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	59
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s912xea128j2maa

Table 8-16. CDCM Encoding

CDCM	Description
00	Match2 mapped to comparator C match..... Match3 mapped to comparator D match.
01	Match2 mapped to comparator C/D inside range..... Match3 disabled.
10	Match2 mapped to comparator C/D outside range..... Match3 disabled.
11	Reserved(1)

1. Currently defaults to Match2 mapped to comparator C : Match3 mapped to comparator D

Table 8-17. ABCM Encoding

ABCM	Description
00	Match0 mapped to comparator A match..... Match1 mapped to comparator B match.
01	Match 0 mapped to comparator A/B inside range..... Match1 disabled.
10	Match 0 mapped to comparator A/B outside range..... Match1 disabled.
11	Reserved(1)

1. Currently defaults to Match0 mapped to comparator A : Match1 mapped to comparator B

8.3.2.5 Debug Trace Buffer Register (DBGTBH:DBGTBL)

Address: 0x0024, 0x0025

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
POR	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Other Resets	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Figure 8-7. Debug Trace Buffer Register (DBGTB)

Read: Only when unlocked AND not secured AND not armed AND with a TSOURCE bit set.

Write: Aligned word writes when disarmed unlock the trace buffer for reading but do not affect trace buffer contents.

Table 8-18. DBGTB Field Descriptions

Field	Description
15–0 Bit[15:0]	**Trace Buffer Data Bits** — The Trace Buffer Register is a window through which the 64-bit wide data lines of the Trace Buffer may be read 16 bits at a time. Each valid read of DBGTB increments an internal trace buffer pointer which points to the next address to be read. When the ARM bit is written to 1 the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by writing to DBGTB with an aligned word write when the module is disarmed. The DBGTB register can be read only as an aligned word, any byte reads or misaligned access of these registers will return 0 and will not cause the trace buffer pointer to increment to the next trace buffer address. The same is true for word reads while the debugger is armed. The POR state is undefined Other resets do not affect the trace buffer contents. .

Module Base +0x0002

Figure 10-4. XGATE Channel ID Register (XGCHID)

Read: Anytime

Write: In Debug Mode[1]

Table 10-3. XGCHID Field Descriptions

Field	Description
6–0 XGCHID[6:0]	**Request Identifier** — ID of the currently active channel

10.3.1.3 XGATE Channel Priority Level (XGCHPL)

The XGATE Channel Priority Level Register (Figure 10-5) shows the priority level of the current thread. In debug mode this register can be used to select a priority level when launching a thread (see Section 10.6.1, “Debug Features”).

Module Base +0x0003

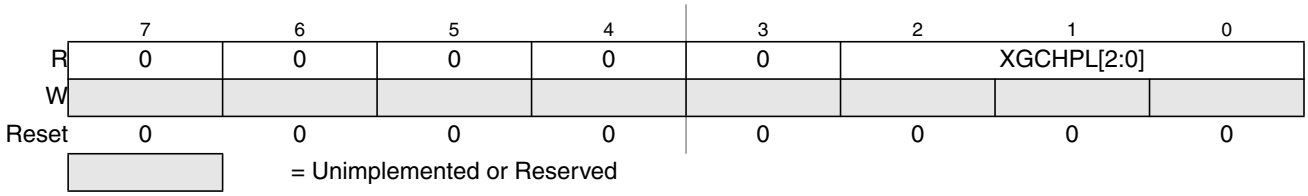

Figure 10-5. XGATE Channel Priority Level Register (XGCHPL)

Read: Anytime

Write: In Debug Mode[1]

Table 10-4. XGCHPL Field Descriptions

Field	Description
2-0 XGCHPL[2:0]	**Priority Level**— Priority level of the currently active channel

10.3.1.4 XGATE Initial Stack Pointer Select Register (XGISPSEL)

The XGATE Initial Stack Pointer Select Register (Figure 10-6) determines the register which is mapped to address “Module Base +0x0006”. A value of zero selects the Vector Base Register (XGVBR). Setting

1. Refer to Section 10.6.1, “Debug Features”

BNE Branch if Not Equal BNE

Operation

If Z = 0, then PC + $0002 + (REL9 << 1) ⇒ PC

Tests the Zero flag and branches if Z = 0.

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
BNE REL9	REL9	0	0	1	0	0	1	0	REL9	PP/P

11.3.2.2 S12XECRG Reference Divider Register (REFDV)

The REFDV register provides a finer granularity for the IPLL multiplier steps.

Module Base + 0x0001

	7	6	5	4	3	2	1	0
R W	REFFRQ[1:0]		REFDIV[5:0]					
Reset	0	0	0	0	0	0	0	0

Figure 11-4. S12XECRG Reference Divider Register (REFDV)

Read: Anytime

Write: Anytime except when PLLSEL = 1

NOTE

Write to this register initializes the lock detector bit.

$$f_{REF} = \frac{f_{OSC}}{(REFDIV + 1)}$$

The REFFRQ[1:0] bit are used to configure the internal PLL filter for optimal stability and lock time. For correct IPLL operation the REFFRQ[1:0] bits have to be selected according to the actual REFCLK frequency as shown in Figure 11-3. Setting the REFFRQ[1:0] bits wrong can result in a non functional IPLL (no locking and/or insufficient stability).

