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	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	32K 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/s912xeq512bcagr

Table 1-14. Interrupt Vector Locations (Sheet 2 of 4)

Vector Address[1]	XGATE Channel ID[2]	Interrupt Source	CCR Mask	Local Enable	STOP Wake up	WAIT Wake up
Vector base + $BE	$5F	SPI1	I bit	SPI1CR1 (SPIE, SPTIE)	No	Yes
Vector base + $BC	$5E	SPI2	I bit	SPI2CR1 (SPIE, SPTIE)	No	Yes
Vector base + $BA	$5D	FLASH Fault Detect	I bit	FCNFG2 (FDIE)	No	No
Vector base + $B8	$5C	FLASH	I bit	FCNFG (CCIE, CBEIE)	No	Yes
Vector base + $B6	$5B	CAN0 wake-up	I bit	CAN0RIER (WUPIE)	Yes	Yes
Vector base + $B4	$5A	CAN0 errors	I bit	CAN0RIER (CSCIE, OVRIE)	No	Yes
Vector base + $B2	$59	CAN0 receive	I bit	CAN0RIER (RXFIE)	No	Yes
Vector base + $B0	$58	CAN0 transmit	I bit	CAN0TIER (TXEIE[2:0])	No	Yes
Vector base + $AE	$57	CAN1 wake-up	I bit	CAN1RIER (WUPIE)	Yes	Yes
Vector base + $AC	$56	CAN1 errors	I bit	CAN1RIER (CSCIE, OVRIE)	No	Yes
Vector base + $AA	$55	CAN1 receive	I bit	CAN1RIER (RXFIE)	No	Yes
Vector base + $A8	$54	CAN1 transmit	I bit	CAN1TIER (TXEIE[2:0])	No	Yes
Vector base + $A6	$53	CAN2 wake-up	I bit	CAN2RIER (WUPIE)	Yes	Yes
Vector base + $A4	$52	CAN2 errors	I bit	CAN2RIER (CSCIE, OVRIE)	No	Yes
Vector base + $A2	$51	CAN2 receive	I bit	CAN2RIER (RXFIE)	No	Yes
Vector base + $A0	$50	CAN2 transmit	I bit	CAN2TIER (TXEIE[2:0])	No	Yes
Vector base + $9E	$4F	CAN3 wake-up	I bit	CAN3RIER (WUPIE)	Yes	Yes
Vector base+ $9C	$4E	CAN3 errors	I bit	CAN3RIER (CSCIE, OVRIE)	No	Yes
Vector base+ $9A	$4D	CAN3 receive	I bit	CAN3RIER (RXFIE)	No	Yes
Vector base + $98	$4C	CAN3 transmit	I bit	CAN3TIER (TXEIE[2:0])	No	Yes
Vector base + $96	$4B	CAN4 wake-up	I bit	CAN4RIER (WUPIE)	Yes	Yes
Vector base + $94	$4A	CAN4 errors	I bit	CAN4RIER (CSCIE, OVRIE)	No	Yes
Vector base + $92	$49	CAN4 receive	I bit	CAN4RIER (RXFIE)	No	Yes
Vector base + $90	$48	CAN4 transmit	I bit	CAN4TIER (TXEIE[2:0])	No	Yes
Vector base + $8E	$47	Port P Interrupt	I bit	PIEP (PIEP7-PIEP0)	Yes	Yes
Vector base+ $8C	$46	PWM emergency shutdown	I bit	PWMSDN (PWMIE)	No	Yes

2.3.76 Port AD0 Pull Up Enable Register 1 (PER1AD0)

Address 0x0277 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	PER1AD07	PER1AD06	PER1AD05	PER1AD04	PER1AD03	PER1AD02	PER1AD01	PER1AD00
Reset	0	0	0	0	0	0	0	0

Figure 2-74. Port AD0 Pull Up Enable Register 1 (PER1AD0)

1. Read: Anytime.
   Write: Anytime.

Table 2-72. PER1AD0 Register Field Descriptions

Field	Description
7-0 PER1AD0	**Port AD0 pull device enable**—Enable pull devices on input pins These bits configure whether a pull device is activated, if the associated pin is used as an input. This bit has no effect if the pin is used as an output. Out of reset no pull device is enabled. 1 Pull device enabled. 0 Pull device disabled.

