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	768KB (768K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	48K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 24x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912xep768j5cagr

2.3.11 Port E Data Register (PORTE)

Address 0x0008 (PRR) Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
W								
Altern. Function	XCLKS or ECLKX2	MODB or TAGHI	MODA or RE or TAGLO	ECLK	EROMCTL or LSTRB or LDS	RW or WE	IRQ	XIRQ
Reset	0	0	0	0	0	0	—(2)	—2

= Unimplemented or Reserved

Figure 2-9. Port E Data Register (PORTE)

1. Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes the data source is depending on the data direction value.
   Write: Anytime. In emulation modes, write operations will also be directed to the external bus.
2. These registers are reset to zero. Two bus clock cycles after reset release the register values are updated with the associated pin values.

Table 2-12. PORTE Register Field Descriptions

Field	Description
7-2 PE	**Port E general purpose input/output data**—Data Register Port E bits 7 through 0 are associated with external bus control signals and interrupt inputs. These include mode select (MODB, MODA), E clock, double frequency E clock, Instruction Tagging High and Low (TAGHI, TAGLO), Read/Write (RW), Read Enable and Write Enable (RE, WE), Lower Data Select (LDS). When not used with the alternative functions, Port E pins 7-2 can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. Pins 6 and 5 are inputs with enabled pull-down devices while RESET pin is low. Pins 7 and 3 are inputs with enabled pull-up devices while RESET pin is low.
1 PE	**Port E general purpose input data and interrupt**—Data Register, IRQ input. This pin can be used as general purpose and IRQ input.
0 PE	**Port E general purpose input data and interrupt**—Data Register, XIRQ input. This pin can be used as general purpose and XIRQ input.

3.3.2.4 Direct Page Register (DIRECT)

Address: 0x0011

	7	6	5	4	3	2	1	0
R W	DP15	DP14	DP13	DP12	DP11	DP10	DP9	DP8
Reset	0	0	0	0	0	0	0	0

Figure 3-8. Direct Register (DIRECT)

Read: Anytime

Write: anytime in special modes, one time only in other modes.

This register determines the position of the 256 Byte direct page within the memory map.It is valid for both global and local mapping scheme.

Table 3-10. DIRECT Field Descriptions

Field	Description
7–0 DP[15:8]	**Direct Page Index Bits 15–8** — These bits are used by the CPU when performing accesses using the direct addressing mode. The bits from this register form bits [15:8] of the address (see Figure 3-9).

CAUTION

XGATE write access to this register during an CPU access which makes use of this register could lead to unexpected results.

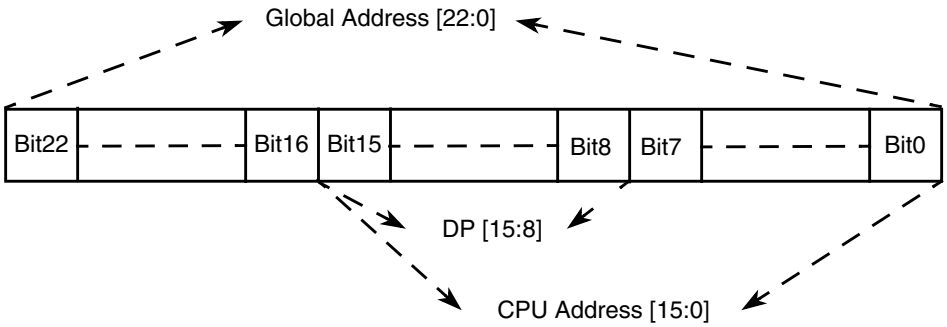


Figure 3-9. DIRECT Address Mapping

Bits [22:16] of the global address will be formed by the GPAGE[6:0] bits in case the CPU executes a global instruction in direct addressing mode or by the appropriate local address to the global address expansion (refer to Section 3.4.2.1.1, “Expansion of the Local Address Map).

