

Welcome to [E-XFL.COM](http://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	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	35
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LQFP
Supplier Device Package	52-TQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12gc64mpbe">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12gc64mpbe</a>

## Chapter 1

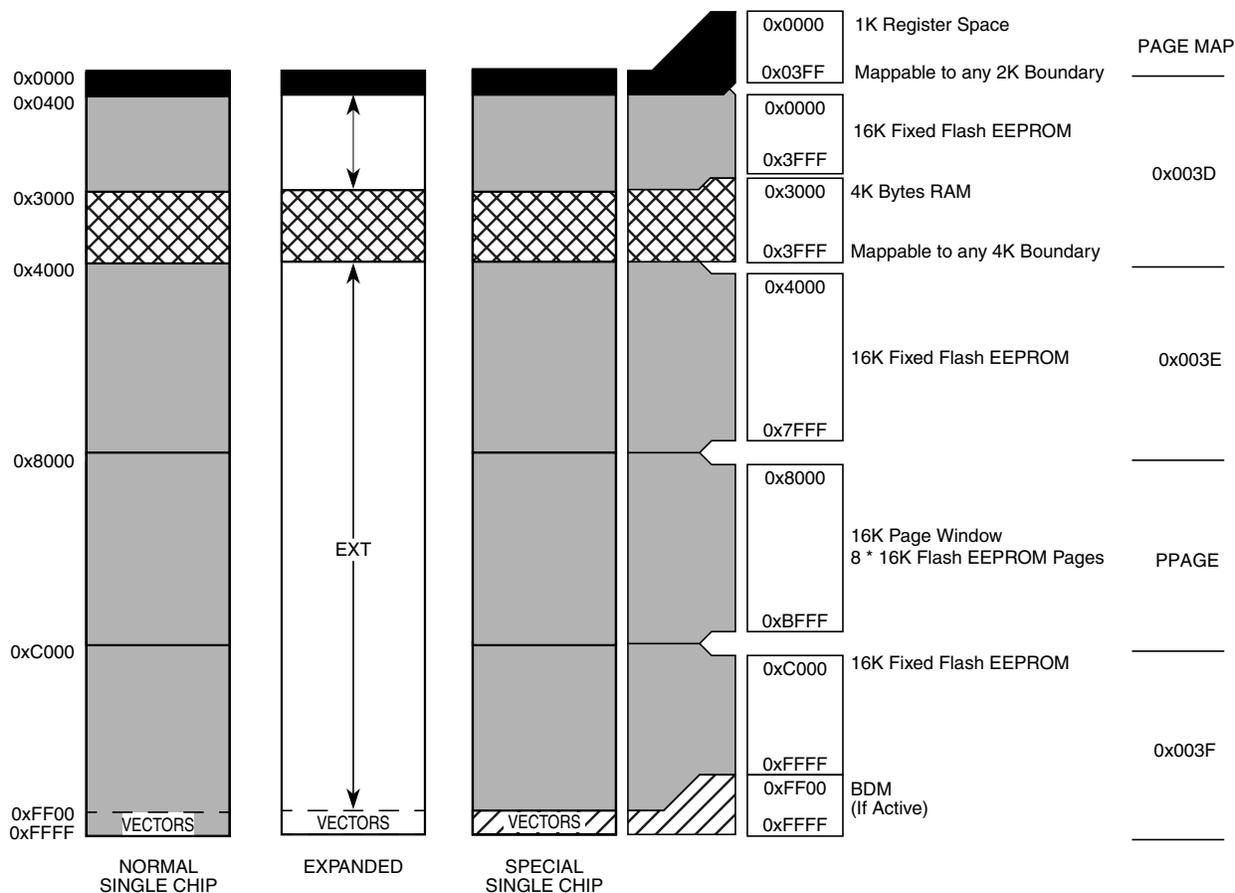
### MC9S12C and MC9S12GC Device Overview (MC9S12C128)

1.1	Introduction .....	17
1.1.1	Features .....	17
1.1.2	Modes of Operation .....	19
1.1.3	Block Diagram .....	20
1.2	Memory Map and Registers .....	21
1.2.1	Device Memory Map .....	21
1.2.2	Detailed Register Map .....	27
1.2.3	Part ID Assignments .....	44
1.3	Signal Description .....	45
1.3.1	Device Pinouts .....	45
1.3.2	Signal Properties Summary .....	48
1.3.3	Pin Initialization for 48- and 52-Pin LQFP Bond Out Versions .....	49
1.3.4	Detailed Signal Descriptions .....	50
1.3.5	Power Supply Pins .....	55
1.4	System Clock Description .....	57
1.5	Modes of Operation .....	57
1.5.1	Chip Configuration Summary .....	57
1.5.2	Security .....	58
1.5.3	Low-Power Modes .....	59
1.6	Resets and Interrupts .....	60
1.6.1	Vectors .....	60
1.6.2	Resets .....	62
1.7	Device Specific Information and Module Dependencies .....	62
1.7.1	PPAGE .....	62
1.7.2	BDM Alternate Clock .....	63
1.7.3	Extended Address Range Emulation Implications .....	63
1.7.4	VREGEN .....	64
1.7.5	$V_{DD1}$ , $V_{DD2}$ , $V_{SS1}$ , $V_{SS2}$ .....	64
1.7.6	Clock Reset Generator And VREG Interface .....	64
1.7.7	Analog-to-Digital Converter .....	64
1.7.8	MODRR Register Port T And Port P Mapping .....	64
1.7.9	Port AD Dependency On PIM And ATD Registers .....	64
1.8	Recommended Printed Circuit Board Layout .....	65

## Chapter 2

### Port Integration Module (PIM9C32) Block Description

2.1	Introduction .....	73
2.1.1	Features .....	73



The figure shows a useful map, which is not the map out of reset. After reset the map is:

- 0x0000–0x03FF: Register Space
- 0x0000–0x0FFF: 4K RAM (only 3K visible 0x0400–0x0FFF)