2.3.77 Port AD1 Data Register 0 (PT0AD1)

Address 0x0278 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	PT0AD17	PT0AD16	PT0AD15	PT0AD14	PT0AD13	PT0AD12	PT0AD11	PT0AD10
Altern. Function	AN15	AN14	AN13	AN12	AN11	AN10	AN9	AN8
Reset	0	0	0	0	0	0	0	0

Figure 2-75. Port AD1 Data Register 0 (PT0AD1)

1. Read: Anytime.
   Write: Anytime.

Table 2-73. PT0AD1 Register Field Descriptions

Field	Description
7-0 PT0AD1	**Port AD1 general purpose input/output data**—Data Register This register is associated with ATD1 analog inputs AN[15:8] on PAD[31:24], respectively. When not used with the alternative function, this pin 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.

5.1.1 Glossary or Terms

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
external resource	Addresses outside MCU
PRR	Port Replacement Registers
PRU	Port Replacement Unit
EMULMEM	External emulation memory
access source	CPU or BDM or XGATE

5.1.2 Features

The XEBI includes the following features:

- Output of up to 23-bit address bus and control signals to be used with a non-muxed external bus
- Bidirectional 16-bit external data bus with option to disable upper half
- Visibility of internal bus activity

5.1.3 Modes of Operation

- Single-chip modes

  The external bus interface is not available in these modes.
- Expanded modes

  Address, data, and control signals are activated on the external bus in normal expanded mode and special test mode.
- Emulation modes

  The external bus is activated to interface to an external tool for emulation of normal expanded mode or normal single-chip mode applications.

Table 8-23. State1 Sequencer Next State Selection (continued)

SC[3:0]	Description
0011	Match2 triggers to State2..... Other matches have no effect
0100	Match2 triggers to State3..... Other matches have no effect
0101	Match2 triggers to Final State..... Other matches have no effect
0110	Match0 triggers to State2..... Match1 triggers to State3..... Other matches have no effect
0111	Match1 triggers to State3..... Match0 triggers Final State..... Other matches have no effect
1000	Match0 triggers to State2..... Match2 triggers to State3..... Other matches have no effect
1001	Match2 triggers to State3..... Match0 triggers Final State..... Other matches have no effect
1010	Match1 triggers to State2..... Match3 triggers to State3..... Other matches have no effect
1011	Match3 triggers to State3..... Match1 triggers to Final State..... Other matches have no effect
1100	Match3 has no effect..... All other matches (M0,M1,M2) trigger to State2
1101	Reserved. (No match triggers state sequencer transition)
1110	Reserved. (No match triggers state sequencer transition)
1111	Reserved. (No match triggers state sequencer transition)

The trigger priorities described in Table 8-42 dictate that in the case of simultaneous matches, the match on the lower channel number (0,1,2,3) has priority. The SC[3:0] encoding ensures that a match leading to final state has priority over all other matches.

8.3.2.7.2 Debug State Control Register 2 (DBGSCR2)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	SC3	SC2	SC1	SC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 8-10. Debug State Control Register 2 (DBGSCR2)

Read: If COMRV[1:0] = 01

Write: If COMRV[1:0] = 01 and S12XDBG is not armed.

This register is visible at 0x0027 only with COMRV[1:0] = 01. The state control register 2 selects the targeted next state whilst in State2. The matches refer to the match channels of the comparator match control logic as depicted in Figure 8-1 and described in Section 8.3.2.8.1. Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 8-24. DBGSCR2 Field Descriptions

Field	Description
3–0 SC[3:0]	These bits select the targeted next state whilst in State2, based upon the match event.