Example 3-2. This example demonstrates usage of the Direct Addressing Mode

```
MOVB     #0x80,DIRECT      ;Set DIRECT register to 0x80. Write once only.
                           ;Global data accesses to the range 0xXX_80XX can be direct.
                           ;Logical data accesses to the range 0x80XX are direct.

LDY      <00               ;Load the Y index register from 0x8000 (direct access).
                           ;< operator forces direct access on some assemblers but in
```

5.3 Memory Map and Register Definition

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

5.3.1 Module Memory Map

The registers associated with the XEBI block are shown in Figure 5-2.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0E EBICTL0	R	ITHRS	0	HDBE	ASIZ4	ASIZ3	ASIZ2	ASIZ1	ASIZ0
	W								
0x0F EBICTL1	R	0	EXSTR12	EXSTR11	EXSTR10	0	EXSTR02	EXSTR01	EXSTR00
	W								

= Unimplemented or Reserved

Figure 5-2. XEBI Register Summary

5.3.2 Register Descriptions

The following sub-sections provide a detailed description of each register and the individual register bits.

All control bits can be written anytime, but this may have no effect on the related function in certain operating modes. This allows specific configurations to be set up before changing into the target operating mode.

NOTE

Depending on the operating mode an available function may be enabled, disabled or depend on the control register bit. Reading the register bits will reflect the status of related function only if the current operating mode allows user control. Please refer the individual bit descriptions.

5.3.2.1 External Bus Interface Control Register 0 (EBICTL0)

Module Base +0x000E (PRR)

	7	6	5	4	3	2	1	0
R	ITHRS	0	HDBE	ASIZ4	ASIZ3	ASIZ2	ASIZ1	ASIZ0
W								
Reset	0	0	1	1	1	1	1	1

= Unimplemented or Reserved

Figure 5-3. External Bus Interface Control Register 0 (EBICTL0)

Read: Anytime. In emulation modes, read operations will return the data from the external bus, in all other modes, the data is read from this register.

Write: Anytime. In emulation modes, write operations will also be directed to the external bus.

10.8.1.3 Immediate 3-Bit Wide (IMM3)

Operands for immediate mode instructions are included in the instruction stream and are fetched into the instruction queue along with the rest of the 16 bit instruction. The '#' symbol is used to indicate an immediate addressing mode operand. This address mode is used for semaphore instructions.

Examples:

```
CSEM    #1      ; Unlock semaphore 1
SSEM    #3      ; Lock Semaphore 3
```

10.8.1.4 Immediate 4 Bit Wide (IMM4)

The 4 bit wide immediate addressing mode is supported by all shift instructions.

RD = RD ∗ IMM4

Examples:

```
LSL     R4,#1   ; R4 = R4 << 1; shift register R4 by 1 bit to the left
LSR     R4,#3   ; R4 = R4 >> 3; shift register R4 by 3 bits to the right
```

10.8.1.5 Immediate 8 Bit Wide (IMM8)

The 8 bit wide immediate addressing mode is supported by four major commands (ADD, SUB, LD, CMP).

RD = RD ∗ imm8

Examples:

```
ADDL    R1,#1   ; adds an 8 bit value to register R1
SUBL    R2,#2   ; subtracts an 8 bit value from register R2
LDH     R3,#3   ; loads an 8 bit immediate into the high byte of Register R3
CMPL    R4,#4   ; compares the low byte of register R4 with an immediate value
```

10.8.1.6 Immediate 16 Bit Wide (IMM16)

The 16 bit wide immediate addressing mode is a construct to simplify assembler code. Instructions which offer this mode are translated into two opcodes using the eight bit wide immediate addressing mode.

RD = RD ∗ IMM16

Examples:

```
LDW     R4,#$1234       ; translated to LDL R4,#$34; LDH R4,#$12
ADD     R4,#$5678       ; translated to ADDL R4,#$78; ADDH R4,#$56
```

10.8.1.7 Monadic Addressing (MON)

In this addressing mode only one operand is explicitly given. This operand can either be the source (*f*(RD)), the target (RD = *f*()), or both source and target of the operation (RD = *f*(RD)).

Examples:

```
JAL     R1      ; PC = R1, R1 = PC+2
SIF     R2      ; Trigger IRQ associated with the channel number in R2.L
```

10.8.1.8 Dyadic Addressing (DYA)

In this mode the result of an operation between two registers is stored in one of the registers used as operands.