Table 11-3. Reference Clock Frequency Selection

REFCLK Frequency Ranges	REFFRQ[1:0]
1MHz <= f_{REF} <= 2MHz	00
2MHz < f_{REF} <= 6MHz	01
6MHz < f_{REF} <= 12MHz	10
f_{REF} >12MHz	11

11.3.2.3 S12XECRG Post Divider Register (POSTDIV)

The POSTDIV register controls the frequency ratio between the VCOCLK and PLLCLK. The count in the final divider divides VCOCLK frequency by 1 or 2*POSTDIV. Note that if POSTDIV = $00 f_{PLL}= f_{VCO} (divide by one).

11.4 Functional Description

11.4.1 Functional Blocks

11.4.1.1 Phase Locked Loop with Internal Filter (IPLL)

The IPLL is used to run the MCU from a different time base than the incoming OSCCLK. Figure 11-15 shows a block diagram of the IPLL.

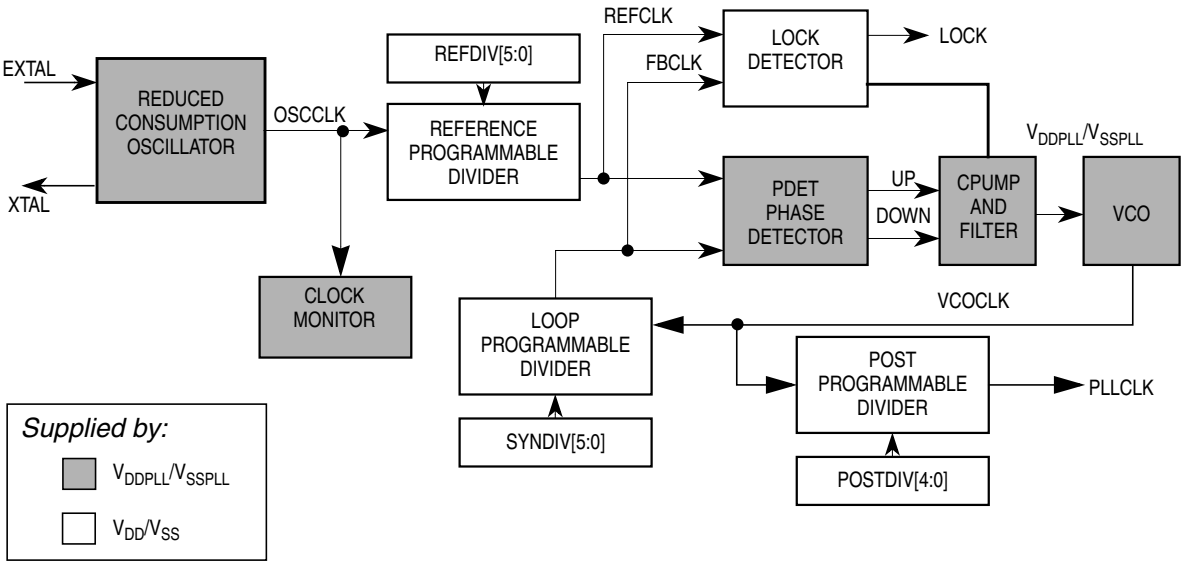

Figure 11-15. IPLL Functional Diagram

For increased flexibility, OSCCLK can be divided in a range of 1 to 64 to generate the reference frequency REFCLK using the REFDIV[5:0] bits. This offers a finer multiplication granularity. Based on the SYNDIV[5:0] bits the IPLL generates the VCOCLK by multiplying the reference clock by a multiple of 2, 4, 6,... 126, 128. Based on the POSTDIV[4:0] bits the VCOCLK can be divided in a range of 1,2,4,6,8,... to 62 to generate the PLLCLK.

$$f_{PLL} = 2 \times f_{OSC} \times \frac{SYNDIV + 1}{[REFDIV + 1][2 \times POSTDIV]}$$

NOTE

Although it is possible to set the dividers to command a very high clock frequency, do not exceed the specified bus frequency limit for the MCU.

If (PLLSEL = 1) then $f_{BUS} = f_{PLL} / 2$.

IF POSTDIV = $00 the f_{PLL} is identical to f_{VCO} (divide by one)

Several examples of IPLL divider settings are shown in Table 11-14. Shaded rows indicated that these settings are not recommended. The following rules help to achieve optimum stability and shortest lock time:

- Use lowest possible f_{VCO} / f_{REF} ratio (SYNDIV value).
- Use highest possible REFCLK frequency f_{REF}.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x001C TC6 (High)	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
0x001D TC6 (Low)	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	W								
0x001E TC7 (High)	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	W								
0x001F TC7 (Low)	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	PA0VI	PAI
	W								
0x0021 PAFLG	R	0	0	0	0	0	0	PA0VF	PAIF
	W								
0x0022 PACN3	R	PACNT7(15)	PACNT6(14)	PACNT5(13)	PACNT4(12)	PACNT3(11)	PACNT2(10)	PACNT1(9)	PACNT0(8)
	W								
0x0023 PACN2	R	PACNT7	PACNT6	PACNT5	PACNT4	PACNT3	PACNT2	PACNT1	PACNT0
	W								
0x0024 PACN1	R	PACNT7(15)	PACNT6(14)	PACNT5(13)	PACNT4(12)	PACNT3(11)	PACNT2(10)	PACNT1(9)	PACNT0(8)
	W								
0x0025 PACN0	R	PACNT7	PACNT6	PACNT5	PACNT4	PACNT3	PACNT2	PACNT1	PACNT0
	W								
0x0026 MCCTL	R	MCZI	MODMC	RDMCL	0	0	MCEN	MCPR1	MCPR0
	W				ICLAT	FLMC			
0x0027 MCFLG	R	MCZF	0	0	0	POLF3	POLF2	POLF1	POLF0
	W								
0x0028 ICPAR	R	0	0	0	0	PA3EN	PA2EN	PA1EN	PA0EN
	W								
0x0029 DLYCT	R	DLY7	DLY6	DLY5	DLY4	DLY3	DLY2	DLY1	DLY0
	W								
0x002A ICOVW	R	NOVW7	NOVW6	NOVW5	NOVW4	NOVW3	NOVW2	NOVW1	NOVW0
	W								

= Unimplemented or Reserved

Figure 14-2. ECT Register Summary (Sheet 3 of 5)

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.

14.3.2.16 Pulse Accumulator A Flag Register (PAFLG)

Module Base + 0x0021

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PAOVF	PAIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 14-37. Pulse Accumulator A Flag Register (PAFLG)

Read: Anytime

Write used in the flag clearing mechanism. Writing a one to the flag clears the flag. Writing a zero will not affect the current status of the bit.

NOTE

When TFFCA = 1, the flags cannot be cleared via the normal flag clearing mechanism (writing a one to the flag). Reference Section 14.3.2.6, “Timer System Control Register 1 (TSCR1)”.

All bits reset to zero.

PAFLG indicates when interrupt conditions have occurred. The flags can be cleared via the normal flag clearing mechanism (writing a one to the flag) or via the fast flag clearing mechanism (Reference TFFCA bit in Section 14.3.2.6, “Timer System Control Register 1 (TSCR1)”).

Table 14-22. PAFLG Field Descriptions

Field	Description
1 PAOVF	**Pulse Accumulator A Overflow Flag** — Set when the 16-bit pulse accumulator A overflows from 0xFFFF to 0x0000, or when 8-bit pulse accumulator 3 (PAC3) overflows from 0x00FF to 0x0000. When PACMX = 1, PAOVF bit can also be set if 8-bit pulse accumulator 3 (PAC3) reaches 0x00FF followed by an active edge on IC3.
0 PAIF	**Pulse Accumulator Input edge Flag** — Set when the selected edge is detected at the IC7 input pin. In event mode the event edge triggers PAIF and in gated time accumulation mode the trailing edge of the gate signal at the IC7 input pin triggers PAIF.

14.3.2.17 Pulse Accumulators Count Registers (PACN3 and PACN2)

Module Base + 0x0022

	7	6	5	4	3	2	1	0
R W	PACNT7(15)	PACNT6(14)	PACNT5(13)	PACNT4(12)	PACNT3(11)	PACNT2(10)	PACNT1(9)	PACNT0(8)
Reset	0	0	0	0	0	0	0	0

Figure 14-38. Pulse Accumulators Count Register 3 (PACN3)

All bits reset to zero.

The two 8-bit pulse accumulators PAC1 and PAC0 are cascaded to form the PACB 16-bit pulse accumulator. When PACB in enabled, (PBEN = 1 in PBCTL) the PACN1 and PACN0 registers contents are respectively the high and low byte of the PACB.

When PACN1 overflows from 0x00FF to 0x0000, the interrupt flag PBOVF in PBFLG is set.

Full count register access will take place in one clock cycle.

NOTE

A separate read/write for high byte and low byte will give a different result than accessing them as a word.

When clocking pulse and write to the registers occurs simultaneously, write takes priority and the register is not incremented.

14.3.2.19 16-Bit Modulus Down-Counter Control Register (MCCTL)

Module Base + 0x0026

	7	6	5	4	3	2	1	0
R	MCZI	MODMC	RDMCL	0	0	MCEN	MCPR1	MCPR0
W				ICLAT	FLMC			
Reset	0	0	0	0	0	0	0	0

Figure 14-42. 16-Bit Modulus Down-Counter Control Register (MCCTL)

Read: Anytime

Write: Anytime

All bits reset to zero.

Table 14-23. MCCTL Field Descriptions

Field	Description
7 MCZI	**Modulus Counter Underflow Interrupt Enable** 0 Modulus counter interrupt is disabled. 1 Modulus counter interrupt is enabled.
6 MODMC	**Modulus Mode Enable** 0 The modulus counter counts down from the value written to it and will stop at 0x0000. 1 Modulus mode is enabled. When the modulus counter reaches 0x0000, the counter is loaded with the latest value written to the modulus count register. **Note:** For proper operation, the MCEN bit should be cleared before modifying the MODMC bit in order to reset the modulus counter to 0xFFFF.
5 RDMCL	**Read Modulus Down-Counter Load** 0 Reads of the modulus count register (MCCNT) will return the present value of the count register. 1 Reads of the modulus count register (MCCNT) will return the contents of the load register.