Table 8-25. State2 —Sequencer Next State Selection

SC[3:0]	Description
0000	Any match triggers to state1

Table 10-9. XGVBR Field Descriptions

Field	Description
15–1 XBVBR[15:1]	**Vector Base Address** — The XGVBR register holds the start address of the vector block in the XGATE memory map.

10.3.1.8 XGATE Channel Interrupt Flag Vector (XGIF)

The XGATE Channel Interrupt Flag Vector (Figure 10-10) provides access to the interrupt flags of all channels. Each flag may be cleared by writing a "1" to its bit location. Refer to Section 10.5.2, “Outgoing Interrupt Requests” for further information.

Module Base +0x0008

	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112
R	0	0	0	0	0	0	0	XGIF_78	XGF_77	XGIF_76	XGIF_75	XGIF_74	XGIF_73	XGIF_72	XGIF_71	XGIF_70
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
R	XGIF_6F	XGIF_6E	XGIF_6D	XGIF_6C	XGIF_6B	XGIF_6A	XGIF_69	XGIF_68	XGF_67	XGIF_66	XGIF_65	XGIF_64	XGIF_63	XGIF_62	XGIF_61	XGIF_60
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
R	XGIF_5F	XGIF_5E	XGIF_5D	XGIF_5C	XGIF_5B	XGIF_5A	XGIF_59	XGIF_58	XGF_57	XGIF_56	XGIF_55	XGIF_54	XGIF_53	XGIF_52	XGIF_51	XGIF_50
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
R	XGIF_4F	XGIF_4E	XGIF_4D	XGIF_4C	XGIF_4B	XGIF_4A	XGIF_49	XGIF_48	XGF _47	XGIF_46	XGIF_45	XGIF_44	XGIF_43	XGIF_42	XGIF_41	XGIF_40
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 10-10. XGATE Channel Interrupt Flag Vector (XGIF)

set_xgsem: 1 is written to XGSEM[*n*] (and 1 is written to XGSEMM[*n*])
clr_xgsem: 0 is written to XGSEM[*n*] (and 1 is written to XGSEMM[*n*])
ssem: Executing SSEM instruction (on semaphore *n*)
csem: Executing CSEM instruction (on semaphore *n*)

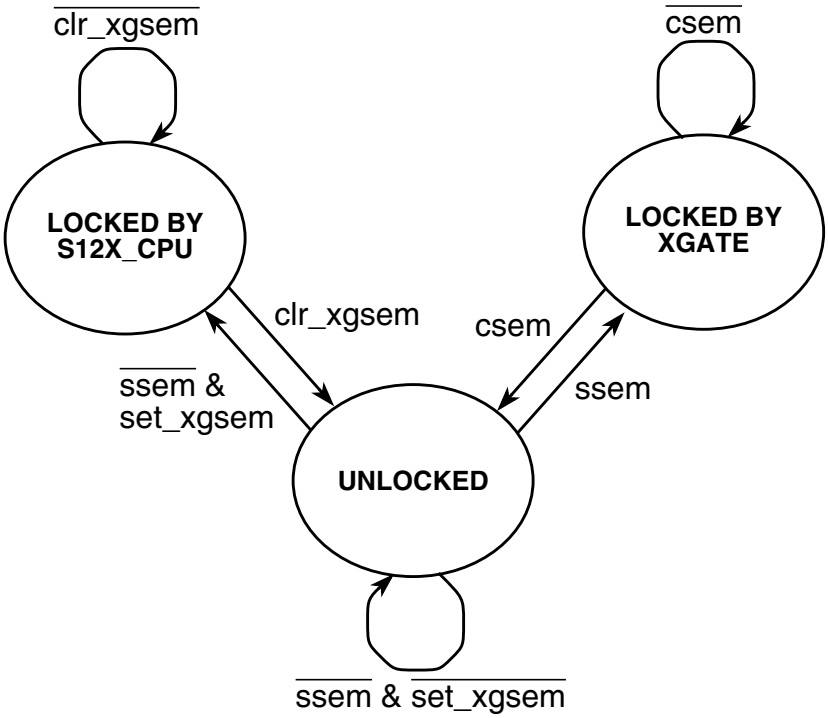


Figure 10-24. Semaphore State Transitions

Figure 10-25 gives an example of the typical usage of the XGATE hardware semaphores.