Flash erase sector size is 1024 bytes

**Figure 1-2. MC9S12C128 and MC9S12GC128 User Configurable Memory Map**

**0x0010–0x0014 MMC Map 1 of 4 (HCS12 Module Mapping Control)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0x0010	INITRM	Read:	RAM15	RAM14	RAM13	RAM12	RAM11	0	0	RAMHAL
		Write:								
0x0011	INITRG	Read:	0	REG14	REG13	REG12	REG11	0	0	0
		Write:								
0x0012	INITEE	Read:	EE15	EE14	EE13	EE12	EE11	0	0	EEON
		Write:								
0x0013	MISC	Read:	0	0	0	0	EXSTR1	EXSTR0	ROMHM	ROMON
		Write:								
0x0014	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

**0x0015–0x0016 INT Map 1 of 2 (HCS12 Interrupt)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0x0015	ITCR	Read:	0	0	0	WRINT	ADR3	ADR2	ADR1	ADR0
		Write:								
0x0016	ITEST	Read:	INTE	INTC	INTA	INT8	INT6	INT4	INT2	INT0
		Write:								

**0x0017–0x0017 MMC Map 2 of 4 (HCS12 Module Mapping Control)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0x0017	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

**0x0018–0x0018 Miscellaneous Peripherals (Device User Guide)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0x0018	Reserved	Read:	0	0	0	0	0	0	0	0
		Write:								

**0x0019–0x0019 VREG3V3 (Voltage Regulator)**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
\$0019	VREGCTRL	Read:	0	0	0	0	0	LVDS	LVIE	LVIF
		Write:								

## 4.3.1 Module Memory Map

Table 4-2. MEBI Memory Map

Address Offset	Use	Access
0x0000	Port A Data Register (PORTA)	R/W
0x0001	Port B Data Register (PORTB)	R/W
0x0002	Data Direction Register A (DDRA)	R/W
0x0003	Data Direction Register B (DDRB)	R/W
0x0004	Reserved	R
0x0005	Reserved	R
0x0006	Reserved	R
0x0007	Reserved	R
0x0008	Port E Data Register (PORTE)	R/W
0x0009	Data Direction Register E (DDRE)	R/W
0x000A	Port E Assignment Register (PEAR)	R/W
0x000B	Mode Register (MODE)	R/W
0x000C	Pull Control Register (PUCR)	R/W
0x000D	Reduced Drive Register (RDRIV)	R/W
0x000E	External Bus Interface Control Register (EBICTL)	R/W
0x000F	Reserved	R
0x001E	IRQ Control Register (IRQCR)	R/W
0x00032	Port K Data Register (PORTK)	R/W
0x00033	Data Direction Register K (DDRK)	R/W

## 4.3.2 Register Descriptions

### 4.3.2.1 Port A Data Register (PORTA)

Module Base + 0x0000

Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W								
Reset	0	0	0	0	0	0	0	0
Single Chip	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Expanded Wide, Emulation Narrow with IVIS, and Peripheral	AB/DB15	AB/DB14	AB/DB13	AB/DB12	AB/DB11	AB/DB10	AB/DB9	AB/DB8
Expanded Narrow	AB15 and DB15/DB7	AB14 and DB14/DB6	AB13 and DB13/DB5	AB12 and DB12/DB4	AB11 and DB11/DB3	AB10 and DB10/DB2	AB9 and DB9/DB1	AB8 and DB8/DB0

Figure 4-2. Port A Data Register (PORTA)

These register locations are not used (reserved). All unused registers and bits in this block return logic 0s when read. Writes to these registers have no effect.

These registers are not in the on-chip map in special peripheral mode.

### 4.3.2.6 Port E Data Register (PORTE)

Module Base + 0x0008

Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	Bit 1	Bit 0
W								
Reset	0	0	0	0	0	0	u	u
Alternate Pin Function	NOACC	MODB or IPIPE1 or CLKTO	MODA or IPIPE0	ECLK	$\overline{\text{LSTRB}}$ or $\overline{\text{TAGLO}}$	R/ $\overline{\text{W}}$	$\overline{\text{IRQ}}$	$\overline{\text{XIRQ}}$

= Unimplemented or Reserved      u = Unaffected by reset

**Figure 4-10. Port E Data Register (PORTE)**

Read: Anytime when register is in the map

Write: Anytime when register is in the map

Port E is associated with external bus control signals and interrupt inputs. These include mode select (MODB/IPIPE1, MODA/IPIPE0), E clock, size ( $\overline{\text{LSTRB}}$ / $\overline{\text{TAGLO}}$ ), read/write (R/ $\overline{\text{W}}$ ),  $\overline{\text{IRQ}}$ , and  $\overline{\text{XIRQ}}$ . When not used for one of these specific functions, port E pins 7:2 can be used as general-purpose I/O and pins 1:0 can be used as general-purpose input. The port E assignment register (PEAR) selects the function of each pin and DDRE determines whether each pin is an input or output when it is configured to be general-purpose I/O. DDRE also determines the source of data for a read of PORTE.

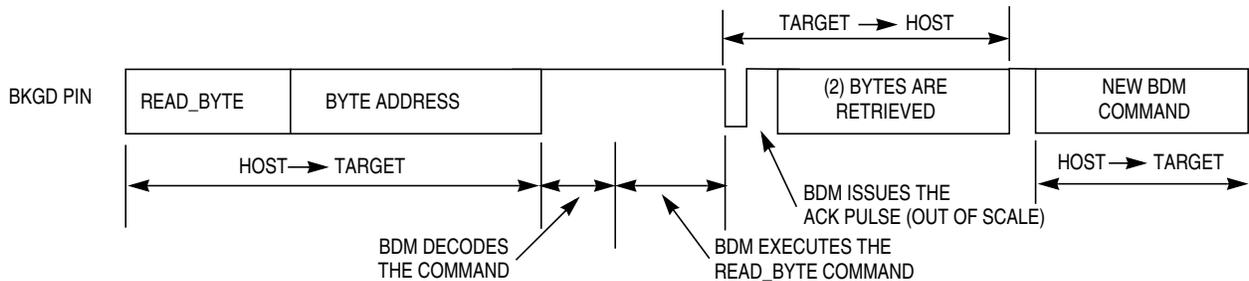
Some of these pins have software selectable pull resistors.  $\overline{\text{IRQ}}$  and  $\overline{\text{XIRQ}}$  can only be pulled up whereas the polarity of the PE7, PE4, PE3, and PE2 pull resistors are determined by chip integration. Please refer to the device overview chapter (Signal Property Summary) to determine the polarity of these resistors. A single control bit enables the pull devices for all of these pins when they are configured as inputs.

