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	12V1
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	54
Program Memory Size	96KB (96K x 8)
Program Memory Type	FLASH
EEPROM Size	3K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12g96f0clh

2.3.5 Pins PD7-0

Table 2-9. Port D Pins PD7-0

PD7-PD0	• These pins feature general-purpose I/O functionality only.

2.3.6 Pins PE1-0

Table 2-10. Port E Pins PE1-0

PE1	• If the CPMU OSC function is active this pin is used as XTAL signal and the pulldown device is disabled. • 20 TSSOP: The SCI0 TXD signal is mapped to this pin when used with the SCI function. If the SCI0 TXD signal is enabled and routed here the I/O state will depend on the SCI0 configuration. • 20 TSSOP: The TIM channel 3 signal is mapped to this pin when used with the timer function. The TIM forces the I/O state to be an output for a timer port associated with an enabled output compare. • 20 TSSOP: The PWM channel 1 signal is mapped to this pin when used with the PWM function. The enabled PWM channel forces the I/O state to be an output. • 20 TSSOP: The ADC ETRIG1 signal is mapped to this pin when used with the ADC function. The enabled external trigger function has no effect on the I/O state. Refer to Section 2.6.4, “ADC External Triggers ETRIG3-0”. • Signal priority: 20 TSSOP: XTAL > TXD0 > IOC3 > PWM1 > GPO Others: XTAL > GPO
PE0	• If the CPMU OSC function is active this pin is used as EXTAL signal and the pulldown device is disabled. • 20 TSSOP: The SCI0 RXD signal is mapped to this pin when used with the SCI function. If the SCI0 RXD signal is enabled and routed here the I/O state will be forced to input. • 20 TSSOP: The TIM channel 2 signal is mapped to this pin when used with the timer function. The TIM forces the I/O state to be an output for a timer port associated with an enabled output compare. • 20 TSSOP: The PWM channel 0 signal is mapped to this pin when used with the PWM function. The enabled PWM channel forces the I/O state to be an output. • 20 TSSOP: The ADC ETRIG0 signal is mapped to this pin when used with the ADC function. The enabled external trigger function has no effect on the I/O state. Refer to Section 2.6.4, “ADC External Triggers ETRIG3-0”. • Signal priority: 20 TSSOP: EXTAL > RXD0 > IOC2 > PWM0 > GPO Others: EXTAL > GPO

2.3.7 Pins PT7-0

Table 2-11. Port T Pins PT7-0

PT7-PT6	• 64/100 LQFP: The TIM channels 7 and 6 signal are mapped to these pins when used with the timer function. The TIM forces the I/O state to be an output for a timer port associated with an enabled output compare. • Signal priority: 64/100 LQFP: IOC7-6 > GPO
PT5	• 48/64/100 LQFP: The TIM channel 5 signal is mapped to this pin when used with the timer function. The TIM forces the I/O state to be an output for a timer port associated with an enabled output compare. If the ACMP timer link is enabled this pin is disconnected from the timer input so that it can still be used as general-purpose I/O or as timer output. The use case for the ACMP timer link requires the timer input capture function to be enabled. • Signal priority: 48/64/100 LQFP: IOC5 > GPO

Table 5-5. 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. These register bits form bits [15:8] of the local address (see Figure 5-6).

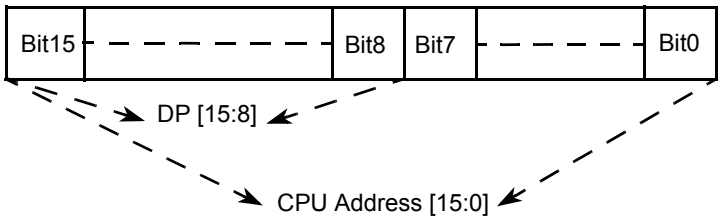

Figure 5-6. DIRECT Address Mapping

Example 5-1. This example demonstrates usage of the Direct Addressing Mode

```
MOVB     #$04,DIRECT       ;Set DIRECT register to 0x04. From this point on, all memory
                           ;accesses using direct addressing mode will be in the local
                           ;address range from 0x0400 to 0x04FF.
LDY      <$12              ;Load the Y index register from 0x0412 (direct access).
```

5.3.2.3 MMC Control Register (MMCCTL1)

Address: 0x0013

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	NVMRES
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 5-7. MMC Control Register (MMCCTL1)

Read: Anytime.

