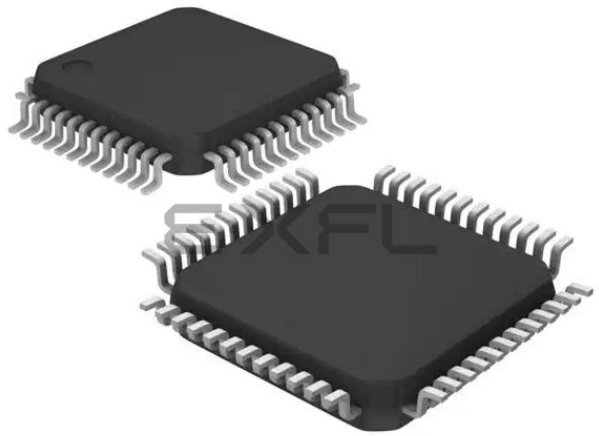


Welcome to E-XFL.COM

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	IrDA, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	40
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.13V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12gn32f0clf

1.3.1 S12 16-Bit Central Processor Unit (CPU)

S12 CPU is a high-speed 16-bit processing unit:

- Full 16-bit data paths supports efficient arithmetic operation and high-speed math execution
- Includes many single-byte instructions. This allows much more efficient use of ROM space.
- Extensive set of indexed addressing capabilities, including:
  - Using the stack pointer as an indexing register in all indexed operations
  - Using the program counter as an indexing register in all but auto increment/decrement mode
  - Accumulator offsets using A, B, or D accumulators
  - Automatic index predecrement, preincrement, postdecrement, and postincrement (by –8 to +8)

1.3.2 On-Chip Flash with ECC

On-chip flash memory on the MC9S12G-Family family features the following:

- Up to 240 Kbyte of program flash memory
  - 32 data bits plus 7 syndrome ECC (error correction code) bits allow single bit error correction and double fault detection
  - Erase sector size 512 bytes
  - Automated program and erase algorithm
  - User margin level setting for reads
  - Protection scheme to prevent accidental program or erase
- Up to 4 Kbyte EEPROM
  - 16 data bits plus 6 syndrome ECC (error correction code) bits allow single bit error correction and double fault detection
  - Erase sector size 4 bytes
  - Automated program and erase algorithm
  - User margin level setting for reads

1.3.3 On-Chip SRAM

- Up to 11 Kbytes of general-purpose RAM

1.3.4 Port Integration Module (PIM)

- Data registers and data direction registers for ports A, B, C, D, E, T, S, M, P, J and AD when used as general-purpose I/O
- Control registers to enable/disable pull devices and select pullups/pulldowns on ports T, S, M, P, J and AD on per-pin basis
- Single control register to enable/disable pull devices on ports A, B, C, D and E, on per-port basis and on BKGD pin
- Control registers to enable/disable open-drain (wired-or) mode on ports S and M

Table 2-12. Port S Pins PS7-0 (continued)

PS4	• The SPI0 MISO signal is mapped to this pin when used with the SPI function. Depending on the configuration of the enabled SPI0 the I/O state is forced to be input or output. • 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 PWM channel 2 signal is mapped to this pin when used with the PWM function. If the PWM channel is enabled and routed here the I/O state is forced to output. • 32 LQFP: The PWM channel 4 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 ETRIG2 signal is mapped to this pin if PWM channel 2 is routed here. 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: MISO0 > RXD0 > PWM2 > GPO 32 LQFP: MISO0 > PWM4 > GPO Others: MISO0 > GPO
PS3	• Except 20 TSSOP and 32 LQFP: The SCI1 TXD signal is mapped to this pin when used with the SCI function. If the SCI1 TXD signal is enabled the I/O state will depend on the SCI1 configuration. • Signal priority: 48/64/100 LQFP: TXD1 > GPO
PS2	• Except 20 TSSOP and 32 LQFP: The SCI1 RXD signal is mapped to this pin when used with the SCI function. If the SCI1 RXD signal is enabled the I/O state will be forced to be input. • Signal priority: 20 TSSOP and 32 LQFP: GPO Others: RXD1 > GPO
PS1	• Except 20 TSSOP: The SCI0 TXD signal is mapped to this pin when used with the SCI function. If the SCI0 TXD signal is enabled the I/O state will depend on the SCI0 configuration. • Signal priority: Except 20 TSSOP: TXD0 > GPO
PS0	• Except 20 TSSOP: The SCI0 RXD signal is mapped to this pin when used with the SCI function. If the SCI0 RXD signal is enabled the I/O state will be forced to be input. • Signal priority: 20 TSSOP: GPO Others: RXD0 > GPO