This register is not in the on-chip map in special peripheral mode or in expanded modes when the EME bit is set. Therefore, these accesses will be echoed externally.

#### NOTE

It is unwise to write PORTE and DDRE as a word access. If you are changing port E pins from being inputs to outputs, the data may have extra transitions during the write. It is best to initialize PORTE before enabling as outputs.

Figure 6-11 shows the ACK handshake protocol in a command level timing diagram. The READ\_BYTE instruction is used as an example. First, the 8-bit instruction opcode is sent by the host, followed by the address of the memory location to be read. The target BDM decodes the instruction. A bus cycle is grabbed (free or stolen) by the BDM and it executes the READ\_BYTE operation. Having retrieved the data, the BDM issues an ACK pulse to the host controller, indicating that the addressed byte is ready to be retrieved. After detecting the ACK pulse, the host initiates the byte retrieval process. Note that data is sent in the form of a word and the host needs to determine which is the appropriate byte based on whether the address was odd or even.



**Figure 6-11. Handshake Protocol at Command Level**

Differently from the normal bit transfer (where the host initiates the transmission), the serial interface ACK handshake pulse is initiated by the target MCU by issuing a falling edge in the BKGD pin. The hardware handshake protocol in Figure 6-10 specifies the timing when the BKGD pin is being driven, so the host should follow this timing constraint in order to avoid the risk of an electrical conflict in the BKGD pin.

**NOTE**

The only place the BKGD pin can have an electrical conflict is when one side is driving low and the other side is issuing a speedup pulse (high). Other “highs” are pulled rather than driven. However, at low rates the time of the speedup pulse can become lengthy and so the potential conflict time becomes longer as well.

The ACK handshake protocol does not support nested ACK pulses. If a BDM command is not acknowledge by an ACK pulse, the host needs to abort the pending command first in order to be able to issue a new BDM command. When the CPU enters WAIT or STOP while the host issues a command that requires CPU execution (e.g., WRITE\_BYTE), the target discards the incoming command due to the WAIT or STOP being detected. Therefore, the command is not acknowledged by the target, which means that the ACK pulse will not be issued in this case. After a certain time the host should decide to abort the ACK sequence in order to be free to issue a new command. Therefore, the protocol should provide a mechanism in which a command, and therefore a pending ACK, could be aborted.

**NOTE**

Differently from a regular BDM command, the ACK pulse does not provide a time out. This means that in the case of a WAIT or STOP instruction being executed, the ACK would be prevented from being issued. If not aborted, the ACK would remain pending indefinitely. See the handshake abort procedure described in Section 6.4.8, “Hardware Handshake Abort Procedure.”

**Table 8-3. ATDCTL3 Field Descriptions (continued)**

Field	Description
2 FIFO	<p><b>Result Register FIFO Mode</b> — If this bit is zero (non-FIFO mode), the A/D conversion results map into the result registers based on the conversion sequence; the result of the first conversion appears in the first result register, the second result in the second result register, and so on.</p> <p>If this bit is one (FIFO mode) the conversion counter is not reset at the beginning or ending of a conversion sequence; sequential conversion results are placed in consecutive result registers. In a continuously scanning conversion sequence, the result register counter will wrap around when it reaches the end of the result register file. The conversion counter value (CC2-0 in ATDSTAT0) can be used to determine where in the result register file, the current conversion result will be placed.</p> <p>Aborting a conversion or starting a new conversion by write to an ATDCTL register (ATDCTL5-0) clears the conversion counter even if FIFO=1. So the first result of a new conversion sequence, started by writing to ATDCTL5, will always be placed in the first result register (ATDDDR0). Intended usage of FIFO mode is continuous conversion (SCAN=1) or triggered conversion (ETRIG=1).</p> <p>Which result registers hold valid data can be tracked using the conversion complete flags. Fast flag clear mode may or may not be useful in a particular application to track valid data.</p> <p>0 Conversion results are placed in the corresponding result register up to the selected sequence length. 1 Conversion results are placed in consecutive result registers (wrap around at end).</p>
1-0 FRIZ[1:0]	<p><b>Background Debug Freeze Enable</b> — When debugging an application, it is useful in many cases to have the ATD pause when a breakpoint (Freeze Mode) is encountered. These 2 bits determine how the ATD will respond to a breakpoint as shown in Table 8-5. Leakage onto the storage node and comparator reference capacitors may compromise the accuracy of an immediately frozen conversion depending on the length of the freeze period.</p>

**Table 8-4. Conversion Sequence Length Coding**

S8C	S4C	S2C	S1C	Number of Conversions per Sequence
0	0	0	0	8
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	X	X	X	8

**Table 8-5. ATD Behavior in Freeze Mode (Breakpoint)**

FRZ1	FRZ0	Behavior in Freeze Mode
0	0	Continue conversion
0	1	Reserved
1	0	Finish current conversion, then freeze
1	1	Freeze Immediately

### 10.1.4 Modes of Operation

The following modes of operation are specific to the MSCAN. See Section 10.4, “Functional Description,” for details.

- Listen-Only Mode
- MSCAN Sleep Mode
- MSCAN Initialization Mode
- MSCAN Power Down Mode

## 10.2 External Signal Description

The MSCAN uses two external pins:

### 10.2.1 RXCAN — CAN Receiver Input Pin

RXCAN is the MSCAN receiver input pin.