Write: Anytime.

The NVMRES bit maps 16k of internal NVM resources (see Section FTMRG) to the global address space 0x04000 to 0x07FFF.

Table 5-6. MODE Field Descriptions

Field	Description
0 NVMRES	Map internal NVM resources into the global memory map Write: Anytime This bit maps internal NVM resources into the global address space. 0 Program flash is mapped to the global address range from 0x04000 to 0x07FFF. 1 NVM resources are mapped to the global address range from 0x04000 to 0x07FFF.

Table 8-20. State3 — Sequencer Next State Selection

SC[3:0]	Description (Unspecified matches have no effect)
0001	Match2 to State2..... Match1 to Final State
0010	Match0 to Final State..... Match1 to State1
0011	Match1 to Final State..... Match2 to State1
0100	Match1 to State2
0101	Match1 to Final State
0110	Match2 to State2..... Match0 to Final State
0111	Match0 to Final State
1000	Reserved
1001	Reserved
1010	Either Match1 or Match2 to State1..... Match0 to Final State
1011	Reserved
1100	Reserved
1101	Either Match1 or Match2 to Final State..... Match0 to State1
1110	Match0 to State2..... Match2 to Final State
1111	Reserved

The priorities described in Table 8-36 dictate that in the case of simultaneous matches, a match leading to final state has priority followed by the match on the lower channel number (0,1,2).

8.3.2.7.4 Debug Match Flag Register (DBGMFR)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	MC2	MC1	MC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 8-12. Debug Match Flag Register (DBGMFR)

Read: If COMRV[1:0] = 11

Write: Never

DBGMFR is visible at 0x0027 only with COMRV[1:0] = 11. It features 3 flag bits each mapped directly to a channel. Should a match occur on the channel during the debug session, then the corresponding flag is set and remains set until the next time the module is armed by writing to the ARM bit. Thus the contents are retained after a debug session for evaluation purposes. These flags cannot be cleared by software, they are cleared only when arming the module. A set flag does not inhibit the setting of other flags. Once a flag is set, further comparator matches on the same channel in the same session have no affect on that flag.

8.3.2.8 Comparator Register Descriptions

Each comparator has a bank of registers that are visible through an 8-byte window in the DBG module register address map. Comparator A consists of 8 register bytes (3 address bus compare registers, two data bus compare registers, two data bus mask registers and a control register). Comparator B consists of four

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x003B	CPMURTI	R	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
		W								
0x003C	CPMUCOP	R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
		W			WRTMASK					
0x003D	RESERVEDCPMUTEST0	R	0	0	0	0	0	0	0	0
		W								
0x003E	RESERVEDCPMUTEST1	R	0	0	0	0	0	0	0	0
		W								
0x003F	CPMU ARMCOP	R	0	0	0	0	0	0	0	0
		W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02F0	RESERVED	R	0	0	0	0	0	0	0	0
		W								
0x02F1	CPMU LVCTL	R	0	0	0	0	0	LVDS	LVIE	LVIF
		W								
0x02F2	CPMU APICTL	R	APICLK	0	0	APIES	APIEA	APIFE	APIE	APIF
		W								
0x02F3	CPMUACLKTR	R	ACLKTR5	ACLKTR4	ACLKTR3	ACLKTR2	ACLKTR1	ACLKTR0	0	0
		W								
0x02F4	CPMUAPIRH	R	APIR15	APIR14	APIR13	APIR12	APIR11	APIR10	APIR9	APIR8
		W								
0x02F5	CPMUAPIRL	R	APIR7	APIR6	APIR5	APIR4	APIR3	APIR2	APIR1	APIR0
		W								
0x02F6	RESERVEDCPMUTEST3	R	0	0	0	0	0	0	0	0
		W								
0x02F7	RESERVED	R	0	0	0	0	0	0	0	0
		W								
0x02F8	CPMU IRCTRIMH	R	TCTRIM[4:0]					0	IRCTRIM[9:8]	
		W								
0x02F9	CPMU IRCTRIML	R	IRCTRIM[7:0]							
		W								
0x02FA	CPMUOSC	R	OSCE	Reserved	OSCPINS_EN	Reserved				
		W								
0x02FB	CPMUPROT	R	0	0	0	0	0	0	0	PROT
		W								
0x02FC	RESERVEDCPMUTEST2	R	0	0	0	0	0	0	0	0
		W								

