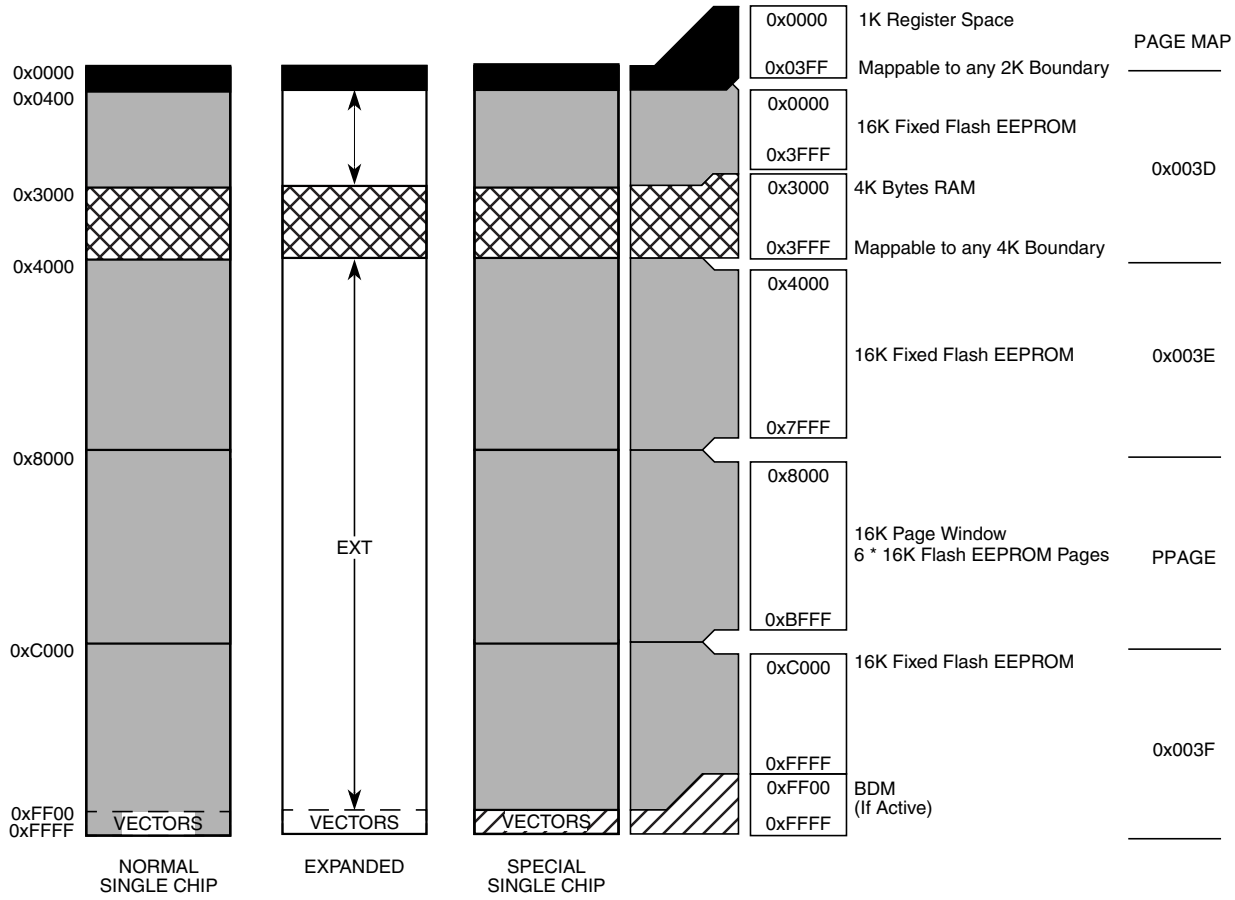
Welcome to E-XFL.COM

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	HCS12
Core Size	16-Bit
Speed	16MHz
Connectivity	CANbus, EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	60
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K 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	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12c32mfue16