If the corresponding NOVWx bit of the ICOVW register is set, the capture register or its holding register cannot be written by an event unless they are empty (see Section 14.4.1.1, “IC Channels”). This will prevent the captured value from being overwritten until it is read or latched in the holding register.

2. IC Queue Mode (LATQ = 0)

   The main timer value is memorized in the IC register by a valid input pin transition (see Figure 14-69 and Figure 14-70).

   If the corresponding NOVWx bit of the ICOVW register is cleared, with a new occurrence of a capture, the value of the IC register will be transferred to its holding register and the IC register memorizes the new timer value.

   If the corresponding NOVWx bit of the ICOVW register is set, the capture register or its holding register cannot be written by an event unless they are empty (see Section 14.4.1.1, “IC Channels”).

   if the TFMOD bit of the ICSYS register is set,the timer flags C3F--C0F in TFLG register are set only when a latch on the corresponding holding register occurs,after C3F--C0F are set,user should clear flag C3F--C0F,then read TCx and TCxH to make TCx and TCxH be empty.

   In queue mode, reads of the holding register will latch the corresponding pulse accumulator value to its holding register.

14.4.1.1.3 Delayed IC Channels

There are four delay counters in this module associated with IC channels 0–3. The use of this feature is explained in the diagram and notes below.

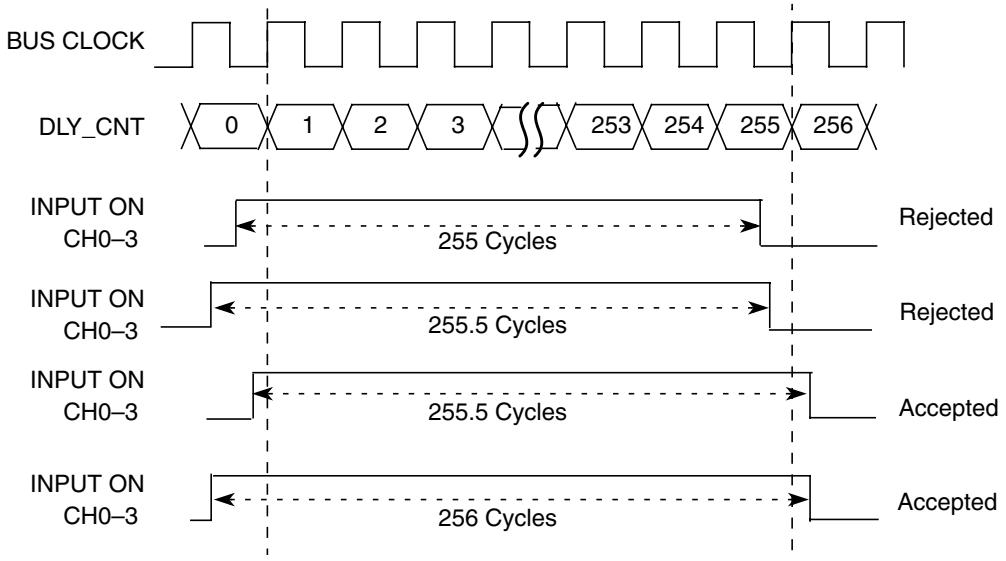

Figure 14-74. Channel Input Validity with Delay Counter Feature

In Figure 14-74 a delay counter value of 256 bus cycles is considered.

1. Input pulses with a duration of (DLY_CNT – 1) cycles or shorter are rejected.
2. Input pulses with a duration between (DLY_CNT – 1) and DLY_CNT cycles may be rejected or accepted, depending on their relative alignment with the sample points.

Table 15-8. IBCR Field Descriptions

Field	Description
7 IBEN	**I-Bus Enable** — This bit controls the software reset of the entire IIC bus module. 0 The module is reset and disabled. This is the power-on reset situation. When low the interface is held in reset but registers can be accessed 1 The IIC bus module is enabled.This bit must be set before any other IBCR bits have any effect If the IIC bus module is enabled in the middle of a byte transfer the interface behaves as follows: slave mode ignores the current transfer on the bus and starts operating whenever a subsequent start condition is detected. Master mode will not be aware that the bus is busy, hence if a start cycle is initiated then the current bus cycle may become corrupt. This would ultimately result in either the current bus master or the IIC bus module losing arbitration, after which bus operation would return to normal.
6 IBIE	**I-Bus Interrupt Enable** 0 Interrupts from the IIC bus module are disabled. Note that this does not clear any currently pending interrupt condition 1 Interrupts from the IIC bus module are enabled. An IIC bus interrupt occurs provided the IBIF bit in the status register is also set.
5 MS/SL	**Master/Slave Mode Select Bit** — Upon reset, this bit is cleared. When this bit is changed from 0 to 1, a START signal is generated on the bus, and the master mode is selected. When this bit is changed from 1 to 0, a STOP signal is generated and the operation mode changes from master to slave.A STOP signal should only be generated if the IBIF flag is set. MS/$\overline{SL}$ is cleared without generating a STOP signal when the master loses arbitration. 0 Slave Mode 1 Master Mode
4 Tx/Rx	**Transmit/Receive Mode Select Bit** — This bit selects the direction of master and slave transfers. When addressed as a slave this bit should be set by software according to the SRW bit in the status register. In master mode this bit should be set according to the type of transfer required. Therefore, for address cycles, this bit will always be high. 0 Receive 1 Transmit
3 TXAK	**Transmit Acknowledge Enable** — This bit specifies the value driven onto SDA during data acknowledge cycles for both master and slave receivers. The IIC module will always acknowledge address matches, provided it is enabled, regardless of the value of TXAK. Note that values written to this bit are only used when the IIC bus is a receiver, not a transmitter. 0 An acknowledge signal will be sent out to the bus at the 9th clock bit after receiving one byte data 1 No acknowledge signal response is sent (i.e., acknowledge bit = 1)
2 RSTA	**Repeat Start** — Writing a 1 to this bit will generate a repeated START condition on the bus, provided it is the current bus master. This bit will always be read as a low. Attempting a repeated start at the wrong time, if the bus is owned by another master, will result in loss of arbitration. 1 Generate repeat start cycle
1 RESERVED	**Reserved** — Bit 1 of the IBCR is reserved for future compatibility. This bit will always read 0.
0 IBSWAI	**I Bus Interface Stop in Wait Mode** 0 IIC bus module clock operates normally 1 Halt IIC bus module clock generation in wait mode