= Unimplemented or Reserved

Figure 10-3. CPMU Register Summary

18.1.1 Glossary

Table 18-2. Terminology

ACK	Acknowledge of CAN message
CAN	Controller Area Network
CRC	Cyclic Redundancy Code
EOF	End of Frame
FIFO	First-In-First-Out Memory
IFS	Inter-Frame Sequence
SOF	Start of Frame
CPU bus	CPU related read/write data bus
CAN bus	CAN protocol related serial bus
oscillator clock	Direct clock from external oscillator
bus clock	CPU bus related clock
CAN clock	CAN protocol related clock

18.1.2 Block Diagram

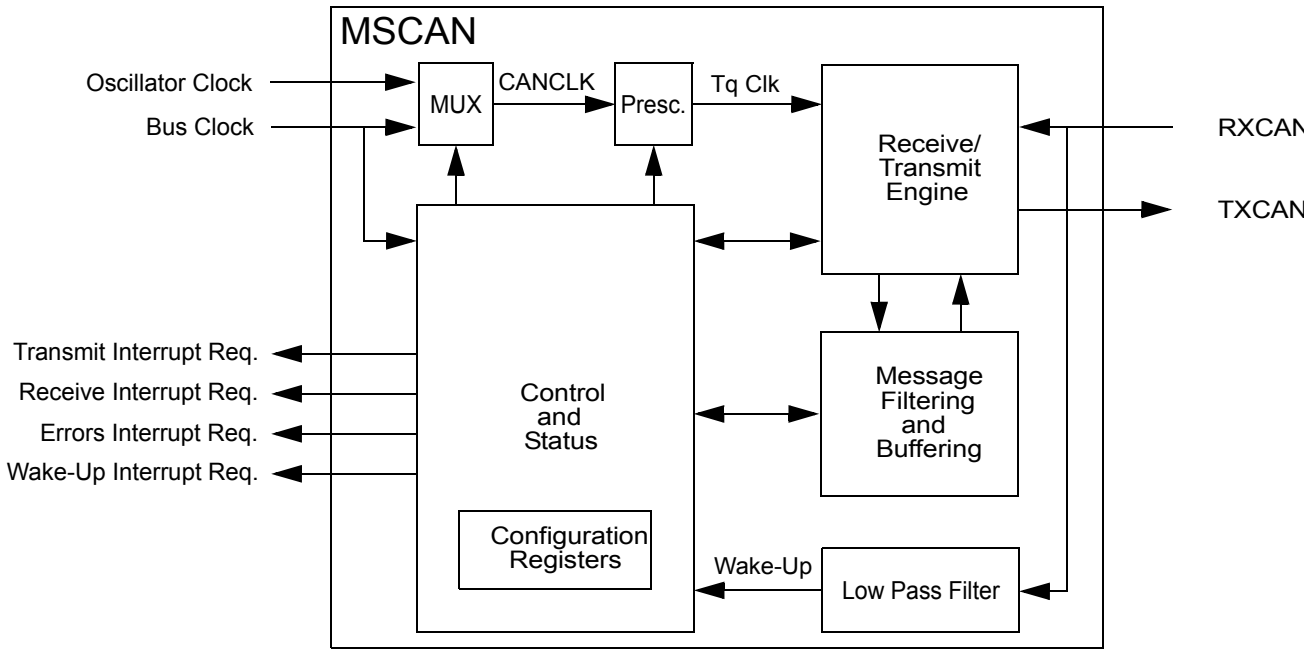

Figure 18-1. MSCAN Block Diagram

message in its RxBG (wrong identifier, transmission errors, etc.) the actual contents of the buffer will be over-written by the next message. The buffer will then not be shifted into the FIFO.

When the MSCAN module is transmitting, the MSCAN receives its own transmitted messages into the background receive buffer, RxBG, but does not shift it into the receiver FIFO, generate a receive interrupt, or acknowledge its own messages on the CAN bus. The exception to this rule is in loopback mode (see Section 18.3.2.2, “MSCAN Control Register 1 (CANCTL1)”) where the MSCAN treats its own messages exactly like all other incoming messages. The MSCAN receives its own transmitted messages in the event that it loses arbitration. If arbitration is lost, the MSCAN must be prepared to become a receiver.