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 1-3. MC9S12C96 and MC9S12GC96 User Configurable Memory Map

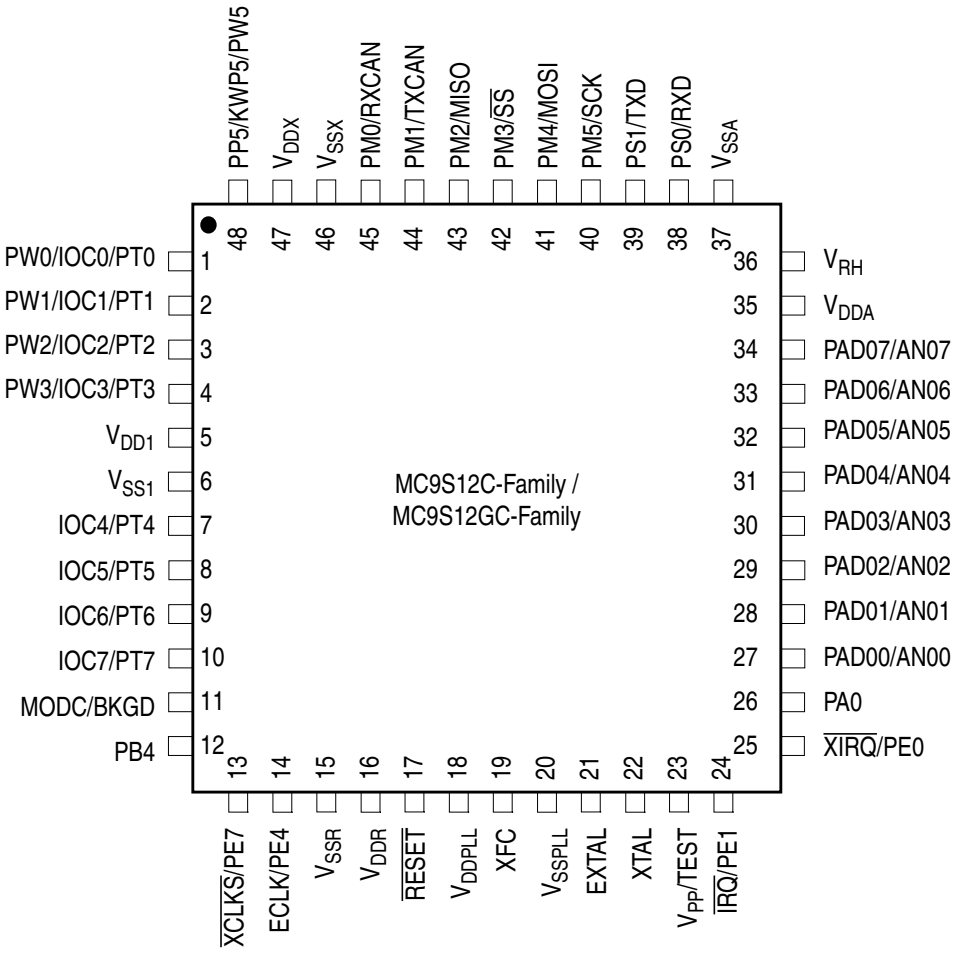


Figure 1-9. Pin Assignments in 48-Pin LQFP

2.3.2.4.5 Port P Pull Device Enable Register (PERP)

Module Base + 0x001C

	7	6	5	4	3	2	1	0
R W	PERP7	PERP6	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0
Reset	0	0	0	0	0	0	0	0

Figure 2-28. Port P Pull Device Enable Register (PERP)

Read: Anytime.

Write: Anytime.

Table 2-24. PERP Field Descriptions

Field	Description
7–0 PERP[7:0]	**Pull Device Enable Port P** — This register configures whether a pull-up or a pull-down device is activated, if the port is used as input. This bit has no effect if the port is used as output. Out of reset no pull device is enabled. 0 Pull-up or pull-down device is disabled. 1 Either a pull-up or pull-down device is enabled.