Wait mode is entered via execution of a CPU WAI instruction. In the event that the IBSWAI bit is set, all clocks internal to the IIC will be stopped and any transmission currently in progress will halt.If the CPU were woken up by a source other than the IIC module, then clocks would restart and the IIC would resume from where was during the previous transmission. It is not possible for the IIC to wake up the CPU when its internal clocks are stopped.

Table 16-23. CANIDAR4–CANIDAR7 Register Field Descriptions

Field	Description
7-0 AC[7:0]	**Acceptance Code Bits** — AC[7:0] comprise a user-defined sequence of bits with which the corresponding bits of the related identifier register (IDRn) of the receive message buffer are compared. The result of this comparison is then masked with the corresponding identifier mask register.

16.3.2.18 MSCAN Identifier Mask Registers (CANIDMR0–CANIDMR7)

The identifier mask register specifies which of the corresponding bits in the identifier acceptance register are relevant for acceptance filtering. To receive standard identifiers in 32 bit filter mode, it is required to program the last three bits (AM[2:0]) in the mask registers CANIDMR1 and CANIDMR5 to "don't care." To receive standard identifiers in 16 bit filter mode, it is required to program the last three bits (AM[2:0]) in the mask registers CANIDMR1, CANIDMR3, CANIDMR5, and CANIDMR7 to "don't care."

Module Base + 0x0014 to Module Base + 0x0017 — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset	0	0	0	0	0	0	0	0

Figure 16-22. MSCAN Identifier Mask Registers (First Bank) — CANIDMR0–CANIDMR3

1. Read: Anytime
   Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 16-24. CANIDMR0–CANIDMR3 Register Field Descriptions

Field	Description
7-0 AM[7:0]	**Acceptance Mask Bits** — If a particular bit in this register is cleared, this indicates that the corresponding bit in the identifier acceptance register must be the same as its identifier bit before a match is detected. The message is accepted if all such bits match. If a bit is set, it indicates that the state of the corresponding bit in the identifier acceptance register does not affect whether or not the message is accepted. 0 Match corresponding acceptance code register and identifier bits 1 Ignore corresponding acceptance code register bit

Module Base + 0x001C to Module Base + 0x001F — Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset	0	0	0	0	0	0	0	0

Figure 16-23. MSCAN Identifier Mask Registers (Second Bank) — CANIDMR4–CANIDMR7

The main element of the SPI system is the SPI data register. The n-bit[1] data register in the master and the n-bit[1] data register in the slave are linked by the MOSI and MISO pins to form a distributed 2n-bit[1] register. When a data transfer operation is performed, this 2n-bit[1] register is serially shifted n[1] bit positions by the S-clock from the master, so data is exchanged between the master and the slave. Data written to the master SPI data register becomes the output data for the slave, and data read from the master SPI data register after a transfer operation is the input data from the slave.

A read of SPISR with SPTEF = 1 followed by a write to SPIDR puts data into the transmit data register. When a transfer is complete and SPIF is cleared, received data is moved into the receive data register. This data register acts as the SPI receive data register for reads and as the SPI transmit data register for writes. A common SPI data register address is shared for reading data from the read data buffer and for writing data to the transmit data register.

The clock phase control bit (CPHA) and a clock polarity control bit (CPOL) in the SPI control register 1 (SPICR1) select one of four possible clock formats to be used by the SPI system. The CPOL bit simply selects a non-inverted or inverted clock. The CPHA bit is used to accommodate two fundamentally different protocols by sampling data on odd numbered SCK edges or on even numbered SCK edges (see Section 21.4.3, “Transmission Formats”).

The SPI can be configured to operate as a master or as a slave. When the MSTR bit in SPI control register1 is set, master mode is selected, when the MSTR bit is clear, slave mode is selected.

NOTE

A change of CPOL or MSTR bit while there is a received byte pending in the receive shift register will destroy the received byte and must be avoided.