An overrun condition occurs when all receive message buffers in the FIFO are filled with correctly received messages with accepted identifiers and another message is correctly received from the CAN bus with an accepted identifier. The latter message is discarded and an error interrupt with overrun indication is generated if enabled (see Section 18.4.7.5, “Error Interrupt”). The MSCAN remains able to transmit messages while the receiver FIFO is being filled, but all incoming messages are discarded. As soon as a receive buffer in the FIFO is available again, new valid messages will be accepted.

18.4.3 Identifier Acceptance Filter

The MSCAN identifier acceptance registers (see Section 18.3.2.12, “MSCAN Identifier Acceptance Control Register (CANIDAC)”) define the acceptable patterns of the standard or extended identifier (ID[10:0] or ID[28:0]). Any of these bits can be marked ‘don’t care’ in the MSCAN identifier mask registers (see Section 18.3.2.18, “MSCAN Identifier Mask Registers (CANIDMR0–CANIDMR7)”).

A filter hit is indicated to the application software by a set receive buffer full flag (RXF = 1) and three bits in the CANIDAC register (see Section 18.3.2.12, “MSCAN Identifier Acceptance Control Register (CANIDAC)”). These identifier hit flags (IDHIT[2:0]) clearly identify the filter section that caused the acceptance. They simplify the application software’s task to identify the cause of the receiver interrupt. If more than one hit occurs (two or more filters match), the lower hit has priority.

A very flexible programmable generic identifier acceptance filter has been introduced to reduce the CPU interrupt loading. The filter is programmable to operate in four different modes:

- Two identifier acceptance filters, each to be applied to:
  - The full 29 bits of the extended identifier and to the following bits of the CAN 2.0B frame:
    - Remote transmission request (RTR)
    - Identifier extension (IDE)
    - Substitute remote request (SRR)
  - The 11 bits of the standard identifier plus the RTR and IDE bits of the CAN 2.0A/B messages. This mode implements two filters for a full length CAN 2.0B compliant extended identifier. Although this mode can be used for standard identifiers, it is recommended to use the four or eight identifier acceptance filters.
    Figure 18-40 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces a filter 0 hit. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces a filter 1 hit.
- Four identifier acceptance filters, each to be applied to:

22.3.2.3 Timer Count Register (TCNT)

Module Base + 0x0004

	15	14	13	12	11	10	9	9
R W	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8
Reset	0	0	0	0	0	0	0	0

Figure 22-6. Timer Count Register High (TCNTH)

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R W	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
Reset	0	0	0	0	0	0	0	0

Figure 22-7. Timer Count Register Low (TCNTL)

The 16-bit main timer is an up counter.

A full access for the counter register should take place in one clock cycle. A separate read/write for high byte and low byte will give a different result than accessing them as a word.

Read: Anytime

Write: Has no meaning or effect in the normal mode; only writable in special modes (test_mode = 1).

The period of the first count after a write to the TCNT registers may be a different size because the write is not synchronized with the prescaler clock.

22.3.2.4 Timer System Control Register 1 (TSCR1)

Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	TEN	TSWAI	TSFRZ	TFFCA	PRNT	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 22-8. Timer System Control Register 1 (TSCR1)

Read: Anytime

Write: Anytime

Any access to TCNT will clear TFLG2 register if the TFFCA bit in TSCR register is set.

Table 23-17. TRLG2 Field Descriptions

Field	Description
7 TOF	**Timer Overflow Flag** — Set when 16-bit free-running timer overflows from 0xFFFF to 0x0000. Clearing this bit requires writing a one to bit 7 of TFLG2 register while the TEN bit of TSCR1 or PAEN bit of PACTL is set to one (See also TCRE control bit explanation) .

23.3.2.14 Timer Input Capture/Output Compare Registers High and Low 0–7(TCxH and TCxL)

Module Base + 0x0010 = TC0H 0x0018=TC4H
0x0012 = TC1H 0x001A=TC5H
0x0014=TC2H 0x001C=TC6H
0x0016=TC3H 0x001E=TC7H

	15	14	13	12	11	10	9	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset	0	0	0	0	0	0	0	0

Figure 23-22. Timer Input Capture/Output Compare Register x High (TCxH)

Module Base + 0x0011 = TC0L 0x0019 =TC4L
0x0013 = TC1L 0x001B=TC5L
0x0015 =TC2L 0x001D=TC6L
0x0017=TC3L 0x001F=TC7L

	7	6	5	4	3	2	1	0
R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	0	0	0	0	0	0	0	0

Figure 23-23. Timer Input Capture/Output Compare Register x Low (TCxL)

[1] This register is available only when the corresponding channel exists and is reserved if that channel does not exist. Writes to a reserved register have no functional effect. Reads from a reserved register return zeroes.