2.3.2.4.6 Port P Polarity Select Register (PPSP)

Module Base + 0x001D

	7	6	5	4	3	2	1	0
R W	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
Reset	0	0	0	0	0	0	0	0

Figure 2-29. Port P Polarity Select Register (PPSP)

Read: Anytime.

Write: Anytime.

Table 2-25. PPSP Field Descriptions

Field	Description
7–0 PPSP[7:0]	**Pull Select Port P** — This register serves a dual purpose by selecting the polarity of the active interrupt edge as well as selecting a pull-up or pull-down device if enabled. 0 Falling edge on the associated port P pin sets the associated flag bit in the PIFP register.A pull-up device is connected to the associated port P pin, if enabled by the associated bit in register PERP and if the port is used as input. 1 Rising edge on the associated port P pin sets the associated flag bit in the PIFP register.A pull-down device is connected to the associated port P pin, if enabled by the associated bit in register PERP and if the port is used as input.

Table 4-8. MODC, MODB, and MODA Write Capability[1]

MODC	MODB	MODA	Mode	MODx Write Capability
0	0	0	Special single chip	MODC, MODB, and MODA write anytime but not to 110[2]
0	0	1	Emulation narrow	No write
0	1	0	Special test	MODC, MODB, and MODA write anytime but not to 110[2]
0	1	1	Emulation wide	No write
1	0	0	Normal single chip	MODC write never, MODB and MODA write once but not to 110
1	0	1	Normal expanded narrow	No write
1	1	0	Special peripheral	No write
1	1	1	Normal expanded wide	No write

1. No writes to the MOD bits are allowed while operating in a secure mode. For more details, refer to the device overview chapter.
2. If you are in a special single-chip or special test mode and you write to this register, changing to normal single-chip mode, then one allowed write to this register remains. If you write to normal expanded or emulation mode, then no writes remain.

4.3.2.10 Pull Control Register (PUCR)

Module Base + 0x000C
Starting address location affected by INITRG register setting.

	7	6	5	4	3	2	1	0
R	PUPKE	0	0	PUPEE	0	0	PUPBE	PUPAE
W								
Reset[1]	1	0	0	1	0	0	0	0

NOTES:
1. The default value of this parameter is shown. Please refer to the device overview chapter to determine the actual reset state of this register.

= Unimplemented or Reserved

Figure 4-14. Pull Control Register (PUCR)

Read: Anytime (provided this register is in the map).

Write: Anytime (provided this register is in the map).

This register is used to select pull resistors for the pins associated with the core ports. Pull resistors are assigned on a per-port basis and apply to any pin in the corresponding port that is currently configured as an input. The polarity of these pull resistors is determined by chip integration. Please refer to the device overview chapter to determine the polarity of these resistors.

control (TBC) block. When PAGSEL = 01, registers DBGCAX, DBGCBX, and DBGCCX are used to match the upper addresses as shown in Table 7-11.

NOTE

If a tagged-type C breakpoint is set at the same address as an A/B tagged-type trigger (including the initial entry in an inside or outside range trigger), the C breakpoint will have priority and the trigger will not be recognized.

7.4.2.1.1 Read or Write Comparison

Read or write comparisons are useful only with TRGSEL = 0, because only opcodes should be tagged as they are “read” from memory. RWAEN and RWBEN are ignored when TRGSEL = 1.

In full modes (“A and B” and “A and not B”) RWAEN and RWA are used to select read or write comparisons for both comparators A and B. Table 7-24 shows the effect for RWAEN, RWA, and RW on the DBGCB comparison conditions. The RWBEN and RWB bits are not used and are ignored in full modes.