### 10.2.2 TXCAN — CAN Transmitter Output Pin

TXCAN is the MSCAN transmitter output pin. The TXCAN output pin represents the logic level on the CAN bus:

- 0 = Dominant state
- 1 = Recessive state

### 10.2.3 CAN System

A typical CAN system with MSCAN is shown in Figure 10-2. Each CAN station is connected physically to the CAN bus lines through a transceiver device. The transceiver is capable of driving the large current needed for the CAN bus and has current protection against defective CAN or defective stations.

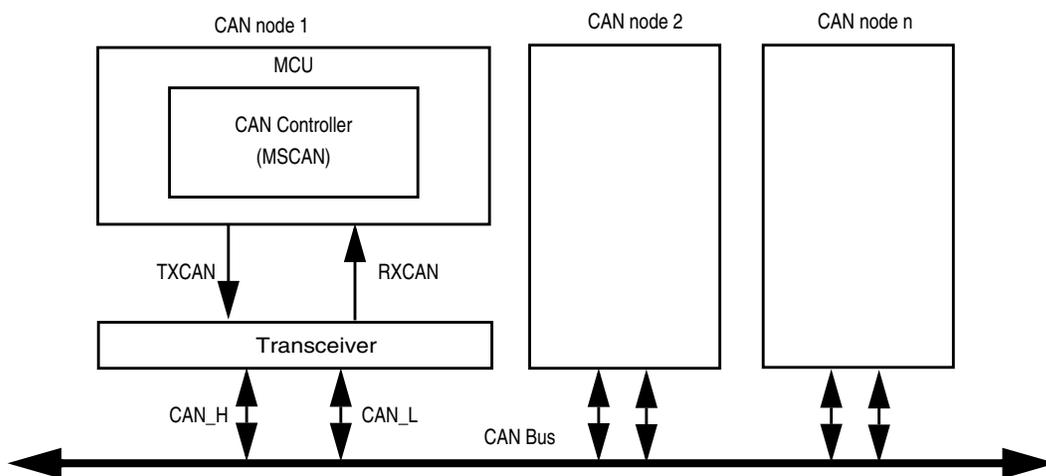


Figure 10-2. CAN System

### 10.3.2.16 MSCAN Identifier Acceptance Registers (CANIDAR0-7)

On reception, each message is written into the background receive buffer. The CPU is only signalled to read the message if it passes the criteria in the identifier acceptance and identifier mask registers (accepted); otherwise, the message is overwritten by the next message (dropped).

The acceptance registers of the MSCAN are applied on the IDR0–IDR3 registers (see Section 10.3.3.1, “Identifier Registers (IDR0–IDR3)”) of incoming messages in a bit by bit manner (see Section 10.4.3, “Identifier Acceptance Filter”).

For extended identifiers, all four acceptance and mask registers are applied. For standard identifiers, only the first two (CANIDAR0/1, CANIDMR0/1) are applied.

Module Base + 0x0010 (CANIDAR0)  
 0x0011 (CANIDAR1)  
 0x0012 (CANIDAR2)  
 0x0013 (CANIDAR3)

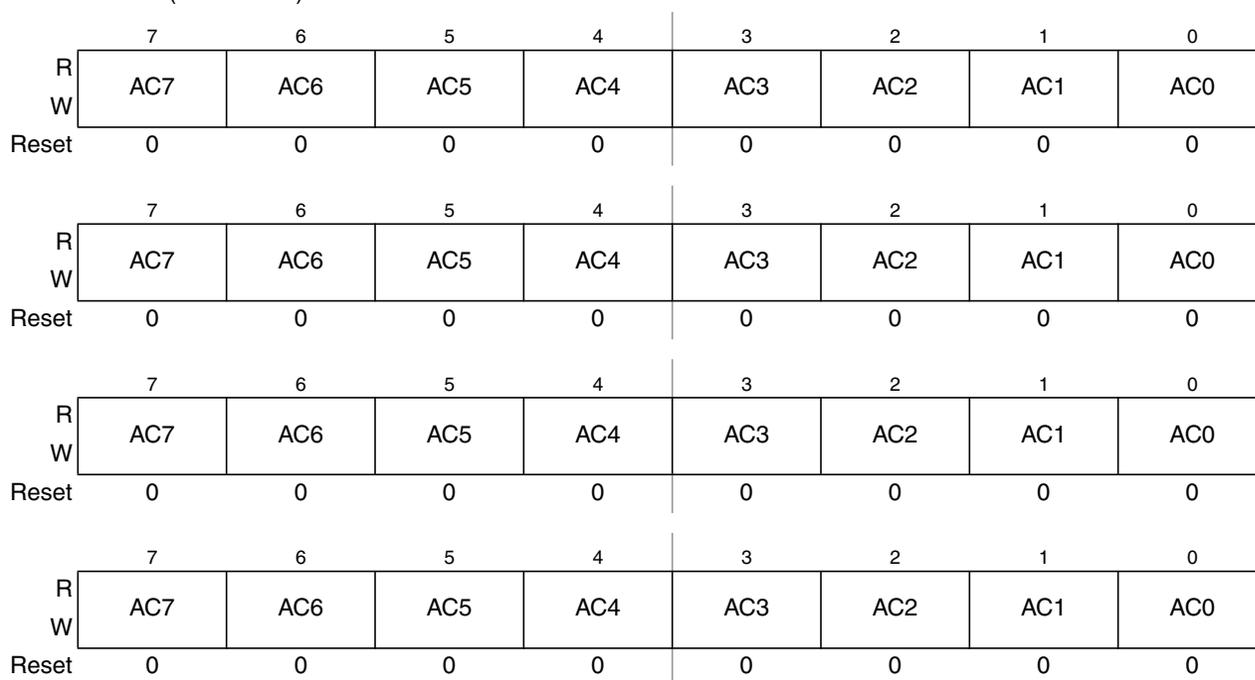


Figure 10-19. MSCAN Identifier Acceptance Registers (First Bank) — CANIDAR0–CANIDAR3

Read: Anytime

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

Table 10-19. CANIDAR0–CANIDAR3 Register Field Descriptions

Field	Description
7:0 AC[7:0]	<b>Acceptance Code Bits</b> — 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.