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Read: Anytime

Write: Anytime for output compare function.Writes to these registers have no meaning or effect during input capture. All timer input capture/output compare registers are reset to 0x0000.

NOTE

Read/Write access in byte mode for high byte should take place before low byte otherwise it will give a different result.

8. Reset the MCU

24.5.3 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 24-25.

24.6 Initialization

On each system reset the flash module executes an initialization sequence which establishes initial values for the Flash Block Configuration Parameters, the FPROT and EEPROT protection registers, and the FOPT and FSEC registers. The initialization routine 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.

CCIF is cleared throughout the initialization sequence. The Flash module holds off all CPU access for a portion of the initialization sequence. Flash reads are allowed once the hold is removed. Completion of the initialization sequence is marked by setting CCIF high which enables user commands.

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.

The user code stored in the P-Flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 25.3.2.2), the MCU can be unsecured by the backdoor key access sequence described below:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Section 25.4.6.11
2. If the Verify Backdoor Access Key command is successful, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 10

The Verify Backdoor Access Key command is monitored by the Memory Controller and an illegal key will prohibit future use of the Verify Backdoor Access Key command. A reset of the MCU is the only method to re-enable the Verify Backdoor Access Key command. The security as defined in the Flash security byte (0x3_FF0F) is not changed by using the Verify Backdoor Access Key command sequence. The backdoor keys stored in addresses 0x3_FF00-0x3_FF07 are unaffected by the Verify Backdoor Access Key command sequence. The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the Flash protection register, FPROT.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the sector containing the Flash security byte can be erased and the Flash security byte can be reprogrammed to the unsecure state, if desired. In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0x3_FF00-0x3_FF07 in the Flash configuration field.

25.5.2 Unsecuring the MCU in Special Single Chip Mode using BDM

A secured MCU can be unsecured in special single chip mode by using the following method to erase the P-Flash and EEPROM memory:

1. Reset the MCU into special single chip mode
2. Delay while the BDM executes the Erase Verify All Blocks command write sequence to check if the P-Flash and EEPROM memories are erased
3. Send BDM commands to disable protection in the P-Flash and EEPROM memory
4. Execute the Erase All Blocks command write sequence to erase the P-Flash and EEPROM memory. Alternatively the Unsecure Flash command can be executed, if so the steps 5 and 6 below are skeeped.
5. After the CCIF flag sets to indicate that the Erase All Blocks operation has completed, reset the MCU into special single chip mode
6. Delay while the BDM executes the Erase Verify All Blocks command write sequence to verify that the P-Flash and EEPROM memory are erased

If the P-Flash and EEPROM memory are verified as erased, the MCU will be unsecured. All BDM commands will now be enabled and the Flash security byte may be programmed to the unsecure state by continuing with the following steps:

7. Send BDM commands to execute the Program P-Flash command write sequence to program the Flash security byte to the unsecured state

26.4.6.2 Erase Verify Block Command

The Erase Verify Block command allows the user to verify that an entire P-Flash or EEPROM block has been erased. The FCCOB FlashBlockSelectionCode[1:0] bits determine which block must be verified.

Table 26-33. Erase Verify Block Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x02	Flash block selection code [1:0]. See Table 26-34

Table 26-34. Flash block selection code description

Selection code[1:0]	Flash block to be verified
00	EEPROM
01	Invalid (ACCERR)
10	Invalid (ACCERR)
11	P-Flash

Upon clearing CCIF to launch the Erase Verify Block command, the Memory Controller will verify that the selected P-Flash or EEPROM block is erased. The CCIF flag will set after the Erase Verify Block operation has completed.If the block is not erased, it means blank check failed, both MGSTAT bits will be set.

Table 26-35. Erase Verify Block Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if an invalid FlashBlockSelectionCode[1:0] is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read or if blank check failed.

26.4.6.3 Erase Verify P-Flash Section Command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases.

Table 26-61. Erase Verify EEPROM Section Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 010 at command launch
		Set if command not available in current mode (see Table 26-27)
		Set if an invalid global address [17:0] is supplied
		Set if a misaligned word address is supplied (global address [0] != 0)
		Set if the requested section breaches the end of the EEPROM block
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read or if blank check failed.