RD = RD ∗ RS is the general register to register format, with register RD being the first operand and RS the second. RD and RS can be any of the 8 general purpose registers R0 … R7. If R0 is used as the destination register, only the condition code flags are updated. This addressing mode is used only for shift operations with a variable shift value

Examples:

```
LSL     R4,R5    ; R4 = R4 << R5
LSR     R4,R5    ; R4 = R4 >> R5
```

10.8.1.9 Triadic Addressing (TRI)

In this mode the result of an operation between two or three registers is stored into a third one.
RD = RS1 ∗ RS2 is the general format used in the order RD, RS1, RS1. RD, RS1, RS2 can be any of the 8 general purpose registers R0 … R7. If R0 is used as the destination register RD, only the condition code flags are updated. This addressing mode is used for all arithmetic and logical operations.

Examples:

```
ADC     R5,R6,R7          ; R5 = R6 + R7 + Carry
SUB     R5,R6,R7          ; R5 = R6 - R7
```

10.8.1.10 Relative Addressing 9-Bit Wide (REL9)

A 9-bit signed word address offset is included in the instruction word. This addressing mode is used for conditional branch instructions.

Examples:

```
BCC     REL9              ; PC = PC + 2 + (REL9 << 1)
BEQ     REL9              ; PC = PC + 2 + (REL9 << 1)
```

10.8.1.11 Relative Addressing 10-Bit Wide (REL10)

An 10-bit signed word address offset is included in the instruction word. This addressing mode is used for the unconditional branch instruction.

Examples:

```
BRA     REL10             ; PC = PC + 2 + (REL10 << 1)
```

10.8.1.12 Index Register plus Immediate Offset (IDO5)

(RS, #OFFS5) provides an unsigned offset from the base register.

Examples:

```
LDB     R4,(R1,#OFFS5)    ; loads a byte from (R1+OFFS5) into R4
STW     R4,(R1,#OFFS5)    ; stores R4 as a word to (R1+OFFS5)
```

LDB — Load Byte from Memory (Low Byte) — LDB

Operation

M[RB, #OFFS5] ⇒ RD.L; $00 ⇒ RD.H
M[RB, RI] ⇒ RD.L; $00 ⇒ RD.H
M[RB, RI] ⇒ RD.L; $00 ⇒ RD.H; RI+1 ⇒ RI;[1]
RI-1 ⇒ RI; M[RS, RI] ⇒ RD.L; $00 ⇒ RD.H

Loads a byte from memory into the low byte of register RD. The high byte is cleared.

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
LDB RD, (RB, #OFFS5)	IDO5	0	1	0	0	0	RD	RB	OFFS5			Pr
LDB RD, (RS, RI)	IDR	0	1	1	0	0	RD	RB	RI	0	0	Pr
LDB RD, (RS, RI+)	IDR+	0	1	1	0	0	RD	RB	RI	0	1	Pr
LDB RD, (RS, -RI)	-IDR	0	1	1	0	0	RD	RB	RI	1	0	Pr

1. If the same general purpose register is used as index (RI) and destination register (RD), the content of the register will not be incremented after the data move: M[RB, RI] ⇒ RD.L; $00 ⇒ RD.H

PAR Calculate Parity PAR

Operation

Calculates the number of ones in the register RD. The Carry flag will be set if the number is odd, otherwise it will be cleared.

CCR Effects

N	Z	V	C
0	Δ	0	Δ

N: 0; cleared.
Z: Set if RD is $0000; cleared otherwise.
V: 0; cleared.
C: Set if the number of ones in the register RD is odd; cleared otherwise.

Code and CPU Cycles

Source Form	Address Mode	Machine Code														Cycles
PAR, RD	MON	0	0	0	0	0	RD	1	1	1	1	0	1	0	1	P

14.3.2.2 Timer Compare Force Register (CFORC)

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	FOC7	FOC6	FOC5	FOC4	FOC3	FOC2	FOC1	FOC0
Reset	0	0	0	0	0	0	0	0

Figure 14-4. Timer Compare Force Register (CFORC)

Read or write: Anytime but reads will always return 0x0000 (1 state is transient).

All bits reset to zero.

