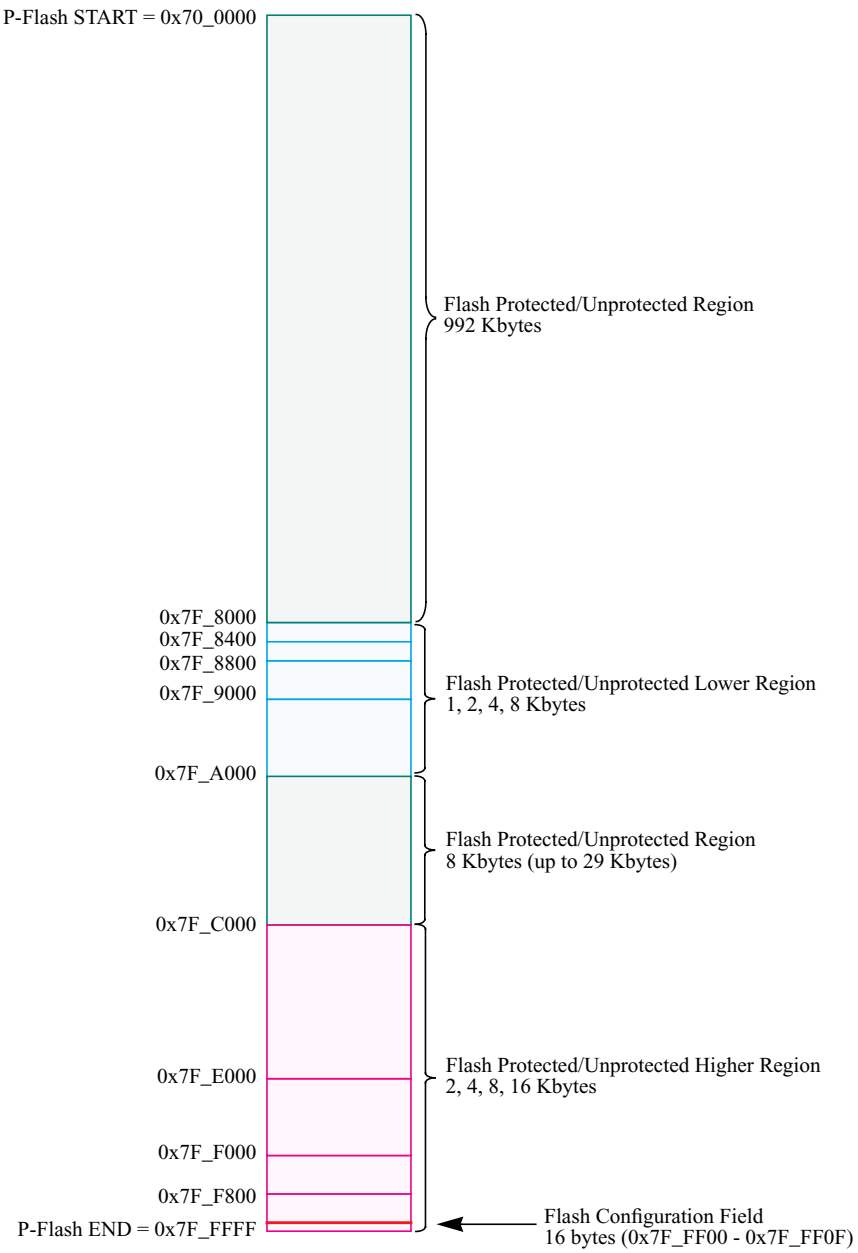

Figure 29-2. P-Flash Memory Map

Table A-13. Run and Wait Current Characteristics

Conditions are shown in Table A-4 unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
Run supply current (No external load, Peripheral Configuration see Table A-11.)							
1	P	Peripheral Set[1] f_{osc}=4MHz, f_{bus}=50MHz	I_{DD35}	—	—	100	mA
Run supply current (No external load, Peripheral Configuration see Table A-10.)							
2		Peripheral Set[(1)] Devices S12XEP100, S12XEP768	I_{DD35}				mA
	C	f_{osc}=4MHz, f_{bus}=50MHz		—	84	—	
	T	f_{osc}=4MHz, f_{bus}=20MHz		—	43	—	
	T	f_{osc}=4MHz, f_{bus}=8MHz		—	24	—	
2a		Peripheral Set[1] All other devices					
	T	f_{osc}=4MHz, f_{bus}=50MHz		—	72	—	mA
3		Peripheral Set[(2)]					mA
	T	f_{osc}=4MHz, f_{bus}=50MHz		—	63	—	
	T	f_{osc}=4MHz, f_{bus}=20MHz		—	35	—	
	T	f_{osc}=4MHz, f_{bus}=8MHz		—	21	—	
4		Peripheral Set[(3)]					mA
	T	f_{osc}=4MHz, f_{bus}=50MHz		—	62	—	
	T	f_{osc}=4MHz, f_{bus}=20MHz		—	34	—	
	T	f_{osc}=4MHz, f_{bus}=8MHz		—	21	—	
5		Peripheral Set[(4)]					mA
	T	f_{osc}=4MHz, f_{bus}=50MHz		—	60	—	
	T	f_{osc}=4MHz, f_{bus}=20MHz		—	33	—	
	T	f_{osc}=4MHz, f_{bus}=8MHz		—	20	—	
6		Peripheral Set[(5)]					mA
	T	f_{osc}=4MHz, f_{bus}=50MHz		—	59	—	
	T	f_{osc}=4MHz, f_{bus}=20MHz		—	33	—	
	T	f_{osc}=4MHz, f_{bus}=8MHz		—	20	—	