26.4.6.15 Program EEPROM Command

The Program EEPROM operation programs one to four previously erased words in the EEPROM block. The Program EEPROM operation will confirm that the targeted location(s) were successfully programmed upon completion.

CAUTION

A Flash word must be in the erased state before being programmed. Cumulative programming of bits within a Flash word is not allowed.

Table 26-62. Program EEPROM Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x11	Global address [17:16] to identify the EEPROM block
001	Global address [15:0] of word to be programmed	
010	Word 0 program value	
011	Word 1 program value, if desired	
100	Word 2 program value, if desired	
101	Word 3 program value, if desired	

Upon clearing CCIF to launch the Program EEPROM command, the user-supplied words will be transferred to the Memory Controller and be programmed if the area is unprotected. The CCOBIX index value at Program EEPROM command launch determines how many words will be programmed in the EEPROM block. The CCIF flag is set when the operation has completed.

26.4.7 Interrupts

The Flash module can generate an interrupt when a Flash command operation has completed or when a Flash command operation has detected an ECC fault.

Table 26-66. Flash Interrupt Sources

Interrupt Source	Interrupt Flag	Local Enable	Global (CCR) Mask
Flash Command Complete	CCIF (FSTAT register)	CCIE (FCNFG register)	I Bit
ECC Double Bit Fault on Flash Read	DFDIF (FERSTAT register)	DFDIE (FERCNFG register)	I Bit
ECC Single Bit Fault on Flash Read	SFDIF (FERSTAT register)	SFDIE (FERCNFG register)	I Bit

NOTE

Vector addresses and their relative interrupt priority are determined at the MCU level.

26.4.7.1 Description of Flash Interrupt Operation

The Flash module uses the CCIF flag in combination with the CCIE interrupt enable bit to generate the Flash command interrupt request. The Flash module uses the DFDIF and SFDIF flags in combination with the DFDIE and SFDIE interrupt enable bits to generate the Flash error interrupt request. For a detailed description of the register bits involved, refer to Section 26.3.2.5, “Flash Configuration Register (FCNFG)”, Section 26.3.2.6, “Flash Error Configuration Register (FERCNFG)”, Section 26.3.2.7, “Flash Status Register (FSTAT)”, and Section 26.3.2.8, “Flash Error Status Register (FERSTAT)”.

The logic used for generating the Flash module interrupts is shown in Figure 26-27.

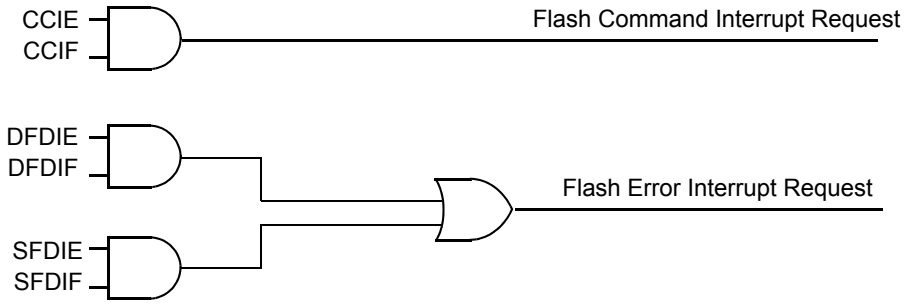

Figure 26-27. Flash Module Interrupts Implementation

28.3.2.5 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.

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 28-9. Flash Configuration Register (FCNFG)

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

Table 28-13. 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 28.3.2.7)
4 IGNSF	**Ignore Single Bit Fault** — The IGNSF controls single bit fault reporting in the FERSTAT register (see Section 28.3.2.8). 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
1 FDFD	**Force Double Bit Fault Detect** — The FDFD bit allows the user to simulate a double bit fault during Flash array read operations and check the associated interrupt routine. The FDFD bit is cleared by writing a 0 to FDFD. 0 Flash array read operations will set the DFDIF flag in the FERSTAT register only if a double bit fault is detected 1 Any Flash array read operation will force the DFDIF flag in the FERSTAT register to be set (see Section 28.3.2.7) and an interrupt will be generated as long as the DFDIE interrupt enable in the FERCNFG register is set (see Section 28.3.2.6)
0 FSFD	**Force Single Bit Fault Detect** — The FSFD bit allows the user to simulate a single bit fault during Flash array read operations and check the associated interrupt routine. The FSFD bit is cleared by writing a 0 to FSFD. 0 Flash array read operations will set the SFDIF flag in the FERSTAT register only if a single bit fault is detected 1 Flash array read operation will force the SFDIF flag in the FERSTAT register to be set (see Section 28.3.2.7) and an interrupt will be generated as long as the SFDIE interrupt enable in the FERCNFG register is set (see Section 28.3.2.6)