Table 14-3. CFORC Field Descriptions

Field	Description
7:0 FOC[7:0]	**Force Output Compare Action for Channel 7:0** — A write to this register with the corresponding data bit(s) set causes the action which is programmed for output compare “x” to occur immediately. The action taken is the same as if a successful comparison had just taken place with the TCx register except the interrupt flag does not get set. **Note:** A channel 7 event, which can be a counter overflow when TTOV[7] is set or A successful channel 7 output compare overrides any channel 6:0 compares. If a forced output compare on any channel occurs at the same time as the successful output compare, then the forced output compare action will take precedence and the interrupt flag will not get set.

14.3.2.3 Output Compare 7 Mask Register (OC7M)

Module Base + 0x0002

	7	6	5	4	3	2	1	0
R W	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0
Reset	0	0	0	0	0	0	0	0

Figure 14-5. Output Compare 7 Mask Register (OC7M)

Read or write: Anytime

All bits reset to zero.

Table 16-11. CANRFLG Register Field Descriptions (continued)

Field	Description
1 OVRIF	**Overrun Interrupt Flag** — This flag is set when a data overrun condition occurs. If not masked, an error interrupt is pending while this flag is set. 0 No data overrun condition 1 A data overrun detected
0 RXF(2)	**Receive Buffer Full Flag** — RXF is set by the MSCAN when a new message is shifted in the receiver FIFO. This flag indicates whether the shifted buffer is loaded with a correctly received message (matching identifier, matching cyclic redundancy code (CRC) and no other errors detected). After the CPU has read that message from the RxFG buffer in the receiver FIFO, the RXF flag must be cleared to release the buffer. A set RXF flag prohibits the shifting of the next FIFO entry into the foreground buffer (RxFG). If not masked, a receive interrupt is pending while this flag is set. 0 No new message available within the RxFG 1 The receiver FIFO is not empty. A new message is available in the RxFG

1. Redundant Information for the most critical CAN bus status which is "bus-off". This only occurs if the Tx error counter exceeds a number of 255 errors. Bus-off affects the receiver state. As soon as the transmitter leaves its bus-off state the receiver state skips to RxOK too. Refer also to TSTAT[1:0] coding in this register.
2. To ensure data integrity, do not read the receive buffer registers while the RXF flag is cleared. For MCUs with dual CPUs, reading the receive buffer registers while the RXF flag is cleared may result in a CPU fault condition.

16.3.2.6 MSCAN Receiver Interrupt Enable Register (CANRIER)

This register contains the interrupt enable bits for the interrupt flags described in the CANRFLG register.

Module Base + 0x0005 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	WUPIE	CSCIE	RSTATE1	RSTATE0	TSTATE1	TSTATE0	OVRIE	RXFIE
Reset:	0	0	0	0	0	0	0	0

Figure 16-9. MSCAN Receiver Interrupt Enable Register (CANRIER)

1. Read: Anytime
   Write: Anytime when not in initialization mode

NOTE

The CANRIER register is held in the reset state when the initialization mode is active (INITRQ=1 and INITAK=1). This register is writable when not in initialization mode (INITRQ=0 and INITAK=0).

The RSTATE[1:0], TSTATE[1:0] bits are not affected by initialization mode.

18.3.0.6 PIT Time-Out Flag Register (PITTF)

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTF3	PTF2	PTF1	PTF0
W								
Reset	0	0	0	0	0	0	0	0

Figure 18-8. PIT Time-Out Flag Register (PITTF)

Read: Anytime

Write: Anytime (write to clear)

Table 18-7. PITTF Field Descriptions

Field	Description
3:0 PTF[3:0]	**PIT Time-out Flag Bits for Timer Channel 3:0** — PTF is set when the corresponding 16-bit timer modulus down-counter and the selected 8-bit micro timer modulus down-counter have counted to zero. The flag can be cleared by writing a one to the flag bit. Writing a zero has no effect. If flag clearing by writing a one and flag setting happen in the same bus clock cycle, the flag remains set. The flag bits are cleared if the PIT module is disabled or if the corresponding timer channel is disabled. 0 Time-out of the corresponding PIT channel has not yet occurred. 1 Time-out of the corresponding PIT channel has occurred.