Table 7-24. Read or Write Comparison Logic Table

RWAEN bit	RWA bit	RW signal	Comment
0	x	0	Write data bus
0	x	1	Read data bus
1	0	0	Write data bus
1	0	1	No data bus compare since RW=1
1	1	0	No data bus compare since RW=0
1	1	1	Read data bus

7.4.2.1.2 Trigger Selection

The TRGSEL bit in DBGC1 is used to determine the triggering condition in DBG mode. TRGSEL applies to both trigger A and B except in the event only trigger modes. By setting TRGSEL, the comparators A and B will qualify a match with the output of opcode tracking logic and a trigger occurs before the tagged instruction executes (tagged-type trigger). With the TRGSEL bit cleared, a comparator match forces a trigger when the matching condition occurs (force-type trigger).

NOTE

If the TRGSEL is set, the address stored in the comparator match address registers must be an opcode address for the trigger to occur.

7.4.2.2 Trace Buffer Control (TBC)

The TBC is the main controller for the DBG module. Its function is to decide whether data should be stored in the trace buffer based on the trigger mode and the match signals from the comparator. The TBC also determines whether a request to break the CPU should occur.

Table 7-26. Breakpoint Setup

BEGIN	TRGSEL	DBGBRK	Type of Debug Run
0	0	0	Fill trace buffer until trigger address (no CPU breakpoint — keep running)
0	0	1	Fill trace buffer until trigger address, then a forced breakpoint request occurs
0	1	0	Fill trace buffer until trigger opcode is about to execute (no CPU breakpoint — keep running)
0	1	1	Fill trace buffer until trigger opcode about to execute, then a tagged breakpoint request occurs
1	0	0	Start trace buffer at trigger address (no CPU breakpoint — keep running)
1	0	1	Start trace buffer at trigger address, a forced breakpoint request occurs when trace buffer is full
1	1	0	Start trace buffer at trigger opcode (no CPU breakpoint — keep running)
1	1	1	Start trace buffer at trigger opcode, a forced breakpoint request occurs when trace buffer is full

7.4.3.2 Breakpoint Based on Comparator C

A breakpoint request to the CPU can be created if BKCEN in DBGC2 is set. Breakpoints based on a successful comparator C match can be accomplished regardless of the mode of operation for comparator A or B, and do not affect the status of the ARM bit. TAGC in DBGC2 is used to select either tagged or forced breakpoint requests for comparator C. Breakpoints based on comparator C are disabled in LOOP1 mode.

NOTE

Because breakpoints cannot be disabled when the DBG is armed, one must be careful to avoid an “infinite breakpoint loop” when using tagged-type C breakpoints while the DBG is armed. If BDM breakpoints are selected, executing a TRACE1 instruction before the GO instruction is the recommended way to avoid re-triggering a breakpoint if one does not wish to de-arm the DBG. If SWI breakpoints are selected, disarming the DBG in the SWI interrupt service routine is the recommended way to avoid re-triggering a breakpoint.

7.5 Resets

The DBG module is disabled after reset.

The DBG module cannot cause a MCU reset.

7.6 Interrupts

The DBG contains one interrupt source. If a breakpoint is requested and BDM in DBGC2 is cleared, an SWI interrupt will be generated.

8.3 Memory Map and Registers

This section provides a detailed description of all registers accessible in the ATD10B8C.

8.3.1 Module Memory Map

Figure 8-2 gives an overview on all ATD10B8C registers.

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000	ATDCTL0	R	0	0	0	0	0	0	0	0
		W								
0x0001	ATDCTL1	R	0	0	0	0	0	0	0	0
		W								
0x0002	ATDCTL2	R	ADPU	AFFC	AWAI	ETRIGLE	ETRIGP	ETRIGE	ASCIE	ASCIF
		W								
0x0003	ATDCTL3	R	0	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
		W								
0x0004	ATDCTL4	R	SRES8	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0
		W								
0x0005	ATDCTL5	R	DJM	DSGN	SCAN	MULT	0	CC	CB	CA
		W								
0x0006	ATDSTAT0	R	SCF	0	ETORF	FIFOR	0	CC2	CC1	CC0
		W								
0x0007	Unimplemented	R	0	0	0	0	0	0	0	0
		W								
0x0008	ATDTEST0	R	U	U	U	U	U	U	U	U
		W								
0x0009	ATDTEST1	R	U	U	U	U	U	U	U	SC
		W								
0x000A	Unimplemented	R	0	0	0	0	0	0	0	0
		W								
0x000B	ATDSTAT1	R	CCF7	CCF6	CCF5	CCF4	CCF3	CCF2	CCF1	CCF0
		W								
0x000C	Unimplemented	R	0	0	0	0	0	0	0	0
		W								
0x000D	ATDDIEN	R	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
		W								
0x000E	Unimplemented	R	0	0	0	0	0	0	0	0
		W								
0x000F	PORTAD	R	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
		W								