Table 2-21. Block Register Map (G3) (continued)

Global Address Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0249 PTIS	R	PTIS7	PTIS6	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0
	W								
0x024A DDRS	R	DDRS7	DDRS6	DDRS5	DDRS4	DDRS3	DDRS2	DDRS1	DDRS0
	W								
0x024B Reserved	R	0	0	0	0	0	0	0	0
	W								
0x024C PERS	R	PERS7	PERS6	PERS5	PERS4	PERS3	PERS2	PERS1	PERS0
	W								
0x024D PPSS	R	PPSS7	PPSS6	PPSS5	PPSS4	PPSS3	PPSS2	PPSS1	PPSS0
	W								
0x024E WOMS	R	WOMS7	WOMS6	WOMS5	WOMS4	WOMS3	WOMS2	WOMS1	WOMS0
	W								
0x024F PRR0	R	PRR0P3	PRR0P2	PRR0T31	PRR0T30	PRR0T21	PRR0T20	PRR0S1	PRR0S0
	W								
0x0250 PTM	R	0	0	0	0	0	0	PTM1	PTM0
	W								
0x0251 PTIM	R	0	0	0	0	0	0	PTIM1	PTIM0
	W								
0x0252 DDRM	R	0	0	0	0	0	0	DDRM1	DDRM0
	W								
0x0253 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0254 PERM	R	0	0	0	0	0	0	PERM1	PERM0
	W								
0x0255 PPSM	R	0	0	0	0	0	0	PPSM1	PPSM0
	W								
0x0256 WOMM	R	0	0	0	0	0	0	WOMM1	WOMM0
	W								
0x0257 PKGCR	R	APICLKS7	0	0	0	0	PKGCR2	PKGCR1	PKGCR0
	W								

= Unimplemented or Reserved

Table 2-40. PTS Register Field Descriptions

Field	Description
7-0 PTS	**Port S general-purpose input/output data**—Data Register When not used with an alternative signal, the associated pin can be used as general-purpose I/O. In general-purpose output mode the port data register bit value is driven to the pin. If the associated data direction bit is set to 1, a read returns the value of the port data register bit, otherwise the buffered pin input state is read.

2.4.3.21 Port S Input Register (PTIS)

Address 0x0249 Access: User read only[1]

	7	6	5	4	3	2	1	0
R	PTIS7	PTIS6	PTIS5	PTIS4	PTIS3	PTIS2	PTIS1	PTIS0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-22. Port S Input Register (PTIS)

[1] Read: Anytime
Write:Never

Table 2-41. PTIS Register Field Descriptions

Field	Description
7-0 PTIS	**Port S input data—** A read always returns the buffered input state of the associated pin. It can be used to detect overload or short circuit conditions on output pins.

2.4.3.22 Port S Data Direction Register (DDRS)

Address 0x024A Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	DDRS7	DDRS6	DDRS5	DDRS4	DDRS3	DDRS2	DDRS1	DDRS0
Reset	0	0	0	0	0	0	0	0

Figure 2-23. Port S Data Direction Register (DDRS)

[1] Read: Anytime
Write: Anytime

SZE	SZ	DBGADHM, DBGADLM	Access DH=DBGADH, DL=DBGADL	Comment
0	X	$FF00	Byte, data(ADDR[n])=DH Word, data(ADDR[n])=DH, data(ADDR[n+1])=X	Match data(ADDR[n])
0	X	$00FF	Word, data(ADDR[n])=X, data(ADDR[n+1])=DL	Match data(ADDR[n+1])
0	X	$00FF	Byte, data(ADDR[n])=X, data(ADDR[n+1])=DL	Possible unintended match
0	X	$FFFF	Word, data(ADDR[n])=DH, data(ADDR[n+1])=DL	Match data(ADDR[n], ADDR[n+1])
0	X	$FFFF	Byte, data(ADDR[n])=DH, data(ADDR[n+1])=DL	Possible unintended match
1	0	$0000	Word	No databus comparison
1	0	$00FF	Word, data(ADDR[n])=X, data(ADDR[n+1])=DL	Match only data at ADDR[n+1]
1	0	$FF00	Word, data(ADDR[n])=DH, data(ADDR[n+1])=X	Match only data at ADDR[n]
1	0	$FFFF	Word, data(ADDR[n])=DH, data(ADDR[n+1])=DL	Match data at ADDR[n] & ADDR[n+1]
1	1	$0000	Byte	No databus comparison
1	1	$FF00	Byte, data(ADDR[n])=DH	Match data at ADDR[n]