## 11.2 External Signal Description

This section lists and describes the signals that connect off chip.

### 11.2.1 $V_{DDPLL}$ and $V_{SSPLL}$ — PLL Operating Voltage, PLL Ground

These pins provide the operating voltage ( $V_{DDPLL}$ ) and ground ( $V_{SSPLL}$ ) for the OSCV2 circuitry. This allows the supply voltage to the OSCV2 to be independently bypassed.

### 11.2.2 EXTAL and XTAL — Clock/Crystal Source Pins

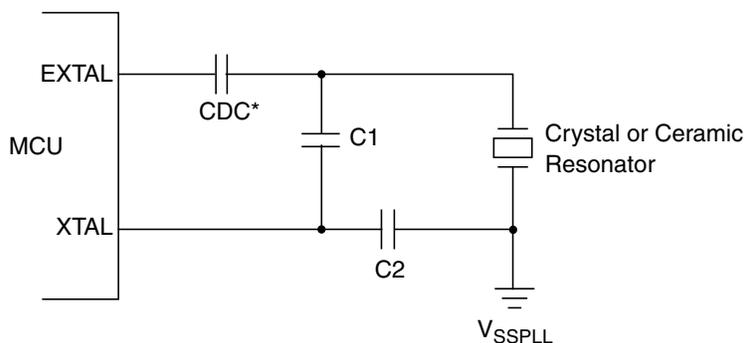
These pins provide the interface for either a crystal or a CMOS compatible clock to control the internal clock generator circuitry. EXTAL is the external clock input or the input to the crystal oscillator amplifier. XTAL is the output of the crystal oscillator amplifier. All the MCU internal system clocks are derived from the EXTAL input frequency. In full stop mode ( $PSTP = 0$ ) the EXTAL pin is pulled down by an internal resistor of typical 200 k $\Omega$ .

#### NOTE

Freescale Semiconductor recommends an evaluation of the application board and chosen resonator or crystal by the resonator or crystal supplier.

The Crystal circuit is changed from standard.

The Colpitts circuit is not suited for overtone resonators and crystals.



\* Due to the nature of a translated ground Colpitts oscillator a DC voltage bias is applied to the crystal.

Please contact the crystal manufacturer for crystal DC bias conditions and recommended capacitor value CDC.

**Figure 11-1. Colpitts Oscillator Connections (XCLKS = 0)**

#### NOTE

The Pierce circuit is not suited for overtone resonators and crystals without a careful component selection.

If software clears TE while a transmission is in progress ( $TC = 0$ ), the frame in the transmit shift register continues to shift out. To avoid accidentally cutting off the last frame in a message, always wait for TDRE to go high after the last frame before clearing TE.

To separate messages with preambles with minimum idle line time, use this sequence between messages:

1. Write the last byte of the first message to SCIDRH/L.
2. Wait for the TDRE flag to go high, indicating the transfer of the last frame to the transmit shift register.
3. Queue a preamble by clearing and then setting the TE bit.
4. Write the first byte of the second message to SCIDRH/L.

### 13.4.3.3 Break Characters

Writing a logic 1 to the send break bit, SBK, in SCI control register 2 (SCICR2) loads the transmit shift register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on the M bit in SCI control register 1 (SCICR1). As long as SBK is at logic 1, transmitter logic continuously loads break characters into the transmit shift register. After software clears the SBK bit, the shift register finishes transmitting the last break character and then transmits at least one logic 1. The automatic logic 1 at the end of a break character guarantees the recognition of the start bit of the next frame.

The SCI recognizes a break character when a start bit is followed by eight or nine logic 0 data bits and a logic 0 where the stop bit should be. Receiving a break character has these effects on SCI registers:

- Sets the framing error flag, FE
- Sets the receive data register full flag, RDRF
- Clears the SCI data registers (SCIDRH/L)
- May set the overrun flag, OR, noise flag, NF, parity error flag, PE, or the receiver active flag, RAF (see Section 13.3.2.4, “SCI Status Register 1 (SCISR1)” and Section 13.3.2.5, “SCI Status Register 2 (SCISR2)”)

### 13.4.3.4 Idle Characters

An idle character contains all logic 1s and has no start, stop, or parity bit. Idle character length depends on the M bit in SCI control register 1 (SCICR1). The preamble is a synchronizing idle character that begins the first transmission initiated after writing the TE bit from 0 to 1.

If the TE bit is cleared during a transmission, the **Tx output** signal becomes idle after completion of the transmission in progress. Clearing and then setting the TE bit during a transmission queues an idle character to be sent after the frame currently being transmitted.

**Table 13-11. Start Bit Verification**

RT3, RT5, and RT7 Samples	Start Bit Verification	Noise Flag
100	Yes	1
101	No	0
110	No	0
111	No	0

If start bit verification is not successful, the RT clock is reset and a new search for a start bit begins.

To determine the value of a data bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. [Table 13-12](#) summarizes the results of the data bit samples.

**Table 13-12. Data Bit Recovery**

RT8, RT9, and RT10 Samples	Data Bit Determination	Noise Flag
000	0	0
001	0	1
010	0	1
011	1	1
100	0	1
101	1	1
110	1	1
111	1	0

#### NOTE

The RT8, RT9, and RT10 samples do not affect start bit verification. If any or all of the RT8, RT9, and RT10 start bit samples are logic 1s following a successful start bit verification, the noise flag (NF) is set and the receiver assumes that the bit is a start bit (logic 0).

To verify a stop bit and to detect noise, recovery logic takes samples at RT8, RT9, and RT10. [Table 13-13](#) summarizes the results of the stop bit samples.