28.3.2.6 Flash Error Configuration Register (FERCNFG)

The FERCNFG register enables the Flash error interrupts for the FERSTAT flags.

29.4.6.2 Erase Verify Block Command

The Erase Verify Block command allows the user to verify that an entire P-Flash or EEPROM block has been erased. The FCCOB FlashBlockSelectionCode[1:0] bits determine which block must be verified.

Table 29-33. Erase Verify Block Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x02	Flash block selection code [1:0]. See Table 29-34

Table 29-34. Flash block selection code description

Selection code[1:0]	Flash block to be verified
00	EEPROM
01	Invalid (ACCERR)
10	P-Flash
11	P-Flash

Upon clearing CCIF to launch the Erase Verify Block command, the Memory Controller will verify that the selected P-Flash or EEPROM block is erased. The CCIF flag will set after the Erase Verify Block operation has completed.If the block is not erased, it means blank check failed, both MGSTAT bits will be set.

Table 29-35. Erase Verify Block Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if an invalid FlashBlockSelectionCode[1:0] is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read or if blank check failed.

29.4.6.3 Erase Verify P-Flash Section Command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases.

31.4.6.15 Program EEPROM Command

The Program EEPROM operation programs one to four previously erased words in the EEPROM block. The Program EEPROM operation will confirm that the targeted location(s) were successfully programmed upon completion.

CAUTION

A Flash word must be in the erased state before being programmed. Cumulative programming of bits within a Flash word is not allowed.

Table 31-62. Program EEPROM Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x11	Global address [17:16] to identify the EEPROM block
001	Global address [15:0] of word to be programmed	
010	Word 0 program value	
011	Word 1 program value, if desired	
100	Word 2 program value, if desired	
101	Word 3 program value, if desired	

Upon clearing CCIF to launch the Program EEPROM command, the user-supplied words will be transferred to the Memory Controller and be programmed if the area is unprotected. The CCOBIX index value at Program EEPROM command launch determines how many words will be programmed in the EEPROM block. The CCIF flag is set when the operation has completed.

Table 31-63. Program EEPROM 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 31-27)
		Set if an invalid global address [17:0] is supplied
		Set if a misaligned word address is supplied (global address [0] != 0)
		Set if the requested group of words breaches the end of the EEPROM block
	FPVIOL	Set if the selected area of the EEPROM 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

31.4.6.16 Erase EEPROM Sector Command

The Erase EEPROM Sector operation will erase all addresses in a sector of the EEPROM block.

reprogrammed to the unsecure state, if desired. In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0x3_FF00-0x3_FF07 in the Flash configuration field.

31.5.2 Unsecuring the MCU in Special Single Chip Mode using BDM

A secured MCU can be unsecured in special single chip mode by using the following method to erase the P-Flash and EEPROM memory:

1. Reset the MCU into special single chip mode
2. Delay while the BDM executes the Erase Verify All Blocks command write sequence to check if the P-Flash and EEPROM memories are erased
3. Send BDM commands to disable protection in the P-Flash and EEPROM memory
4. Execute the Erase All Blocks command write sequence to erase the P-Flash and EEPROM memory. Alternatively the Unsecure Flash command can be executed, if so the steps 5 and 6 below are skeeped.
5. After the CCIF flag sets to indicate that the Erase All Blocks operation has completed, reset the MCU into special single chip mode
6. Delay while the BDM executes the Erase Verify All Blocks command write sequence to verify that the P-Flash and EEPROM memory are erased

If the P-Flash and EEPROM memory are verified as erased, the MCU will be unsecured. All BDM commands will now be enabled and the Flash security byte may be programmed to the unsecure state by continuing with the following steps:

7. Send BDM commands to execute the Program P-Flash command write sequence to program the Flash security byte to the unsecured state
8. Reset the MCU

31.5.3 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 31-27.