8.4.2.1.4 Comparator A Data Bus Comparison NDB Dependency

Comparator A features an NDB control bit, which allows data bus comparators to be configured to either trigger on equivalence or trigger on difference. This allows monitoring of a difference in the contents of an address location from an expected value.

When matching on an equivalence (NDB=0), each individual data bus bit position can be masked out by clearing the corresponding mask bit (DBGADHM/DBGADLM) so that it is ignored in the comparison. A match occurs when all data bus bits with corresponding mask bits set are equivalent. If all mask register bits are clear, then a match is based on the address bus only, the data bus is ignored.

When matching on a difference, mask bits can be cleared to ignore bit positions. A match occurs when any data bus bit with corresponding mask bit set is different. Clearing all mask bits, causes all bits to be ignored and prevents a match because no difference can be detected. In this case address bus equivalence does not cause a match.

Table 8-35. NDB and MASK bit dependency

NDB	DBGADHM[n] / DBGADLM[n]	Comment
0	0	Do not compare data bus bit.
0	1	Compare data bus bit. Match on equivalence.
1	0	Do not compare data bus bit.
1	1	Compare data bus bit. Match on difference.

8.4.2.2 Range Comparisons

Using the AB comparator pair for a range comparison, the data bus can also be used for qualification by using the comparator A data registers. Furthermore the DBGACTL RW and RWE bits can be used to qualify the range comparison on either a read or a write access. The corresponding DBGBCTL bits are ignored. The SZE and SZ control bits are ignored in range mode. The comparator A TAG bit is used to tag

Table 10-14. COP Watchdog Rates if COPOSCSEL1=1

CR2	CR1	CR0	COPCLK Cycles to Time-out (COPCLK is ACLK - internal RC-Oscillator clock)
0	0	0	COP disabled
0	0	1	2^7
0	1	0	2^9
0	1	1	2^{11}
1	0	0	2^{13}
1	0	1	2^{15}
1	1	0	2^{16}
1	1	1	2^{17}

10.3.2.10 Reserved Register CPMUTEST0

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special Mode can alter the S12CPMU's functionality.

0x003D

	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 10-13. Reserved Register (CPMUTEST0)

Read: Anytime

Write: Only in Special Mode

10.3.2.11 Reserved Register CPMUTEST1

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special Mode can alter the S12CPMU's functionality.

16.3.2.12 ATD Conversion Result Registers (ATDDR*n*)

The A/D conversion results are stored in 16 result registers. Results are always in unsigned data representation. Left and right justification is selected using the DJM control bit in ATDCTL3.

If automatic compare of conversions results is enabled (CMPE[*n*]=1 in ATDCMPE), these registers must be written with the compare values in left or right justified format depending on the actual value of the DJM bit. In this case, as the ATDDR*n* register is used to hold the compare value, the result will not be stored there at the end of the conversion but is lost.

Attention, n is the conversion number, NOT the channel number!

Read: Anytime

Write: Anytime

NOTE

For conversions not using automatic compare, results are stored in the result registers after each conversion. In this case avoid writing to ATDDRn except for initial values, because an A/D result might be overwritten.

16.3.2.12.1 Left Justified Result Data (DJM=0)

Module Base +
0x0010 = ATDDR0, 0x0012 = ATDDR1, 0x0014 = ATDDR2, 0x0016 = ATDDR3
0x0018 = ATDDR4, 0x001A = ATDDR5, 0x001C = ATDDR6, 0x001E = ATDDR7
0x0020 = ATDDR8, 0x0022 = ATDDR9, 0x0024 = ATDDR10, 0x0026 = ATDDR11
0x0028 = ATDDR12, 0x002A = ATDDR13, 0x002C = ATDDR14, 0x002E = ATDDR15

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Result-Bit[11:0]												0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 16-14. Left justified ATD conversion result register (ATDDR*n*)

Table 16-21 shows how depending on the A/D resolution the conversion result is transferred to the ATD result registers for left justified data. Compare is always done using all 12 bits of both the conversion result and the compare value in ATDDRn.