**Table 13-13. Stop Bit Recovery**

RT8, RT9, and RT10 Samples	Framing Error Flag	Noise Flag
000	1	0
001	1	1
010	1	1
011	0	1
100	1	1
101	0	1
110	0	1
111	0	0



## 17.3.2 Register Descriptions

The Flash module contains a set of 16 control and status registers located between module base + 0x0000 and 0x000F. A summary of the Flash module registers is given in Figure 17-3. Detailed descriptions of each register bit are provided.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000	R	FDIVLD	PRDIV8	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
FCLKDIV	W								
0x0001	R	KEYEN1	KEYEN0	NV5	NV4	NV3	NV2	SEC1	SEC0
FSEC	W								
0x0002	R	0	0	0	0	0	0	0	0
RESERVED1 <sup>(1)</sup>	W								
0x0003	R	CBEIE	CCIE	KEYACC	0	0	0	0	0
FCNFG	W								
0x0004	R	FPOPEN	NV6	FPHDIS	FPHS1	FPHS0	NV2	NV1	NV0
FPROT	W								
0x0005	R	CBEIF	CCIF	PVIOL	ACCERR	0	BLANK	FAIL	DONE
FSTAT	W								
0x0006	R	0	CMDB6	CMDB5	0	0	CMDB2	0	CMDB0
FCMD	W								
0x0007	R	0	0	0	0	0	0	0	0
RESERVED2 <sup>1</sup>	W								
0x0008	R	0	0	0	FABHI				
FADDRHI <sup>1</sup>	W								
0x0009	R	FABLO							
FADDRLO <sup>1</sup>	W								
0x000A	R	FDHI							
FDATAHI <sup>1</sup>	W								
0x000B	R	FDLO							
FDATALO <sup>1</sup>	W								
0x000C	R	0	0	0	0	0	0	0	0
RESERVED3 <sup>1</sup>	W								
0x000D	R	0	0	0	0	0	0	0	0
RESERVED4 <sup>1</sup>	W								
0x000E	R	0	0	0	0	0	0	0	0
RESERVED5 <sup>1</sup>	W								
0x000F	R	0	0	0	0	0	0	0	0
RESERVED6 <sup>1</sup>	W								

= Unimplemented or Reserved

**Figure 17-3. Flash Register Summary**

1. Intended for factory test purposes only.

### 18.4.1.3 Valid Flash Commands

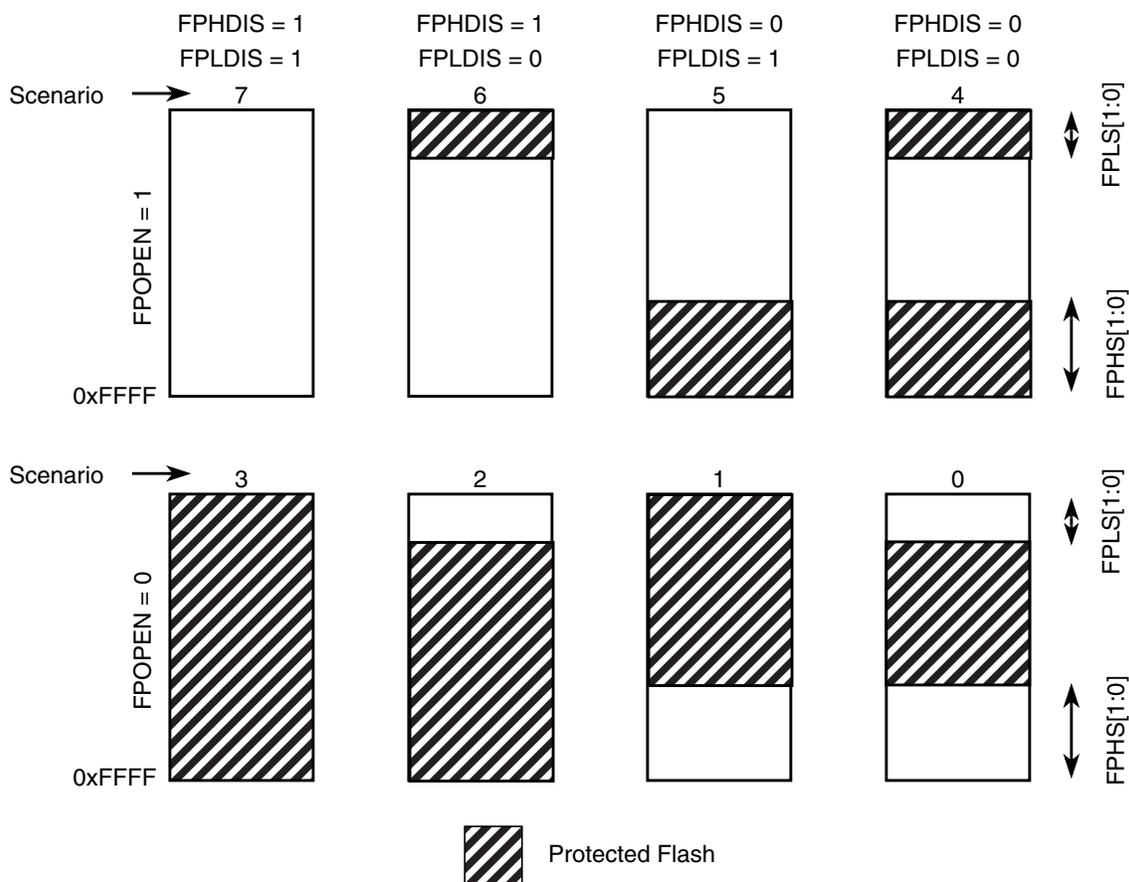
Table 18-16 summarizes the valid Flash commands along with the effects of the commands on the Flash array.

**Table 18-16. Valid Flash Commands**

FCMD	Meaning	Function on Flash Array
0x05	Erase Verify	Verify all bytes in the Flash array are erased. If the Flash array is erased, the BLANK bit will set in the FSTAT register upon command completion.
0x20	Program	Program a word (2 bytes) in the Flash array.
0x40	Sector Erase	Erase all 512 bytes in a sector of the Flash array.
0x41	Mass Erase	Erase all bytes in the Flash array. A mass erase of the full Flash array is only possible when FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register are set prior to launching the command.