Two concurrent threads are running on the system. One is running on the S12X_CPU and the other is running on the RISC core. They both have a critical section of code that accesses the same system resource. To guarantee that the system resource is only accessed by one thread at a time, the critical code sequence must be embedded in a semaphore lock/release sequence as shown.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 CANCTL0	R	RXFRM	RXACT	CSWAI	SYNCH	TIME	WUPE	SLPRQ	INITRQ
	W								
0x0001 CANCTL1	R	CANE	CLKSRC	LOOPB	LISTEN	BORM	WUPM	SLPAK	INITAK
	W								
0x0002 CANBTR0	R	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
	W								
0x0003 CANBTR1	R	SAMP	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
	W								
0x0004 CANRFLG	R	WUPIF	CSCIF	RSTAT1	RSTAT0	TSTAT1	TSTAT0	OVRIF	RXF
	W								
0x0005 CANRIER	R	WUPIE	CSCIE	RSTATE1	RSTATE0	TSTATE1	TSTATE0	OVRIE	RXFIE
	W								
0x0006 CANTFLG	R	0	0	0	0	0	TXE2	TXE1	TXE0
	W								
0x0007 CANTIER	R	0	0	0	0	0	TXEIE2	TXEIE1	TXEIE0
	W								
0x0008 CANTARQ	R	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
	W								
0x0009 CANTAAK	R	0	0	0	0	0	ABTAK2	ABTAK1	ABTAK0
	W								
0x000A CANTBSEL	R	0	0	0	0	0	TX2	TX1	TX0
	W								
0x000B CANIDAC	R	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
	W								
0x000C Reserved	R	0	0	0	0	0	0	0	0
	W								
0x000D CANMISC	R	0	0	0	0	0	0	0	BOHOLD
	W								
0x000E CANRXERR	R	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
	W								

= Unimplemented or Reserved

Figure 16-3. MSCAN Register Summary

22.2.7 IOC1 — Input Capture and Output Compare Channel 1 Pin

This pin serves as input capture or output compare for channel 1.

22.2.8 IOC0 — Input Capture and Output Compare Channel 0 Pin

This pin serves as input capture or output compare for channel 0.

NOTE

For the description of interrupts see Section 22.6, “Interrupts”.

22.3 Memory Map and Register Definition

This section provides a detailed description of all memory and registers.

22.3.1 Module Memory Map

The memory map for the TIM16B8CV2 module is given below in Figure 22-5. The address listed for each register is the address offset. The total address for each register is the sum of the base address for the TIM16B8CV2 module and the address offset for each register.

22.3.2 Register Descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 TIOS	R W	IOS7	IOS6	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0
0x0001 CFORC	R W	0 FOC7	0 FOC6	0 FOC5	0 FOC4	0 FOC3	0 FOC2	0 FOC1	0 FOC0
0x0002 OC7M	R W	OC7M7	OC7M6	OC7M5	OC7M4	OC7M3	OC7M2	OC7M1	OC7M0
0x0003 OC7D	R W	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0
0x0004 TCNTH	R W	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8
0x0005 TCNTL	R W	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0

= Unimplemented or Reserved

Figure 22-5. TIM16B8CV2 Register Summary (Sheet 1 of 3)

Offset Module Base + 0x0011

	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 24-23. Flash Reserved0 Register (FRSV0)

All bits in the FRSV0 register read 0 and are not writable.