18.3.0.7 PIT Micro Timer Load Register 0 to 1 (PITMTLD0–1)

Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	PMTLD7	PMTLD6	PMTLD5	PMTLD4	PMTLD3	PMTLD2	PMTLD1	PMTLD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 18-9. PIT Micro Timer Load Register 0 (PITMTLD0)

Module Base + 0x0007

	7	6	5	4	3	2	1	0
R	PMTLD7	PMTLD6	PMTLD5	PMTLD4	PMTLD3	PMTLD2	PMTLD1	PMTLD0
W								
Reset	0	0	0	0	0	0	0	0

Figure 18-10. PIT Micro Timer Load Register 1 (PITMTLD1)

Read: Anytime

Write: Anytime

21.3.2.3 SPI Baud Rate Register (SPIBR)

Module Base +0x0002

	7	6	5	4	3	2	1	0
R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 21-5. SPI Baud Rate Register (SPIBR)

Read: Anytime

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

Table 21-6. SPIBR Field Descriptions

Field	Description
6–4 SPPR[2:0]	**SPI Baud Rate Preselection Bits** — These bits specify the SPI baud rates as shown in Table 21-7. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.
2–0 SPR[2:0]	**SPI Baud Rate Selection Bits** — These bits specify the SPI baud rates as shown in Table 21-7. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.

The baud rate divisor equation is as follows:

$$\text{BaudRateDivisor} = (\text{SPPR} + 1) \bullet 2^{(\text{SPR} + 1)} \qquad \textit{Eqn. 21-1}$$

The baud rate can be calculated with the following equation:

$$\text{Baud Rate} = \text{BusClock} / \text{BaudRateDivisor} \qquad \textit{Eqn. 21-2}$$

NOTE

For maximum allowed baud rates, please refer to the SPI Electrical Specification in the Electricals chapter of this data sheet.

Table 21-7. Example SPI Baud Rate Selection (25 MHz Bus Clock) (Sheet 1 of 3)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud Rate Divisor	Baud Rate
0	0	0	0	0	0	2	12.5 Mbit/s
0	0	0	0	0	1	4	6.25 Mbit/s
0	0	0	0	1	0	8	3.125 Mbit/s
0	0	0	0	1	1	16	1.5625 Mbit/s
0	0	0	1	0	0	32	781.25 kbit/s
0	0	0	1	0	1	64	390.63 kbit/s
0	0	0	1	1	0	128	195.31 kbit/s
0	0	0	1	1	1	256	97.66 kbit/s
0	0	1	0	0	0	4	6.25 Mbit/s
0	0	1	0	0	1	8	3.125 Mbit/s
0	0	1	0	1	0	16	1.5625 Mbit/s
0	0	1	0	1	1	32	781.25 kbit/s

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0006 TSCR1	R	TEN	TSWAI	TSFRZ	TFFCA	PRNT	0	0	0
	W								
0x0007 TTOV	R	TOV7	TOV6	TOV5	TOV4	TOV3	TOV2	TOV1	TOV0
	W								
0x0008 TCTL1	R	OM7	OL7	OM6	OL6	OM5	OL5	OM4	OL4
	W								
0x0009 TCTL2	R	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
	W								
0x000A TCTL3	R	EDG7B	EDG7A	EDG6B	EDG6A	EDG5B	EDG5A	EDG4B	EDG4A
	W								
0x000B TCTL4	R	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
	W								
0x000C TIE	R	C7I	C6I	C5I	C4I	C3I	C2I	C1I	C0I
	W								
0x000D TSCR2	R	TOI	0	0	0	TCRE	PR2	PR1	PR0
	W								
0x000E TFLG1	R	C7F	C6F	C5F	C4F	C3F	C2F	C1F	C0F
	W								
0x000F TFLG2	R	TOF	0	0	0	0	0	0	0
	W								
0x0010–0x001F TCxH–TCxL	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
0x0020 PACTL	R	0	PAEN	PAMOD	PEDGE	CLK1	CLK0	PAOVI	PAI
	W								
0x0021 PAFLG	R	0	0	0	0	0	0	PAOVF	PAIF
	W								
0x0022 PACNTH	R	PACNT15	PACNT14	PACNT13	PACNT12	PACNT11	PACNT10	PACNT9	PACNT8
	W								
0x0023 PACNTL	R	PACNT7	PACNT6	PACNT5	PACNT4	PACNT3	PACNT2	PACNT1	PACNT0
	W								
0x0024–0x002B Reserved	R								
	W								