#### CAUTION

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


**Figure 19-11. Flash Protection Scenarios**

### 19.3.2.5.1 Flash Protection Restrictions

The general guideline is that protection can only be added, not removed. All valid transitions between Flash protection scenarios are specified in [Table 19-13](#). Any attempt to write an invalid scenario to the FPROT register will be ignored and the FPROT register will remain unchanged. The contents of the FPROT register reflect the active protection scenario.

**Table 19-13. Flash Protection Scenario Transitions**

From Protection Scenario	To Protection Scenario <sup>(1)</sup>							
	0	1	2	3	4	5	6	7
0	X	X	X	X				
1		X		X				
2			X	X				
3				X				
4				X	X			
5			X	X	X	X		

### 19.4.1.2 Command Write Sequence

The Flash command controller is used to supervise the command write sequence to execute program, erase, and erase verify algorithms.

Before starting a command write sequence, the ACCERR and PVIOL flags in the FSTAT register must be clear and the CBEIF flag should be tested to determine the state of the address, data, and command buffers. If the CBEIF flag is set, indicating the buffers are empty, a new command write sequence can be started. If the CBEIF flag is clear, indicating the buffers are not available, a new command write sequence will overwrite the contents of the address, data, and command buffers.

A command write sequence consists of three steps which must be strictly adhered to with writes to the Flash module not permitted between the steps. However, Flash register and array reads are allowed during a command write sequence. The basic command write sequence is as follows:

1. Write to a valid address in the Flash array memory.
2. Write a valid command to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the command.

The address written in step 1 will be stored in the FADDR registers and the data will be stored in the FDATA registers. When the CBEIF flag is cleared in step 3, the CCIF flag is cleared by the Flash command controller indicating that the command was successfully launched. For all command write sequences, the CBEIF flag will set after the CCIF flag is cleared indicating that the address, data, and command buffers are ready for a new command write sequence to begin. A buffered command will wait for the active operation to be completed before being launched. Once a command is launched, the completion of the command operation is indicated by the setting of the CCIF flag in the FSTAT register. The CCIF flag will set upon completion of all active and buffered commands.

**Table 21-14. FCMD Field Descriptions**

Field	Description
6, 5, 2, 0 CMDB[6:5] CMDB[2] CMDB[0]	Valid Flash commands are shown in <a href="#">Table 21-15</a> . An attempt to execute any command other than those listed in <a href="#">Table 21-15</a> will set the ACCERR bit in the FSTAT register (see <a href="#">Section 21.3.2.6</a> ).

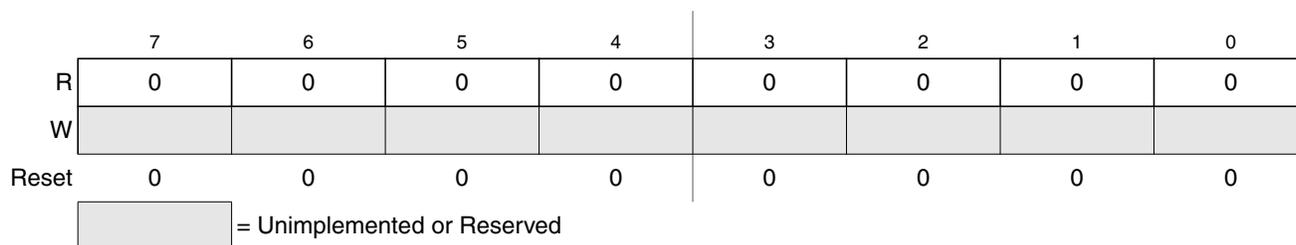
**Table 21-15. Valid Flash Command List**

CMDB	NVM Command
0x05	Erase verify
0x20	Word program
0x40	Sector erase
0x41	Mass erase

### 21.3.2.8 RESERVED2

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x0007



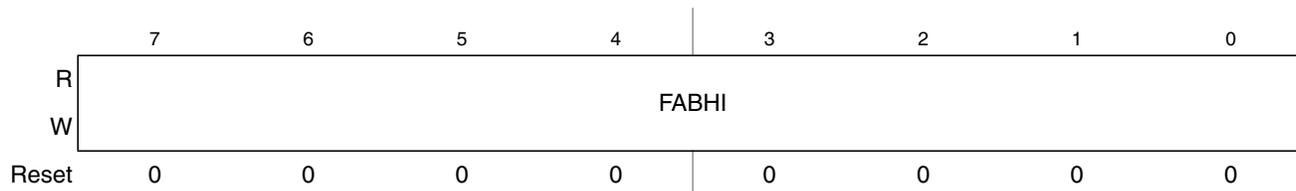
**Figure 21-12. RESERVED2**

All bits read 0 and are not writable.

### 21.3.2.9 Flash Address Register (FADDR)

FADDRHI and FADDRLO are the Flash address registers.