Table 16-21. Conversion result mapping to ATDDRn

A/D resolution	DJM	conversion result mapping to ATDDR*n*
8-bit data	0	Result-Bit[11:4] = conversion result, Result-Bit[3:0]=0000
10-bit data	0	Result-Bit[11:2] = conversion result, Result-Bit[1:0]=00
12-bit data	0	Result-Bit[11:0] = result

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:

25.3.2.11 Flash Common Command Object Register (FCCOB)

The FCCOB is an array of six words addressed via the CCOBIX index found in the FCCOBIX register. Byte wide reads and writes are allowed to the FCCOB register.

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 25-16. 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 25-17. Flash Common Command Object Low Register (FCCOBLO)

25.3.2.11.1 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 25-24. 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 25-24 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 25.4.6.

Table 25-24. 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	6'h0, Global address [17:16]
001	HI	Global address [15:8]
	LO	Global address [7:0]

Table 25-36. Erase Verify P-Flash Section Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x03	Global address [17:16] of a P-Flash block
001	Global address [15:0] of the first phrase to be verified	
010	Number of phrases to be verified	

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

Table 25-37. Erase Verify P-Flash 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 25-27)
		Set if an invalid global address [17:0] is supplied see Table 25-3)
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
		Set if the requested section crosses a the P-Flash address boundary
	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.

25.4.6.4 Read Once Command

The Read Once command provides read access to a reserved 64 byte field (8 phrases) located in the nonvolatile information register of P-Flash. The Read Once field is programmed using the Program Once command described in Section 25.4.6.6. The Read Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 25-38. Read Once Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x04	Not Required
001	Read Once phrase index (0x0000 - 0x0007)	
010	Read Once word 0 value	
011	Read Once word 1 value	
100	Read Once word 2 value	
101	Read Once word 3 value	

Table 26-36. Erase Verify P-Flash Section Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x03	Global address [17:16] of a P-Flash block
001	Global address [15:0] of the first phrase to be verified	
010	Number of phrases to be verified	

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

Table 26-37. Erase Verify P-Flash 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 see Table 26-3)
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
		Set if the requested section crosses a the P-Flash address boundary
	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.

26.4.6.4 Read Once Command

The Read Once command provides read access to a reserved 64 byte field (8 phrases) located in the nonvolatile information register of P-Flash. The Read Once field is programmed using the Program Once command described in Section 26.4.6.6. The Read Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 26-38. Read Once Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x04	Not Required
001	Read Once phrase index (0x0000 - 0x0007)	
010	Read Once word 0 value	
011	Read Once word 1 value	
100	Read Once word 2 value	
101	Read Once word 3 value	

The FPROT register, described in Section 29.3.2.9, can be set to protect regions in the Flash memory from accidental program or erase. Three separate memory regions, one growing upward from global address 0x3_8000 in the Flash memory (called the lower region), one growing downward from global address 0x3_FFFF in the Flash memory (called the higher region), and the remaining addresses in the Flash memory, can be activated for protection. The Flash memory addresses covered by these protectable regions are shown in the P-Flash memory map. The higher address region is mainly targeted to hold the boot loader code since it covers the vector space. Default protection settings as well as security information that allows the MCU to restrict access to the Flash module are stored in the Flash configuration field as described in Table 29-4.

Table 29-4. Flash Configuration Field

Global Address	Size (Bytes)	Description
0x3_FF00-0x3_FF07	8	Backdoor Comparison Key Refer to Section 29.4.6.11, “Verify Backdoor Access Key Command,” and Section 29.5.1, “Unsecuring the MCU using Backdoor Key Access”
0x3_FF08-0x3_FF0B[1]	4	Reserved
0x3_FF0C[1]	1	P-Flash Protection byte. Refer to Section 29.3.2.9, “P-Flash Protection Register (FPROT)”
0x3_FF0D[1]	1	EEPROM Protection byte. Refer to Section 29.3.2.10, “EEPROM Protection Register (DFPROT)”
0x3_FF0E[1]	1	Flash Nonvolatile byte Refer to Section 29.3.2.16, “Flash Option Register (FOPT)”
0x3_FF0F[1]	1	Flash Security byte Refer to Section 29.3.2.2, “Flash Security Register (FSEC)”

1 0x3FF08-0x3_FF0F form a Flash phrase and must be programmed in a single command write sequence. Each byte in the 0x3_FF08 - 0x3_FF0B reserved field should be programmed to 0xFF.