24.3.2.16 Flash Reserved1 Register (FRSV1)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0012

	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 24-24. Flash Reserved1 Register (FRSV1)

All bits in the FRSV1 register read 0 and are not writable.

24.3.2.17 Flash Reserved2 Register (FRSV2)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0013

	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 24-25. Flash Reserved2 Register (FRSV2)

All bits in the FRSV2 register read 0 and are not writable.

Address & Name		7	6	5	4	3	2	1	0
0x0006 FSTAT	R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT1	MGSTAT0
	W								
0x0007 FERSTAT	R	ERSERIF	PGMERIF	0	EPVIOLIF	ERSVIF1	ERSVIF0	DFDIF	SFDIF
	W								
0x0008 FPROT	R	FPOPEN	RNV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
	W								
0x0009 EPROT	R	EPOPEN	RNV6	RNV5	RNV4	EPDIS	EPS2	EPS1	EPS0
	W								
0x000A FCCOBHI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x000B FCCOBLO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x000C ETAGHI	R	ETAG15	ETAG14	ETAG13	ETAG12	ETAG11	ETAG10	ETAG9	ETAG8
	W								
0x000D ETAGLO	R	ETAG7	ETAG6	ETAG5	ETAG4	ETAG3	ETAG2	ETAG1	ETAG0
	W								
0x000E FECCRHI	R	ECCR15	ECCR14	ECCR13	ECCR12	ECCR11	ECCR10	ECCR9	ECCR8
	W								
0x000F FECCRLO	R	ECCR7	ECCR6	ECCR5	ECCR4	ECCR3	ECCR2	ECCR1	ECCR0
	W								
0x0010 FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
	W								
0x0011 FRSV0	R	0	0	0	0	0	0	0	0
	W								
0x0012 FRSV1	R	0	0	0	0	0	0	0	0
	W								
0x0013 FRSV2	R	0	0	0	0	0	0	0	0
	W								

Figure 25-4. FTM256K2 Register Summary (continued)

Table 25-14. FECCRIX Field Descriptions

Field	Description
2-0 ECCRIX[2:0]	**ECC Error Register Index**— The ECCRIX bits are used to select which word of the FECCR register array is being read. See Section 25.3.2.13, “Flash ECC Error Results Register (FECCR),” for more details.

25.3.2.5 Flash Configuration Register (FCNFG)

The FCNFG register enables the Flash command complete interrupt and forces ECC faults on Flash array read access from the CPU or XGATE.

Offset Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 25-9. Flash Configuration Register (FCNFG)

CCIE, IGNSF, FDFD, and FSFD bits are readable and writable while remaining bits read 0 and are not writable.

Table 25-15. FCNFG Field Descriptions

Field	Description
7 CCIE	**Command Complete Interrupt Enable** — The CCIE bit controls interrupt generation when a Flash command has completed. 0 Command complete interrupt disabled 1 An interrupt will be requested whenever the CCIF flag in the FSTAT register is set (see Section 25.3.2.7)
4 IGNSF	**Ignore Single Bit Fault** — The IGNSF controls single bit fault reporting in the FERSTAT register (see Section 25.3.2.8). 0 All single bit faults detected during array reads are reported 1 Single bit faults detected during array reads are not reported and the single bit fault interrupt will not be generated

Table 25-41. Load Data Field Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x05	Global address [22:16] to identify P-Flash block
001	Global address [15:0] of phrase location to be programmed(1)	
010	Word 0	
011	Word 1	
100	Word 2	
101	Word 3	

1. Global address [2:0] must be 000

Upon clearing CCIF to launch the Load Data Field command, the FCCOB registers will be transferred to the Memory Controller and be programmed in the block specified at the global address given with a future Program P-Flash command launched on a P-Flash block. The CCIF flag will set after the Load Data Field operation has completed. Note that once a Load Data Field command sequence has been initiated, the Load Data Field command sequence will be cancelled if any command other than Load Data Field or the future Program P-Flash is launched. Similarly, if an error occurs after launching a Load Data Field or Program P-Flash command, the associated Load Data Field command sequence will be cancelled.