7		Peripheral Set[(6)]					mA
	T	f_{osc}=4MHz, f_{bus}=50MHz		—	57	—	
	T	f_{osc}=4MHz, f_{bus}=20MHz		—	33	—	
	T	f_{osc}=4MHz, f_{bus}=8MHz		—	20	—	
Wait supply current							
8	C	Peripheral Set[1] ,PLL on XGATE executing code from RAM	I_{DDW}	—	—	85	mA
9		Peripheral Set[2]					
	T	f_{osc}=4MHz, f_{bus}=50MHz		—	50	—	
	T	f_{osc}=4MHz, f_{bus}=8MHz		—	12	—	
10	P	All modules disabled, RTI enabled, PLL off		—	—	10	

1. The following peripherals are on: ATD0/ATD1/ECT/IIC1/PWM/SPI0-SPI2/SCI0-SCI7/CAN0-CAN4/XGATE
2. The following peripherals are on: ATD0/ATD1/ECT/IIC1/PWM/SPI0-SPI2/SCI0-SCI7/CAN0-CAN4
3. The following peripherals are on: ATD0/ATD1/ECT/IIC1/PWM/SPI0-SPI2/SCI0-SCI7
4. The following peripherals are on: ATD0/ATD1/ECT/IIC1/PWM/SPI0-SPI2
5. The following peripherals are on: ATD0/ATD1/ECT/IIC1/PWM
6. The following peripherals are on: ATD0/ATD1/ECT/IIC1

0x01C0–0x01FF MSCAN (CAN2) Map (Sheet 3 of 3)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x01DF	CAN2IDMR7	R W	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
0x01E0– 0x01EF	CAN2RXFG	R	FOREGROUND RECEIVE BUFFER (See Detailed MSCAN Foreground Receive and Transmit Buffer Layout)							
		W								
0x01F0– 0x01FF	CAN2TXFG	R	FOREGROUND TRANSMIT BUFFER (See Detailed MSCAN Foreground Receive and Transmit Buffer Layout)							
		W								