= Unimplemented or Reserved

Figure 22-5. TIM16B8CV2 Register Summary (Sheet 2 of 3)

Table 24-53. Verify Backdoor Access Key Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters
011	Key 2
100	Key 3

Upon clearing CCIF to launch the Verify Backdoor Access Key command, the Memory Controller will check the FSEC KEYEN bits to verify that this command is enabled. If not enabled, the Memory Controller sets the ACCERR bit in the FSTAT register and terminates. If the command is enabled, the Memory Controller compares the key provided in FCCOB to the backdoor comparison key in the Flash configuration field with Key 0 compared to 0x7F_FF00, etc. If the backdoor keys match, security will be released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are aborted (set ACCERR) until a reset occurs. The CCIF flag is set after the Verify Backdoor Access Key operation has completed.

Table 24-54. 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 an incorrect backdoor key is supplied
		Set if backdoor key access has not been enabled (KEYEN[1:0] != 10, see Section 24.3.2.2)
		Set if the backdoor key has mismatched since the last reset
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

24.4.2.12 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 24-55. 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.

Table 24-66. Program D-Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] < 010 at command launch
		Set if CCOBIX[2:0] > 101 at command launch
		Set if command not available in current mode (see Table 24-30)
		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 an area in the D-Flash EEE partition
		Set if the requested group of words breaches the end of the D-Flash block or goes into 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

24.4.2.17 Erase D-Flash Sector Command

The Erase D-Flash Sector operation will erase all addresses in a sector of the D-Flash user partition.

Table 24-67. Erase D-Flash Sector Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x12	Global address [22:16] to identify D-Flash block
001	Global address [15:0] anywhere within the sector to be erased. See Section 24.1.2.2 for D-Flash sector size.	