Table 29-49. Erase P-Flash Sector 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 29-27)
		Set if an invalid global address [17:16] is supplied (see Table 29-3)
		Set if a misaligned phrase address is supplied (global address [2:0] != 000)
	FPVIOL	Set if the selected P-Flash sector 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

29.4.6.10 Unsecure Flash Command

The Unsecure Flash command will erase the entire P-Flash and EEPROM memory space and, if the erase is successful, will release security.

Table 29-50. Unsecure Flash Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0B	Not required

Upon clearing CCIF to launch the Unsecure Flash command, the Memory Controller will erase the entire P-Flash and EEPROM 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. If the erase verify is not successful, the Unsecure Flash operation sets MGSTAT1 and terminates without changing the security state. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag is set after the Unsecure Flash operation has completed.

Table 29-51. Unsecure Flash Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 000 at command launch
		Set if command not available in current mode (see Table 29-27)
	FPVIOL	Set if any area of the P-Flash or 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

29.4.6.11 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command will only execute if it is enabled by the KEYEN bits in the FSEC register (see Table 29-10). The Verify Backdoor Access Key command releases security if user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see

CCOBIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 30-12. FCCOBIX Field Descriptions

Field	Description
2–0 CCOBIX[1:0]	**Common Command Register Index**— The CCOBIX bits are used to select which word of the FCCOB register array is being read or written to. See 30.3.2.11 Flash Common Command Object Register (FCCOB)," for more details.

30.3.2.4 Flash Reserved0 Register (FRSV0)

This Flash register is reserved for factory testing.

Offset Module Base + 0x000C

	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 30-8. Flash Reserved0 Register (FRSV0)

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

30.3.2.5 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 30-9. Flash Configuration Register (FCNFG)

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

Table 31-24. FCCOB - NVM Command Mode (Typical Usage)

CCOBIX[2:0]	Byte	FCCOB Parameter Fields (NVM Command Mode)
010	HI	Data 0 [15:8]
	LO	Data 0 [7:0]
011	HI	Data 1 [15:8]
	LO	Data 1 [7:0]
100	HI	Data 2 [15:8]
	LO	Data 2 [7:0]
101	HI	Data 3 [15:8]
	LO	Data 3 [7:0]

31.3.2.12 Flash Reserved1 Register (FRSV1)

This Flash register is reserved for factory testing.

Offset Module Base + 0x000C

	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 31-18. Flash Reserved1 Register (FRSV1)

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

31.3.2.13 Flash Reserved2 Register (FRSV2)

This Flash register is reserved for factory testing.

Offset Module Base + 0x000D

	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 31-19. Flash Reserved2 Register (FRSV2)

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

31.3.2.14 Flash Reserved3 Register (FRSV3)

This Flash register is reserved for factory testing.

31.4.4.3 Valid Flash Module Commands

Table 31-27 present the valid Flash commands, as enabled by the combination of the functional MCU mode (Normal SingleChip NS, Special Singlechip SS) with the MCU security state (Unsecured, Secured).

Special Singlechip mode is selected by input mmc_ss_mode_ts2 asserted. MCU Secured state is selected by input mmc_secure input asserted.

Table 31-27. Flash Commands by Mode and Security State

FCMD	Command	Unsecured		Secured	
		NS[1]	SS[2]	NS[3]	SS[4]
0x01	Erase Verify All Blocks	*	*	*	*
0x02	Erase Verify Block	*	*	*	*
0x03	Erase Verify P-Flash Section	*	*	*	
0x04	Read Once	*	*	*	
0x06	Program P-Flash	*	*	*	
0x07	Program Once	*	*	*	
0x08	Erase All Blocks		*		*
0x09	Erase Flash Block	*	*	*	
0x0A	Erase P-Flash Sector	*	*	*	
0x0B	Unsecure Flash		*		*
0x0C	Verify Backdoor Access Key	*		*	
0x0D	Set User Margin Level	*	*	*	
0x0E	Set Field Margin Level		*		
0x10	Erase Verify EEPROM Section	*	*	*	
0x11	Program EEPROM	*	*	*	
0x12	Erase EEPROM Sector	*	*	*	

1 Unsecured Normal Single Chip mode
2 Unsecured Special Single Chip mode.
3 Secured Normal Single Chip mode.
4 Secured Special Single Chip mode.