Table 25-42. Load Data Field Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 101 at command launch
		Set if command not available in current mode (see Table 25-30)
		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 a Load Data Field command sequence is currently active and the selected block has previously been selected in the same command sequence
		Set if a Load Data Field command sequence is currently active and global address [16:0] does not match that previously supplied in the same command sequence
	FPVIOL	Set if the global address [22:0] points to a protected area
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

25.4.2.6 Program P-Flash Command

The Program P-Flash operation will program a previously erased phrase in the P-Flash memory using an embedded algorithm.

Table 25-47. 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 25-48. Erase All Blocks 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 25-30)
	FPVIOL	Set if any area of the P-Flash memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation
FERSTAT	EPVIOLIF	Set if any area of the buffer RAM EEE partition is protected

25.4.2.9 Erase P-Flash Block Command

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

Table 25-49. 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.

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 25-56. 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 25.3.2.2)
		Set if the backdoor key has mismatched since the last reset
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

25.4.2.13 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 25-57. 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 25-58.

Table 25-58. Valid Set User Margin Level Settings

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

Table 26-14. FECCRIX Field Descriptions

Field	Description
2-0 ECCRIX[2:0]	**ECC Error Register Index**— The ECCRIX bits are used to select which word of the FECCR register array is being read. See Section 26.3.2.13, “Flash ECC Error Results Register (FECCR),” for more details.

26.3.2.5 Flash Configuration Register (FCNFG)

The FCNFG register enables the Flash command complete interrupt and forces ECC faults on Flash array read access from the CPU or XGATE.

Offset Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 26-9. Flash Configuration Register (FCNFG)

CCIE, IGNSF, FDFD, and FSFD bits are readable and writable while remaining bits read 0 and are not writable.

Table 26-15. FCNFG Field Descriptions

Field	Description
7 CCIE	**Command Complete Interrupt Enable** — The CCIE bit controls interrupt generation when a Flash command has completed. 0 Command complete interrupt disabled 1 An interrupt will be requested whenever the CCIF flag in the FSTAT register is set (see Section 26.3.2.7)
4 IGNSF	**Ignore Single Bit Fault** — The IGNSF controls single bit fault reporting in the FERSTAT register (see Section 26.3.2.8). 0 All single bit faults detected during array reads are reported 1 Single bit faults detected during array reads are not reported and the single bit fault interrupt will not be generated

Table 27-16. FERCNFG Field Descriptions (continued)

Field	Description
3 ERSVIE1	**EEE Error Type 1 Interrupt Enable** — The ERSVIE1 bit controls interrupt generation when a change state error is detected during an EEE operation. 0 ERSVIF1 interrupt disabled 1 An interrupt will be requested whenever the ERSVIF1 flag is set (see Section 27.3.2.8)
2 ERSVIE0	**EEE Error Type 0 Interrupt Enable** — The ERSVIE0 bit controls interrupt generation when a sector format error is detected during an EEE operation. 0 ERSVIF0 interrupt disabled 1 An interrupt will be requested whenever the ERSVIF0 flag is set (see Section 27.3.2.8)
1 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. 0 DFDIF interrupt disabled 1 An interrupt will be requested whenever the DFDIF flag is set (see Section 27.3.2.8)
0 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. 0 SFDIF interrupt disabled whenever the SFDIF flag is set (see Section 27.3.2.8) 1 An interrupt will be requested whenever the SFDIF flag is set (see Section 27.3.2.8)

27.3.2.7 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 27-11. 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 27.6).

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.

(0x7F_FF0F). The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the Flash protection register, FPROT.