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

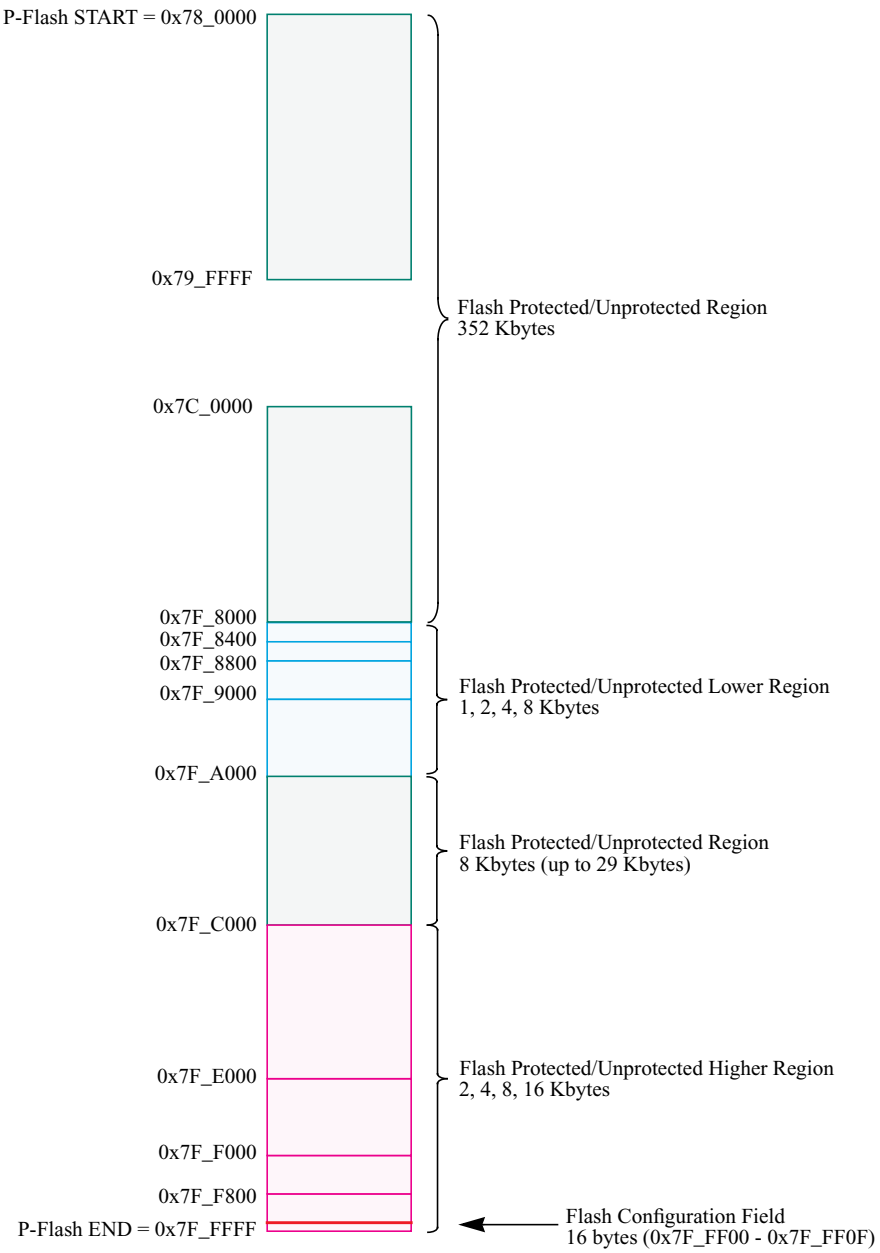


Figure 26-2. P-Flash Memory Map

state. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag is set after the Unsecure Flash operation has completed.

Table 26-54. Unsecure Flash 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 command not available in current mode (see Table 26-30)
	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 FTM512K3.

26.4.2.12 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command will only execute if it is enabled by the KEYEN bits in the FSEC register (see Table 26-11). The Verify Backdoor Access Key command releases security if user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see Table 26-3). The Verify Backdoor Access Key command must not be executed from the Flash block containing the backdoor comparison key to avoid code runaway.

Table 26-55. Verify Backdoor Access Key Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0C	Not required
001	Key 0	
010	Key 1	
011	Key 2	
100	Key 3	

Upon clearing CCIF to launch the Verify Backdoor Access Key command, the Memory Controller will check the FSEC KEYEN bits to verify that this command is enabled. If not enabled, the Memory Controller sets the ACCERR bit in the FSTAT register and terminates. If the command is enabled, the Memory Controller compares the key provided in FCCOB to the backdoor comparison key in the Flash configuration field with Key 0 compared to 0x7F_FF00, etc. If the backdoor keys match, security will be released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are aborted (set ACCERR) until a reset occurs. The CCIF flag is set after the Verify Backdoor Access Key operation has completed.

27.1.2.2 D-Flash Features

- Up to 32 Kbytes of D-Flash memory with 256 byte sectors for user access
- Dedicated commands to control access to the D-Flash memory over EEE operation
- Single bit fault correction and double bit fault detection within a word during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and word program operation
- Ability to program up to four words in a burst sequence

27.1.2.3 Emulated EEPROM Features

- Up to 4 Kbytes of emulated EEPROM (EEE) accessible as 4 Kbytes of RAM
- Flexible protection scheme to prevent accidental program or erase of data
- Automatic EEE file handling using an internal Memory Controller
- Automatic transfer of valid EEE data from D-Flash memory to buffer RAM on reset
- Ability to monitor the number of outstanding EEE related buffer RAM words left to be programmed into D-Flash memory
- Ability to disable EEE operation and allow priority access to the D-Flash memory
- Ability to cancel all pending EEE operations and allow priority access to the D-Flash memory