21.4.1 Master Mode

The SPI operates in master mode when the MSTR bit is set. Only a master SPI module can initiate transmissions. A transmission begins by writing to the master SPI data register. If the shift register is empty, data immediately transfers to the shift register. Data begins shifting out on the MOSI pin under the control of the serial clock.

- Serial clock

  The SPR2, SPR1, and SPR0 baud rate selection bits, in conjunction with the SPPR2, SPPR1, and SPPR0 baud rate preselection bits in the SPI baud rate register, control the baud rate generator and determine the speed of the transmission. The SCK pin is the SPI clock output. Through the SCK pin, the baud rate generator of the master controls the shift register of the slave peripheral.
- MOSI, MISO pin

  In master mode, the function of the serial data output pin (MOSI) and the serial data input pin (MISO) is determined by the SPC0 and BIDIROE control bits.
- $\overline{SS}$ pin

  If MODFEN and SSOE are set, the $\overline{SS}$ pin is configured as slave select output. The $\overline{SS}$ output becomes low during each transmission and is high when the SPI is in idle state.

  If MODFEN is set and SSOE is cleared, the $\overline{SS}$ pin is configured as input for detecting mode fault error. If the $\overline{SS}$ input becomes low this indicates a mode fault error where another master tries to

1. n depends on the selected transfer width, please refer to Section 21.3.2.2, “SPI Control Register 2 (SPICR2)

Valid margin level settings for the Set User Margin Level command are defined in Table 24-56.

Table 24-56. 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 24-57. 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 command not available in current mode (see Table 24-30)
		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 FTM256K2.

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.

24.4.2.13 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 24-58. 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	

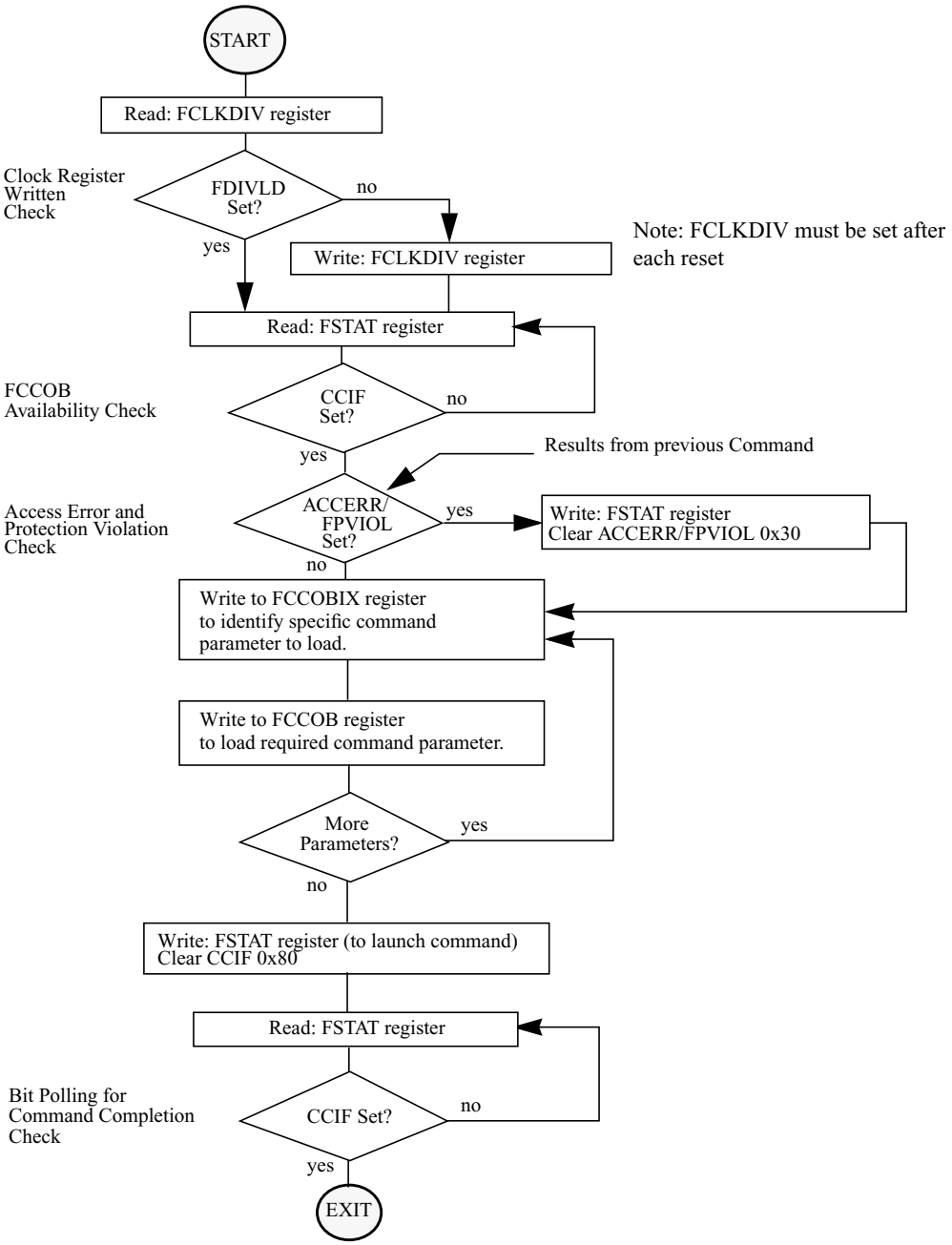

Figure 25-26. Generic Flash Command Write Sequence Flowchart

Valid margin level settings for the Set Field Margin Level command are defined in Table 25-61.