= Unimplemented or Reserved

Figure 8-2. ATD Register Summary (Sheet 1 of 4)

8.7 Interrupts

The interrupt requested by the ATD10B8C is listed in Table 8-20. Refer to MCU specification for related vector address and priority.

Table 8-20. ATD10B8C Interrupt Vectors

Interrupt Source	CCR Mask	Local Enable
Sequence complete interrupt	I bit	ASCIE in ATDCTL2

See Section 8.3.2, “Register Descriptions” for further details.

12.4.2.5 Left Aligned Outputs

The PWM timer provides the choice of two types of outputs, left aligned or center aligned outputs. They are selected with the CAEx bits in the PWMCAE register. If the CAEx bit is cleared (CAEx = 0), the corresponding PWM output will be left aligned.

In left aligned output mode, the 8-bit counter is configured as an up counter only. It compares to two registers, a duty register and a period register as shown in the block diagram in Figure 12-35. When the PWM counter matches the duty register the output flip-flop changes state causing the PWM waveform to also change state. A match between the PWM counter and the period register resets the counter and the output flip-flop as shown in Figure 12-35 as well as performing a load from the double buffer period and duty register to the associated registers as described in Section 12.4.2.3, "PWM Period and Duty." The counter counts from 0 to the value in the period register – 1.

NOTE

Changing the PWM output mode from left aligned output to center aligned output (or vice versa) while channels are operating can cause irregularities in the PWM output. It is recommended to program the output mode before enabling the PWM channel.

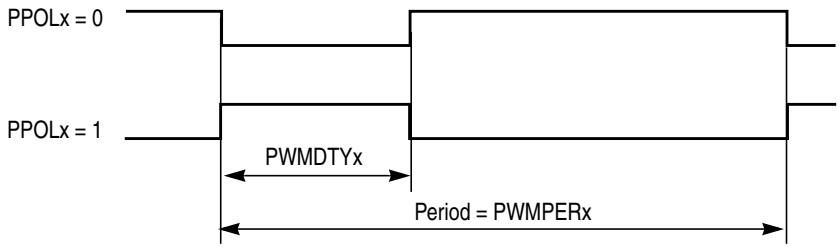


Figure 12-36. PWM Left Aligned Output Waveform

To calculate the output frequency in left aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB) and divide it by the value in the period register for that channel.

- PWMx frequency = clock (A, B, SA, or SB) / PWMPERx
- PWMx duty cycle (high time as a% of period):
  - Polarity = 0 (PPOLx = 0)
    Duty cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
  - Polarity = 1 (PPOLx = 1)
    Duty cycle = [PWMDTYx / PWMPERx] * 100%

As an example of a left aligned output, consider the following case:

Clock source = bus clock, where bus clock = 10 MHz (100 ns period)
PPOLx = 0
PWMPERx = 4
PWMDTYx = 1
PWMx frequency = 10 MHz/4 = 2.5 MHz
PWMx period = 400 ns
PWMx duty cycle = 3/4 *100% = 75%

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

Module Base + 0x0010 = TC0H, 0x0012 = TC1H, 0x0014 = TC2H, 0x0016 = TC3H, 0x0018 = TC4H, 0x001A = TC5H, 0x001C = TC6H, 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 15-22. Timer Input Capture/Output Compare Register x High (TCxH)

Module Base + 0x0011 = TC0L, 0x0013 = TC1L, 0x0015 = TC2L, 0x0017 = TC3L, 0x0019 = TC4L, 0x001B = TC5L, 0x001D = TC6L, 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 15-23. Timer Input Capture/Output Compare Register x Low (TCxL)

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 takes place before low byte otherwise it will give a different result.