0x0240–0x027F Port Integration Module (PIM) Map 5 of 6 (continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0260	PTH	R W	PTH7	PTH6	PTH5	PTH4	PTH3	PTH2	PTH1	PTH0
0x0261	PTIH	R W	PTIH7	PTIH6	PTIH5	PTIH4	PTIH3	PTIH2	PTIH1	PTIH0
0x0262	DDRH	R W	DDRH7	DDRH7	DDRH5	DDRH4	DDRH3	DDRH2	DDRH1	DDRH0
0x0263	RDRH	R W	RDRH7	RDRH6	RDRH5	RDRH4	RDRH3	RDRH2	RDRH1	RDRH0
0x0264	PERH	R W	PERH7	PERH6	PERH5	PERH4	PERH3	PERH2	PERH1	PERH0
0x0265	PPSH	R W	PPSH7	PPSH6	PPSH5	PPSH4	PPSH3	PPSH2	PPSH1	PPSH0
0x0266	PIEH	R W	PIEH7	PIEH6	PIEH5	PIEH4	PIEH3	PIEH2	PIEH1	PIEH0
0x0267	PIFH	R W	PIFH7	PIFH6	PIFH5	PIFH4	PIFH3	PIFH2	PIFH1	PIFH0
0x0268	PTJ	R W	PTJ7	PTJ6	PTJ5	PTJ4	PTJ3	PTJ2	PTJ1	PTJ0
0x0269	PTIJ	R W	PTIJ7	PTIJ6	PTIJ5	PTIJ4	PTIJ3	PTIJ2	PTIJ1	PTIJ0
0x026A	DDRJ	R W	DDRJ7	DDRJ7	DDRJ5	DDRJ4	DDRJ3	DDRJ2	DDRJ1	DDRJ0
0x026B	RDRJ	R W	RDRJ7	RDRJ6	RDRJ5	RDRJ4	RDRJ3	RDRJ2	RDRJ1	RDRJ0
0x026C	PERJ	R W	PERJ7	PERJ6	PERJ5	PERJ4	PERJ3	PERJ2	PERJ1	PERJ0
0x026D	PPSJ	R W	PPSJ7	PPSJ6	PPSJ5	PPSJ4	PPSJ3	PPSJ2	PPSJ1	PPSJ0
0x026E	PIEJ	R W	PIEJ7	PIEJ6	PIEJ5	PIEJ4	PIEJ3	PIEJ2	PIEJ1	PIEJ0
0x026f	PIFJ	R W	PIFJ7	PIFJ6	PIFJ5	PIFJ4	PIFJ3	PIFJ2	PIFJ1	PIFJ0
0x0270	PT0AD0	R W	PT0AD07	PT0AD06	PT0AD05	PT0AD04	PT0AD03	PT0AD02	PT0AD01	PT0AD00
0x0271	PT1AD0	R W	PT1AD07	PT1AD06	PT1AD05	PT1AD04	PT1AD03	PT1AD02	PT1AD01	PT1AD00
0x0272	DDR0AD0	R W	DDR0AD07	DDR0AD06	DDR0AD05	DDR0AD04	DDR0AD03	DDR0AD02	DDR0AD01	DDR0AD00
0x0273	DDR1AD0	R W	DDR1AD07	DDR1AD06	DDR1AD05	DDR1AD04	DDR1AD03	DDR1AD02	DDR1AD01	DDR1AD00
0x0274	RDR0AD0	R W	RDR0AD07	RDR0AD06	RDR0AD05	RDR0AD04	RDR0AD03	RDR0AD02	RDR0AD01	RDR0AD00
0x0275	RDR1AD0	R W	RDR1AD07	RDR1AD06	RDR1AD05	RDR1AD04	RDR1AD03	RDR1AD02	RDR1AD01	RDR1AD00
0x0276	PER0AD0	R W	PER0AD07	PER0AD06	PER0AD05	PER0AD04	PER0AD03	PER0AD02	PER0AD01	PER0AD00
0x0277	PER1AD0	R W	PER1AD07	PER1AD06	PER1AD05	PER1AD04	PER1AD03	PER1AD02	PER1AD01	PER1AD00