Table 25-61. Valid Set Field 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)
0x0003	Field Margin-1 Level[1]
0x0004	Field Margin-0 Level[2]

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

Table 25-62. Set Field 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 25-30)
		Set if an invalid global address [22:16] is supplied
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

CAUTION

Field margin levels must only be used during verify of the initial factory programming.

NOTE

Field margin levels can be used to check that Flash memory contents have adequate margin for data retention at the normal level setting. If unexpected results are encountered when checking Flash memory contents at field margin levels, the Flash memory contents should be erased and reprogrammed.

25.4.2.15 Full Partition D-Flash Command

The Full Partition D-Flash command allows the user to allocate sectors within the D-Flash block for applications and a partition within the buffer RAM for EEPROM access. The D-Flash block consists of 128 sectors with 256 bytes per sector.

Table 27-64. Full Partition D-Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if a Load Data Field command sequence is currently active
		Set if command not available in current mode (see Table 27-30)
		Set if an invalid DFPART or ERPART selection is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read
FERSTAT	EPVIOLIF	None

27.4.2.16 Erase Verify D-Flash Section Command

The Erase Verify D-Flash Section command will verify that a section of code in the D-Flash user partition is erased. The Erase Verify D-Flash Section command defines the starting point of the data to be verified and the number of words.

Table 27-65. Erase Verify D-Flash Section Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x10	Global address [22:16] to identify the D-Flash block
001	Global address [15:0] of the first word to be verified	
010	Number of words to be verified	

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

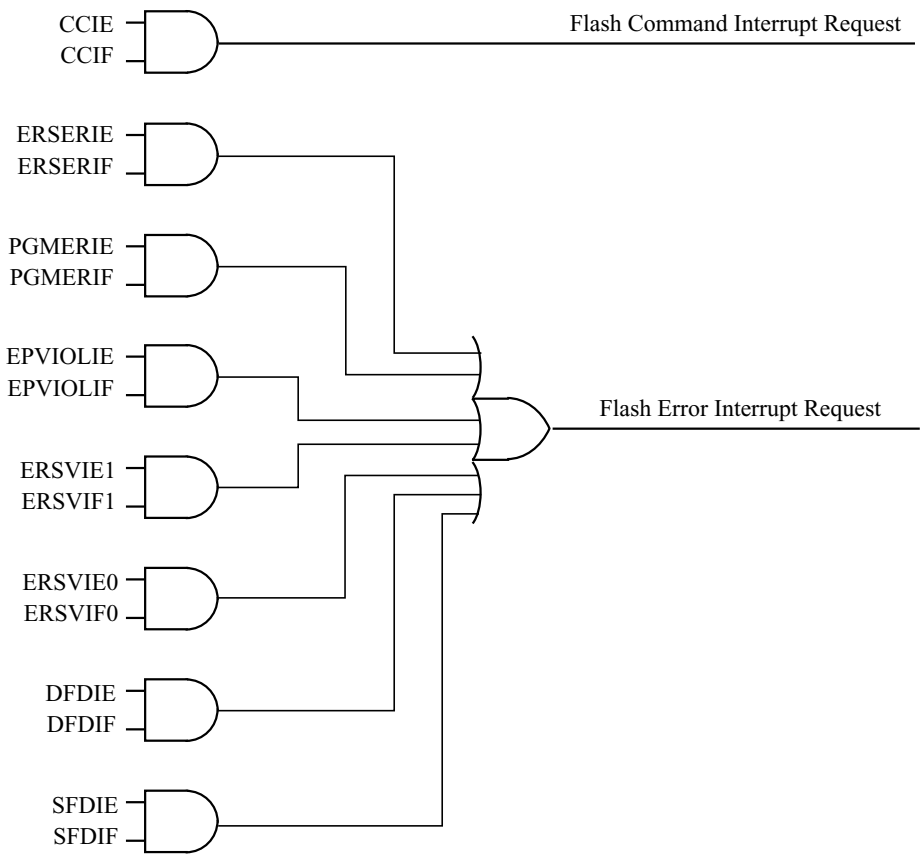

Figure 28-27. Flash Module Interrupts Implementation

28.4.4 Wait Mode

The Flash module is not affected if the MCU enters wait mode. The Flash module can recover the MCU from wait via the CCIF interrupt (see Section 28.4.3, “Interrupts”).

28.4.5 Stop Mode

If a Flash command is active (CCIF = 0) or an EE-Emulation operation is pending when the MCU requests stop mode, the current Flash operation will be completed before the CPU is allowed to enter stop mode.

28.5 Security

The Flash module provides security information to the MCU. The Flash security state is defined by the SEC bits of the FSEC register (see Table 28-12). During reset, the Flash module initializes the FSEC register using data read from the security byte of the Flash configuration field at global address 0x7F_FF0F.