17.4.1.4 Illegal Flash Operations

17.4.1.4.1 Access Error

The ACCERR flag in the FSTAT register will be set during the command write sequence if any of the following illegal Flash operations are performed causing the command write sequence to immediately abort:

1. Writing to the Flash address space before initializing the FCLKDIV register
2. Writing a misaligned word or a byte to the valid Flash address space
3. Writing to the Flash address space while CBEIF is not set
4. Writing a second word to the Flash address space before executing a program or erase command on the previously written word
5. Writing to any Flash register other than FCMD after writing a word to the Flash address space
6. Writing a second command to the FCMD register before executing the previously written command
7. Writing an invalid command to the FCMD register
8. Writing to any Flash register other than FSTAT (to clear CBEIF) after writing to the FCMD register
9. The part enters stop mode and a program or erase command is in progress. The command is aborted and any pending command is killed
10. When security is enabled, a command other than mass erase originating from a non-secure memory or from the background debug mode is written to the FCMD register
11. A 0 is written to the CBEIF bit in the FSTAT register to abort a command write sequence.

The ACCERR flag will not be set if any Flash register is read during the command write sequence. If the Flash array is read during execution of an algorithm (CCIF=0), the Flash module will return invalid data and the ACCERR flag will not be set. If an ACCERR flag is set in the FSTAT register, the Flash command controller is locked. It is not possible to launch another command until the ACCERR flag is cleared.

17.4.1.4.2 Protection Violation

The PVIOL flag in the FSTAT register will be set during the command write sequence after the word write to the Flash address space if any of the following illegal Flash operations are performed, causing the command write sequence to immediately abort:

1. Writing a Flash address to program in a protected area of the Flash array (see Section 17.3.2.5).
2. Writing a Flash address to erase in a protected area of the Flash array.
3. Writing the mass erase command to the FCMD register while any protection is enabled.

If the PVIOL flag is set, the Flash command controller is locked. It is not possible to launch another command until the PVIOL flag is cleared.

18.4.1.3.3 Sector Erase Command

The sector erase operation will erase all addresses in a 512 byte sector of the Flash array using an embedded algorithm.

An example flow to execute the sector erase operation is shown in Figure 18-24. The sector erase command write sequence is as follows:

1. Write to a Flash array address to start the command write sequence for the sector erase command. The Flash address written determines the sector to be erased while MCU address bits [8:0] and the data written are ignored.
2. Write the sector erase command, 0x40, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the sector erase command.

If a Flash sector to be erased is in a protected area of the Flash array, the PVIOL flag in the FSTAT register will set and the sector erase command will not launch. Once the sector erase command has successfully launched, the CCIF flag in the FSTAT register will set after the sector erase operation has completed unless a new command write sequence has been buffered.

18.4.2 Operating Modes

18.4.2.1 Wait Mode

If the MCU enters wait mode while a Flash command is active (CCIF = 0), that command and any buffered command will be completed.

The Flash module can recover the MCU from wait mode if the interrupts are enabled (see Section 18.4.5).

18.4.2.2 Stop Mode

If the MCU enters stop mode while a Flash command is active (CCIF = 0), that command will be aborted and the data being programmed or erased is lost. The high voltage circuitry to the Flash array will be switched off when entering stop mode. CCIF and ACCERR flags will be set. Upon exit from stop mode, the CBEIF flag will be set and any buffered command will not be executed. The ACCERR flag must be cleared before returning to normal operation.

NOTE

As active Flash commands are immediately aborted when the MCU enters stop mode, it is strongly recommended that the user does not use the STOP instruction during program and erase execution.