28.5.2 Unsecuring the MCU in Special Single Chip Mode using BDM

The MCU can be unsecured in special single chip mode by erasing the P-Flash and D-Flash memory by one of the following methods:

- Reset the MCU into special single chip mode, delay while the erase test is performed by the BDM, send BDM commands to disable protection in the P-Flash and D-Flash memory, and execute the Erase All Blocks command write sequence to erase the P-Flash and D-Flash memory.
- Reset the MCU into special expanded wide mode, disable protection in the P-Flash and D-Flash memory and run code from external memory to execute the Erase All Blocks command write sequence to erase the P-Flash and D-Flash memory.

After the CCIF flag sets to indicate that the Erase All Blocks operation has completed, reset the MCU into special single chip mode. The BDM will execute the Erase Verify All Blocks command write sequence to verify that the P-Flash and D-Flash memory is erased. If the P-Flash and D-Flash memory are verified as erased the MCU will be unsecured. All BDM commands will be enabled and the Flash security byte may be programmed to the unsecure state by the following method:

- Send BDM commands to execute a ‘Program P-Flash’ command sequence to program the Flash security byte to the unsecured state and reset the MCU.

28.5.3 Mode and Security Effects on Flash Command Availability

The availability of Flash module commands depends on the MCU operating mode and security state as shown in Table 28-30.

28.6 Initialization

On each system reset the Flash module executes a reset sequence which establishes initial values for the Flash Block Configuration Parameters, the FPROT and DFPROT protection registers, and the FOPT and FSEC registers. The Flash module reverts to built-in default values that leave the module in a fully protected and secured state if errors are encountered during execution of the reset sequence. If a double bit fault is detected during the reset sequence, both MGSTAT bits in the FSTAT register will be set. The ACCERR bit in the FSTAT register is set if errors are encountered while initializing the EEE buffer ram during the reset sequence.

CCIF remains clear throughout the reset sequence. The Flash module holds off all CPU access for the initial portion of the reset sequence. While Flash reads are possible when the hold is removed, writes to the FCCOBIX, FCCOBHI, and FCCOBLO registers are ignored to prevent command activity while the Memory Controller remains busy. Completion of the reset sequence is marked by setting CCIF high which enables writes to the FCCOBIX, FCCOBHI, and FCCOBLO registers to launch any available Flash command.

If a reset occurs while any Flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.

Address & Name	7	6	5	4	3	2	1	0

= Unimplemented or Reserved

Figure 29-4. FTM1024K5 Register Summary (continued)

29.3.2.1 Flash Clock Divider Register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

Offset Module Base + 0x0000

	7	6	5	4	3	2	1	0
R	FDIVLD	FDIV[6:0]						
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 29-5. Flash Clock Divider Register (FCLKDIV)

All bits in the FCLKDIV register are readable, bits 6–0 are write once and bit 7 is not writable.