Offset Module Base + 0x000A

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

Figure 29-16. Flash Common Command Object High Register (FCCOBHI)

Offset Module Base + 0x000B

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

Figure 29-17. Flash Common Command Object Low Register (FCCOBLO)

29.3.2.11.1 FCCOB - NVM Command Mode

NVM command mode uses the indexed FCCOB register to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all FCCOB parameter fields are locked and cannot be changed by the user until the command completes (as evidenced by the Memory Controller returning CCIF to 1). Some commands return information to the FCCOB register array.

The generic format for the FCCOB parameter fields in NVM command mode is shown in Table 29-26. The return values are available for reading after the CCIF flag in the FSTAT register has been returned to 1 by the Memory Controller. Writes to the unimplemented parameter fields (CCOBIX = 110 and CCOBIX = 111) are ignored with reads from these fields returning 0x0000.

Table 29-26 shows the generic Flash command format. The high byte of the first word in the CCOB array contains the command code, followed by the parameters for this specific Flash command. For details on the FCCOB settings required by each command, see the Flash command descriptions in Section 29.4.2.

Table 29-26. FCCOB - NVM Command Mode (Typical Usage)

CCOBIX[2:0]	Byte	FCCOB Parameter Fields (NVM Command Mode)
000	HI	FCMD[7:0] defining Flash command
	LO	0, Global address [22:16]
001	HI	Global address [15:8]
	LO	Global address [7:0]
010	HI	Data 0 [15:8]
	LO	Data 0 [7:0]

A.7.3.5 External Tag Trigger Timing

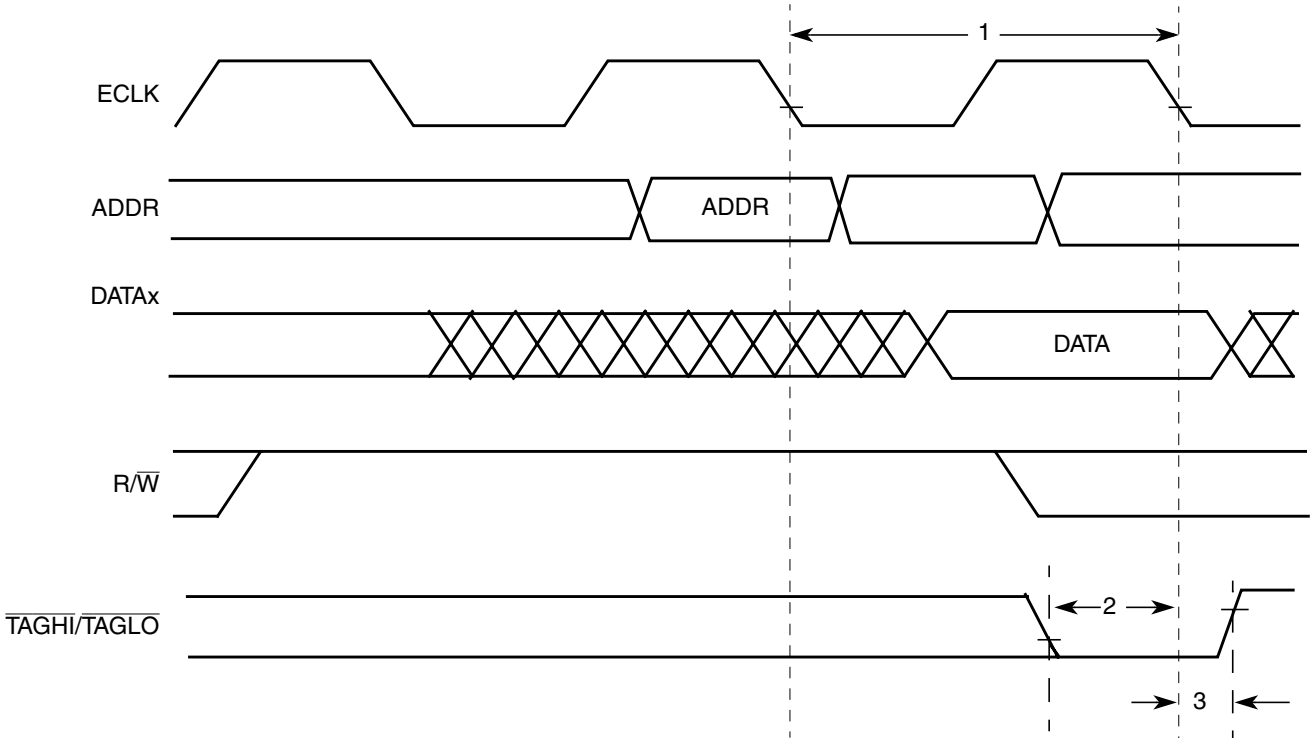

Figure A-18. External Trigger Timing

Table A-34. External Tag Trigger Timing V_{DD35} = 5.0 V

No.	C	Characteristic [1]	Symbol	Min	Max	Unit
-	D	Frequency of internal bus	f_i	D.C.	50.0	MHz
1	D	Cycle time	t_{cyc}	20	∞	ns
2	D	$\overline{TAGHI}/\overline{TAGLO}$ setup time	t_{TS}	10	—	ns
3	D	$\overline{TAGHI}/\overline{TAGLO}$ hold time	t_{TH}	0	—	ns

1. Typical supply and silicon, room temperature only