Module Base + 0x0008



**Figure 21-13. Flash Address High Register (FADDRHI)**

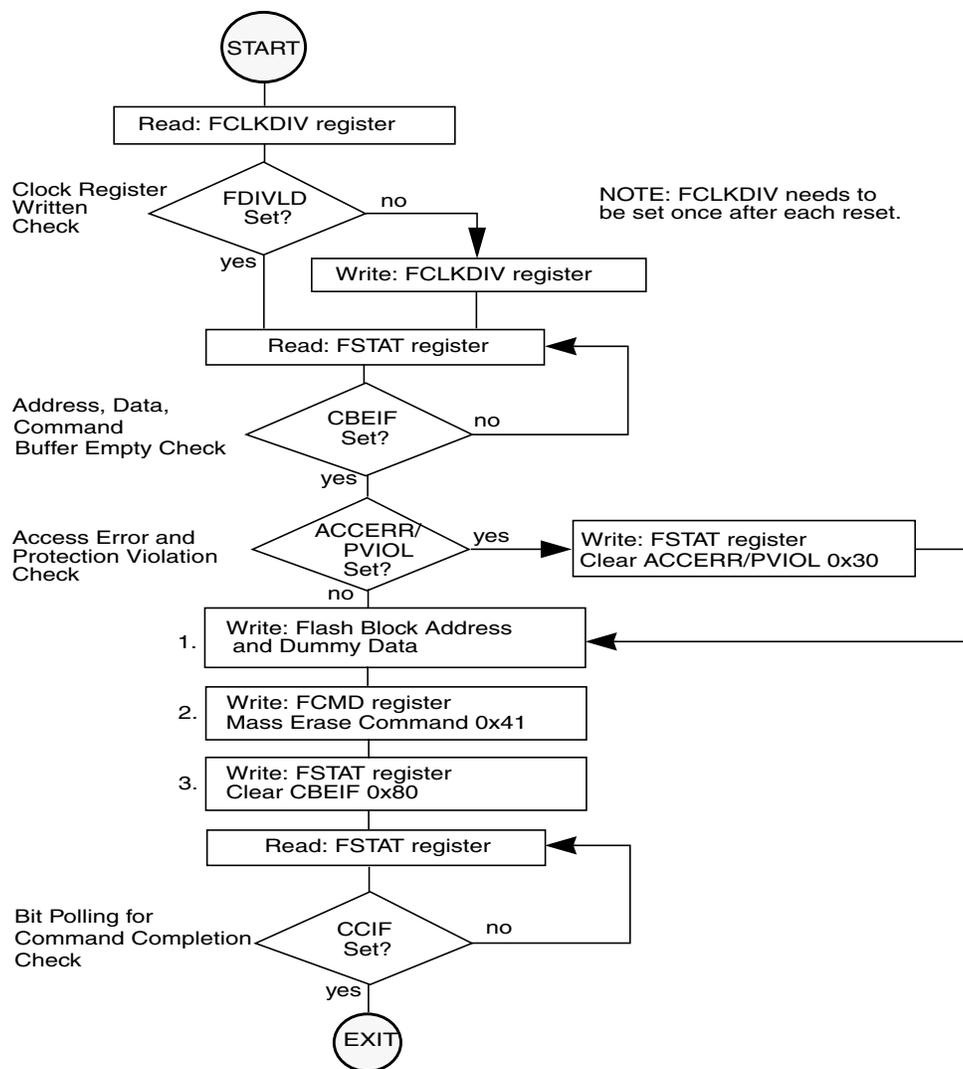
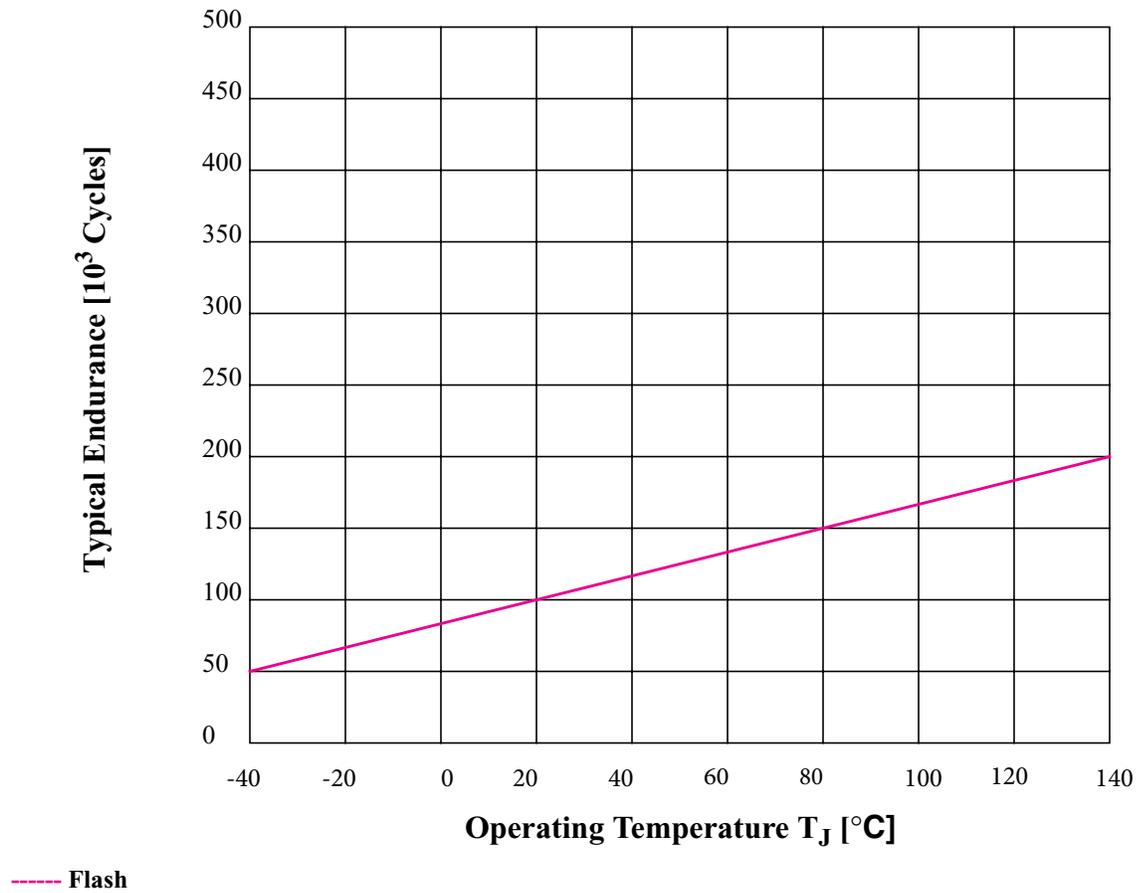


Figure 21-25. Example Mass Erase Command Flow

**Figure A-5. Typical Endurance vs Temperature**


## A.6 SPI

This section provides electrical parametrics and ratings for the SPI.

In [Table A-20](#) the measurement conditions are listed.

**Table A-20. Measurement Conditions**

Description	Value	Unit
Drive mode	Full drive mode	—
Load capacitance $C_{LOAD}$ , on all outputs	50	pF
Thresholds for delay measurement points	(20% / 80%) $V_{DDX}$	V

### A.6.1 Master Mode

In [Figure A-6](#) the timing diagram for master mode with transmission format CPHA=0 is depicted.