31.6 Initialization

On each system reset the flash module executes an initialization sequence which establishes initial values for the Flash Block Configuration Parameters, the FPROT and EEPROT protection registers, and the FOPT and FSEC registers. The initialization routine 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.

CCIF is cleared throughout the initialization sequence. The Flash module holds off all CPU access for a portion of the initialization sequence. Flash reads are allowed once the hold is removed. Completion of the initialization sequence is marked by setting CCIF high which enables user commands.

Table A-17. Full Stop Current Characteristics

Conditions are: Typ: V_{DDX},V_{DDR},V_{DDA}=5V, Max: V_{DDX},V_{DDR},V_{DDA}=5.5V API see Table A-13.							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
S12GN16, S12GN32							
Stop Current API disabled							
1	P	-40°C	I_{DDS}		14.4	24	µA
2	P	25°C	I_{DDS}		16.5	28	µA
3	P	150°C	I_{DDS}		120	320	µA
4	C	160°C	I_{DDS}		140		µA
Stop Current API enabled							
5	C	-40°C	I_{DDS}		18.5		µA
6	C	25°C	I_{DDS}		21.5		µA
7	C	150°C	I_{DDS}		130		µA
8	C	160°C	I_{DDS}		150		µA
S12GN48, S12G48, S12G64							
Stop Current API disabled							
9	P	-40°C	I_{DDS}		16	27	µA
10	P	25°C	I_{DDS}		18.5	30	µA
11	P	150°C	I_{DDS}		140	370	µA
Stop Current API enabled							
12	C	-40°C	I_{DDS}		20		µA
13	C	25°C	I_{DDS}		23.5		µA
14	C	150°C	I_{DDS}		150		µA
S12G96, S12G128							
Stop Current API disabled							
15	P	-40°C	I_{DDS}		16.5	28	µA
16	P	25°C	I_{DDS}		19	32	µA
17	P	150°C	I_{DDS}		150	400	µA
Stop Current API enabled							
18	C	-40°C	I_{DDS}		20.5		µA
19	C	25°C	I_{DDS}		24		µA
20	C	150°C	I_{DDS}		160		µA
S12G192, S12GA192, S12G240, S12GA240							
Stop Current API disabled							
21	P	-40°C	I_{DDS}		17	30	µA
22	P	25°C	I_{DDS}		19.5	34	µA
23	P	150°C	I_{DDS}		155	420	µA
Stop Current API enabled							
24	C	-40°C	I_{DDS}		21		µA
25	C	25°C	I_{DDS}		24.5		µA
26	C	150°C	I_{DDS}		160		µA

A.10 Electrical Characteristics for the Oscillator (XOSCLCP)

Table A-44. XOSCLCP Characteristics (Junction Temperature From –40°C To +150°C)

Conditions are shown in Table A-4 unless otherwise noted							
Num	C	Rating	Symbol	Min	Typ	Max	Unit
1	C	Nominal crystal or resonator frequency	f_{OSC}	4.0		16	MHz
2	P	Startup Current	i_{OSC}	100			µA
3a	C	Oscillator start-up time (4MHz)[1]	t_{UPOSC}	—	2	10	ms
3b	C	Oscillator start-up time (8MHz)[1]	t_{UPOSC}	—	1.6	8	ms
3c	C	Oscillator start-up time (16MHz)[1]	t_{UPOSC}	—	1	5	ms
4	P	Clock Monitor Failure Assert Frequency	f_{CMFA}	200	450	1200	KHz
5	D	Input Capacitance (EXTAL, XTAL pins)	C_{IN}		7		pF
6	C	EXTAL Pin Input Hysteresis	$V_{HYS,EXTAL}$	—	120	—	mV
7	C	EXTAL Pin oscillation amplitude (loop controlled Pierce) all mask sets except for 2N75C and 2N55V	$V_{PP,EXTAL}$	—	1.0	—	V
8	D	EXTAL Pin oscillation required amplitude[2] (loop controlled Pierce) all mask sets except for 2N75C and 2N55V	$V_{PP,EXTAL}$	0.8	—	1.5	V

1 These values apply for carefully designed PCB layouts with capacitors that match the crystal/resonator requirements.

2 Needs to be measured at room temperature on the application board using a probe with very low (<=5pF) input capacitance.