31.4.4.4 P-Flash Commands

Table 31-28 summarizes the valid P-Flash commands along with the effects of the commands on the P-Flash block and other resources within the Flash module.

Table 31-28. P-Flash Commands

FCMD	Command	Function on P-Flash Memory
0x01	Erase Verify All Blocks	Verify that all P-Flash (and EEPROM) blocks are erased.

Each command timing is given by:

$$t_{command} = \left(f_{NVMOP(cycle)} \cdot \frac{1}{f_{NVMOP}} + f_{NVMBUS(cycle)} \cdot \frac{1}{f_{NVMBUS}} \right)$$

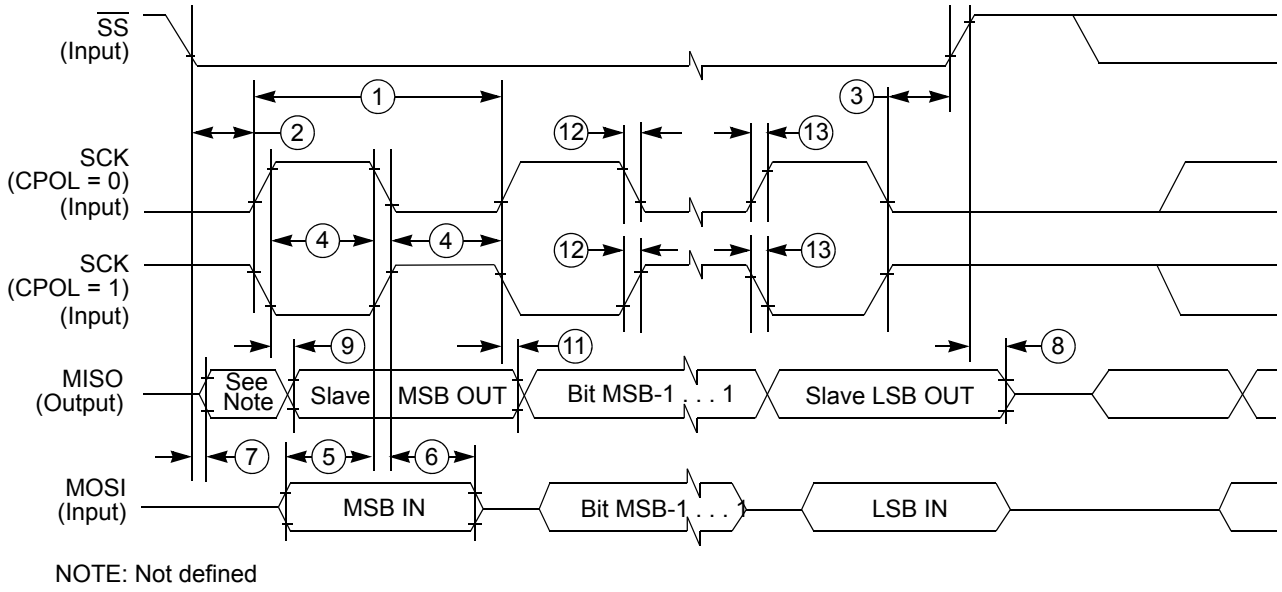
The timing parameters are captured exclusively during command execution (CCIF=0), excluding any time spent on the command write sequence to load and start the command. The formula above and the number of cycles in the following tables apply for the cases where the commands executed successfully in a new device, reflected in the minimum and typical timing parameters; however, due to aging, some of the commands will adjust their execution according to different margin settings and may eventually take longer to run than what the formula may return. The Max and Lfmax timing columns in the tables below already reflect this adjustment where applicable.

A summary of key timing parameters can be found from Table A-34 to Table A-38.

Table A-34. NVM Clock Timing Characteristics

Num	Rating	Symbol	Min	Typ	Max	Unit
1	Bus frequency	f_{NVMBUS}	1	25	25	MHz
2	Operating frequency	f_{NVMOP}	0.8	1.0	1.05	MHz

In Figure A-10 the timing diagram for slave mode with transmission format CPHA = 1 is depicted.



NOTE: Not defined

Figure A-10. SPI Slave Timing (CPHA = 1)