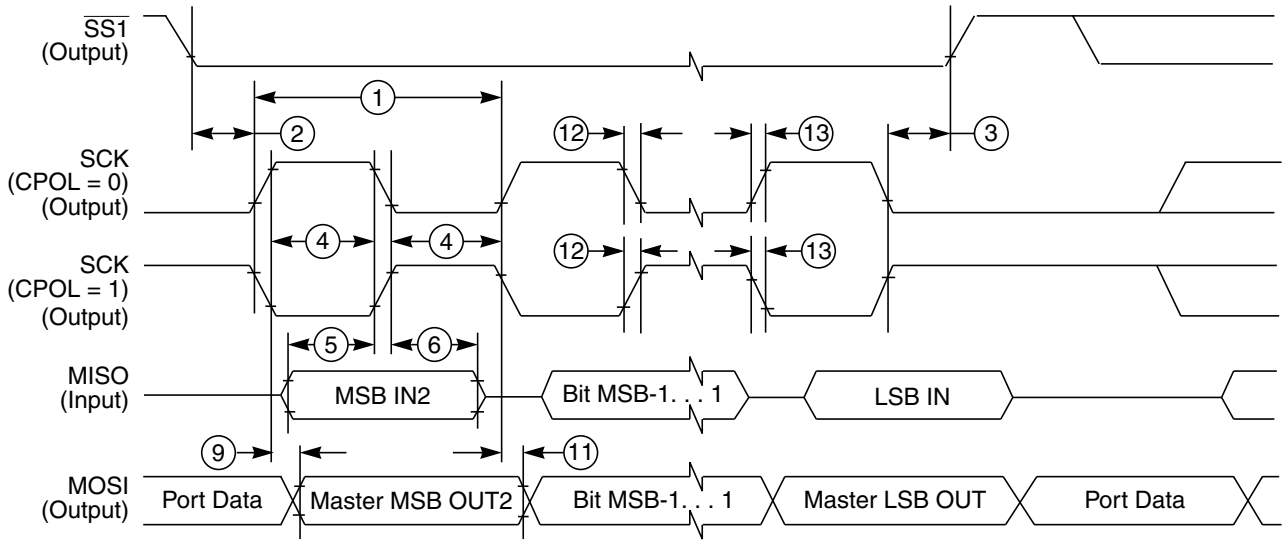
Table 29-8. FCLKDIV Field Descriptions

Field	Description
7 FDIVLD	**Clock Divider Loaded** 0 FCLKDIV register has not been written 1 FCLKDIV register has been written since the last reset
6–0 FDIV[6:0]	**Clock Divider Bits** — FDIV[6:0] must be set to effectively divide OSCCLK down to generate an internal Flash clock, FCLK, with a target frequency of 1 MHz for use by the Flash module to control timed events during program and erase algorithms. Table 29-9 shows recommended values for FDIV[6:0] based on OSCCLK frequency. Please refer to Section 29.4.1, “Flash Command Operations,” for more information.

CAUTION

The FCLKDIV register should never be written while a Flash command is executing (CCIF=0). The FCLKDIV register is writable during the Flash reset sequence even though CCIF is clear.

In Figure A-8 the timing diagram for master mode with transmission format CPHA=1 is depicted.



1.If configured as output
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1,bit 2... MSB.

Figure A-8. SPI Master Timing (CPHA = 1)

D.2 Pinout explanations:

- Pinout compatibility is maintained throughout the device family
- A/D is the number of modules/total number of A/D channels.
- I/O is the sum of ports capable to act as digital input or output. .
- For additional flexibility especially for the low pin count packages several I/O functions can be routed under software control to different pins. For details refer to the device overview section..

- Versions with 5 CAN modules will have CAN0, CAN1, CAN2, CAN3 and CAN4.
- Versions with 4 CAN modules will have CAN0, CAN1, CAN2 and CAN4.
- Versions with 3 CAN modules will have CAN0, CAN1 and CAN4.
- Versions with 2 CAN modules will have CAN0 and CAN4.
- Versions with 1 CAN module will have CAN0.

- Versions with 3 SPI modules will have SPI0, SPI1 and SPI2.
- Versions with 2 SPI modules will have SPI0 and SPI1.
- Versions with 1 SPI modules will have SPI0.

- Versions with 8 SCI modules will have SCI0, SCI1, SCI2, SCI3, SCI4, SCI5, SCI6 and SCI7.
- Versions with 7 SCI modules will have SCI0, SCI1, SCI2, SCI3, SCI4, SCI5, and SCI6.
- Versions with 6 SCI modules will have SCI0, SCI1, SCI2, SCI3, SCI4 and SCI5.
- Versions with 5 SCI modules will have SCI0, SCI1, SCI2, SCI3 and SCI4.
- Versions with 4 SCI modules will have SCI0, SCI1, SCI2 and SCI4.
- Versions with 3 SCI modules will have SCI0, SCI1 and SCI2.
- Versions with 2 SCI modules will have SCI0 and SCI1.
- Versions with 1 SCI module will have SCI0.

- Versions with 2 IIC modules will have IIC0 and IIC1.
- Versions with 1 IIC module will have IIC0.

- Versions with 1 ATD module will have ATD0.