18.4.2.3 Background Debug Mode

In background debug mode (BDM), the FPROT register is writable. If the MCU is unsecured, then all Flash commands listed in Table 18-16 can be executed. If the MCU is secured and is in special single chip mode, the only possible command to execute is mass erase.

18.4.3 Flash Module Security

The Flash module provides the necessary security information to the MCU. After each reset, the Flash module determines the security state of the MCU as defined in Section 18.3.2.2, “Flash Security Register (FSEC)”.

The contents of the Flash security/options byte at address 0xFF0F in the Flash configuration field must be changed directly by programming address 0xFF0F when the device is unsecured and the higher address sector is unprotected. If the Flash security/options byte is left in the secure state, any reset will cause the MCU to return to the secure operating mode.

18.4.3.1 Unsecuring the MCU using Backdoor Key Access

The MCU may only be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor key (four 16-bit words programmed at addresses 0xFF00–0xFF07). If KEYEN[1:0] = 1:0 and the KEYACC bit is set, a write to a backdoor key address in the Flash array triggers a comparison between the written data and the backdoor key data stored in the Flash array. If all four words of data are written to the correct addresses in the correct order and the data matches the backdoor key stored in the Flash array, the MCU will be unsecured. The data must be written to the backdoor key

21.4.1.3.1 Erase Verify Command

The erase verify operation will verify that a Flash array is erased.

An example flow to execute the erase verify operation is shown in Figure 21-22. The erase verify command write sequence is as follows:

1. Write to a Flash array address to start the command write sequence for the erase verify command. The address and data written will be ignored.
2. Write the erase verify command, 0x05, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the erase verify command.

After launching the erase verify command, the CCIF flag in the FSTAT register will set after the operation has completed unless a new command write sequence has been buffered. Upon completion of the erase verify operation, the BLANK flag in the FSTAT register will be set if all addresses in the Flash array are verified to be erased. If any address in the Flash array is not erased, the erase verify operation will terminate and the BLANK flag in the FSTAT register will remain clear.

A.5.2 NVM Reliability

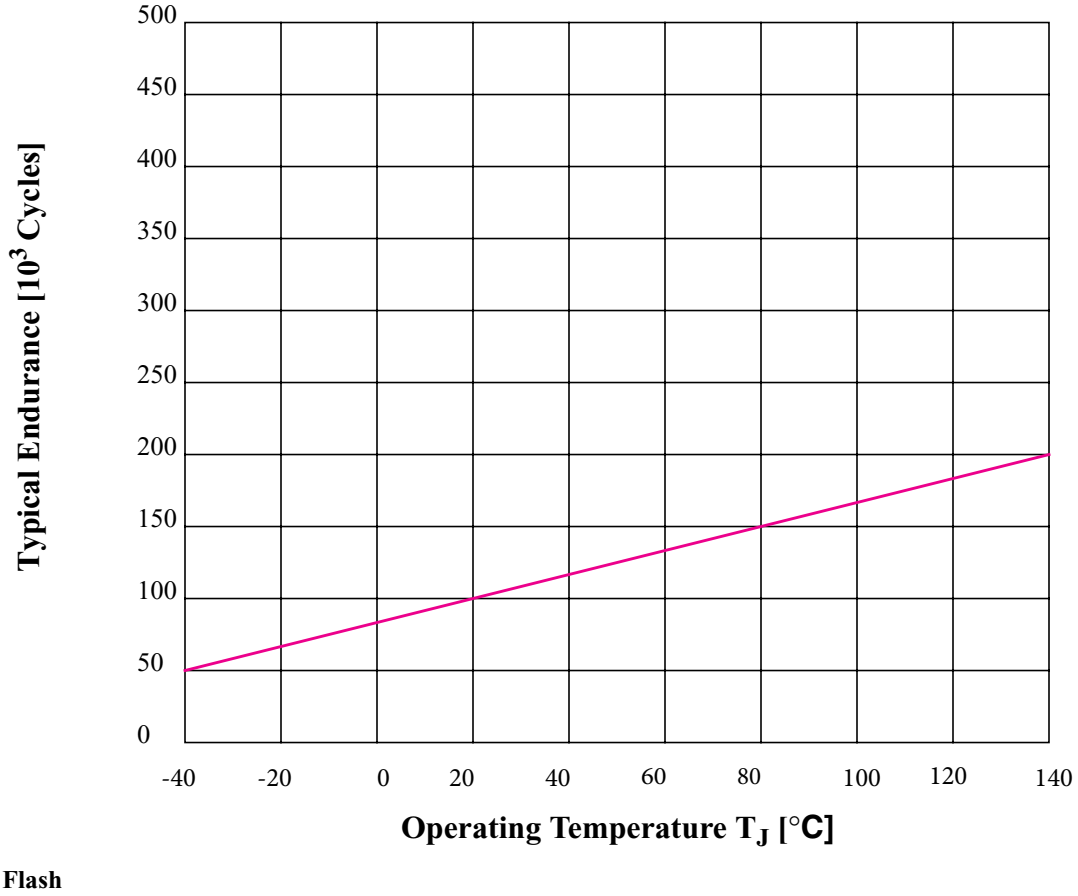
The reliability of the NVM blocks is guaranteed by stress test during qualification, constant process monitors and burn-in to screen early life failures. The program/erase cycle count on the sector is incremented every time a sector or mass erase event is executed.