27.1.2.4 User Buffer RAM Features

- Up to 4 Kbytes of RAM for user access

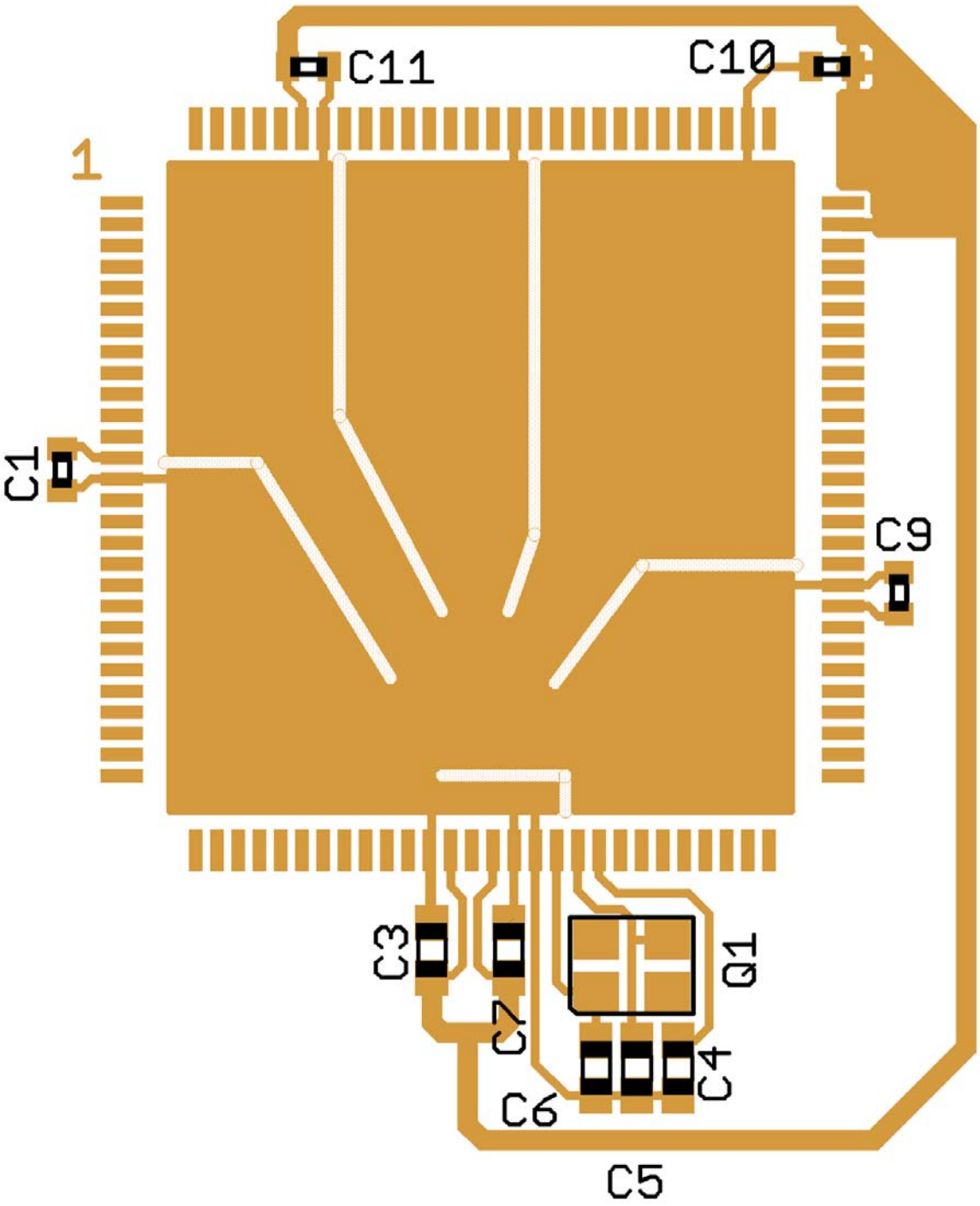
27.1.2.5 Other Flash Module Features

- No external high-voltage power supply required for Flash memory program and erase operations
- Interrupt generation on Flash command completion and Flash error detection
- Security mechanism to prevent unauthorized access to the Flash memory

27.1.3 Block Diagram

The block diagram of the Flash module is shown in Figure 27-1.

Figure C-2. 112-Pin LQFP Recommended PCB Layout (Loop Controlled Pierce Oscillator)



0x0400–0x07FF Reserved

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0400–0x07FF	Reserved	R	0	0	0	0	0	0	0	0
		W								