Table A-19. NVM Reliability Characteristics[1]

Conditions are shown in Table A-4. unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
Flash Reliability Characteristics							
1	C	Data retention after 10,000 program/erase cycles at an average junction temperature of $T_{Javg} \leq 85°C$	t_{FLRET}	15	100[2]	—	Years
2	C	Data retention with <100 program/erase cycles at an average junction temperature $T_{Javg} \leq 85°C$		20	100[2]	—	
3	C	Number of program/erase cycles ($-40°C \leq T_J \leq 0°C$)	n_{FL}	10,000	—	—	Cycles
4	C	Number of program/erase cycles ($0°C \leq T_J \leq 140°C$)		10,000	100,000[3]	—	

1. T_{Javg} will not exeed 85°C considering a typical temperature profile over the lifetime of a consumer, industrial or automotive application.
2. Typical data retention values are based on intrinsic capability of the technology measured at high temperature and de-rated to 25°C using the Arrhenius equation. For additional information on how Freescale defines Typical Data Retention, please refer to Engineering Bulletin EB618.
3. Spec table quotes typical endurance evaluated at 25°C for this product family, typical endurance at various temperature can be estimated using the graph below. For additional information on how Freescale defines Typical Endurance, please refer to Engineering Bulletin EB619.

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.

Appendix C
Package Information

C.1 General

This section provides the physical dimensions of the packages 48LQFP, 52LQFP, 80QFP.

Appendix D
Derivative Differences

The Device User Guide provides information about the MC9S12C-Family and the MC9S12GC-Family. The C-Family and the GC-Family offer an extensive range of package, temperature and speed options. The members of the GC-Family are a subset of the C-family that do not feature a CAN module.

Table D-1. shows a feature overview of the C and GC family members.

Table D-1. List of MC9S12C and MC9S12GC Family members(1)

Flash	RAM	Device	CAN	SCI	SPI	A/D	PWM	Timer
128K	4K	MC9S12C128	1	1	1	8ch	6ch	8ch
		MC9S12GC128	—	1	1	8ch	6ch	8ch
96K	4K	MC9S12C96	1	1	1	8ch	6ch	8ch
		MC9S12GC96	—	1	1	8ch	6ch	8ch
64K	4K	MC9S12C64	1	1	1	8ch	6ch	8ch
		MC9S12GC64	—	1	1	8ch	6ch	8ch
32K	2K	MC9S12C32	1	1	1	8ch	6ch	8ch
		MC9S12GC32	—	1	1	8ch	6ch	8ch
16K	1K	MC9S12GC16	—	1	1	8ch	6ch	8ch

1. All family memebers are available in 80QFP, 52LQFP and 48LQFP package options