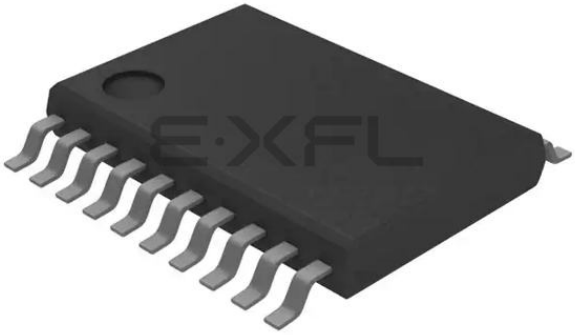


Welcome to E-XFL.COM

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	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	16
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	20-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s08sl8ctj

Chapter 3
Modes of Operation

3.1 Introduction

The operating modes of the MC9S08EL32 Series and MC9S08SL16 Series are described in this chapter. Entry into each mode, exit from each mode, and functionality while in each of the modes is described.

3.2 Features

- Active background mode for code development
- Wait mode — CPU shuts down to conserve power; system clocks are running and full regulation is maintained
- Stop modes — System clocks are stopped and voltage regulator is in standby
  - Stop3 — All internal circuits are powered for fast recovery; RAM and register contents are retained
  - Stop2 — Partial power down of internal circuits; RAM content is retained

3.3 Run Mode

This is the normal operating mode for the MC9S08EL32 Series and MC9S08SL16 Series. This mode is selected when the BKGD/MS pin is high at the rising edge of reset. In this mode, the CPU executes code from internal memory with execution beginning at the address fetched from memory at 0xFFFE–0xFFFF after reset.

3.4 Active Background Mode

The active background mode functions are managed through the background debug controller (BDC) in the HCS08 core. The BDC, together with the on-chip debug module (DBG), provide the means for analyzing MCU operation during software development.

Active background mode is entered in any of five ways:

- When the BKGD/MS pin is low at the rising edge of reset
- When a BACKGROUND command is received through the BKGD/MS pin
- When a BGND instruction is executed
- When encountering a BDC breakpoint
- When encountering a DBG breakpoint

After entering active background mode, the CPU is held in a suspended state waiting for serial background commands rather than executing instructions from the user application program.

Table 3-1 shows all of the control bits that affect stop mode selection and the mode selected under various conditions. The selected mode is entered following the execution of a STOP instruction.

Table 3-1. Stop Mode Selection

STOPE	ENBDM [1]	LVDE	LVDSE	PPDC	Stop Mode
0	x	x		x	Stop modes disabled; illegal opcode reset if STOP instruction executed
1	1	x		x	Stop3 with BDM enabled [2]
1	0	Both bits must be 1		0	Stop3 with voltage regulator active
1	0	Either bit a 0		0	Stop3
1	0	Either bit a 0		1	Stop2

1 ENBDM is located in the BDCSCR, which is only accessible through BDC commands, see Section 17.4.1.1, "BDC Status and Control Register (BDCSCR)".

2 When in Stop3 mode with BDM enabled, The S_{IDD} will be near R_{IDD} levels because internal clocks are enabled.

3.6.1 Stop3 Mode

Stop3 mode is entered by executing a STOP instruction under the conditions as shown in Table 3-1. The states of all of the internal registers and logic, RAM contents, and I/O pin states are maintained.

Exit from stop3 is done by asserting $\overline{\text{RESET}}$, or an asynchronous interrupt pin. The asynchronous interrupt pins are PIA0-PIA3, PIB0 -PIB3, and PIC0-PIC7. Exit from stop3 can also be done by the low-voltage detection (LVD) reset, the low-voltage warning (LVW) interrupt, the ADC conversion complete interrupt, the analog comparator (ACMP) interrupt, the real-time counter (RTC) interrupt, the SLIC wake-up interrupt, or the SCI receiver interrupt.

If stop3 is exited by means of the $\overline{\text{RESET}}$ pin, the MCU will be reset and operation will resume after fetching the reset vector. Exit by means of an asynchronous interrupt, analog comparator interrupt, or the real-time interrupt will result in the MCU fetching the appropriate interrupt vector.

3.6.1.1 LVD Enabled in Stop Mode

The LVD system is capable of generating either an interrupt or a reset when the supply voltage drops below the LVD voltage. If the LVD is enabled in stop (LVDE and LVDSE bits in SPMSC1 both set) at the time the CPU executes a STOP instruction, then the voltage regulator remains active during stop mode.

For the ADC to operate the LVD must be left enabled when entering stop3.

3.6.1.2 Active BDM Enabled in Stop Mode

Entry into the active background mode from run mode is enabled if ENBDM in BDCSCR is set. This register is described in Chapter 17, "Development Support." If ENBDM is set when the CPU executes a STOP instruction, the system clocks to the background debug logic remain active when the MCU enters stop mode. Because of this, background debug communication remains possible. In addition, the voltage regulator does not enter its low-power standby state but maintains full internal regulation.

High-page registers, shown in Table 4-3, are accessed much less often than other I/O and control registers so they have been located outside the direct addressable memory space, starting at 0x1800.

Table 4-3. High-Page Register Summary (Sheet 1 of 2)

Address	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
0x1800	SRS	POR	PIN	COP	ILOP	ILAD	0	LVD	0
0x1801	SBDFR	0	0	0	0	0	0	0	BDFR
0x1802	SOPT1	COPT		STOPE	SCIPS	IICPS		0	0
0x1803	SOPT2	COPCLKS	COPW	0	ACIC	T2CH1PS	T2CH0PS	T1CH1PS	T1CH0PS
0x1804 – 0x1805	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
0x1806	SDIDH	—	—	—	—	ID11	ID10	ID9	ID8
0x1807	SDIDL	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
0x1808	Reserved	—	—	—	—	—	—	—	—
0x1809	SPMSC1	LVWF	LVWACK	LVWIE	LVDRE	LVDSE	LVDE	0	BGBE
0x180A	SPMSC2	0	0	LVDV	LVWV	PPDF	PPDACK	—	PPDC
0x180B– 0x180F	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
0x1810	DBGCAH	Bit 15	14	13	12	11	10	9	Bit 8
0x1811	DBGCAL	Bit 7	6	5	4	3	2	1	Bit 0
0x1812	DBGCBH	Bit 15	14	13	12	11	10	9	Bit 8
0x1813	DBGCBL	Bit 7	6	5	4	3	2	1	Bit 0
0x1814	DBGFH	Bit 15	14	13	12	11	10	9	Bit 8
0x1815	DBGFL	Bit 7	6	5	4	3	2	1	Bit 0
0x1816	DBGC	DBGEN	ARM	TAG	BRKEN	RWA	RWAEN	RWB	RWBEN
0x1817	DBGT	TRGSEL	BEGIN	0	0	TRG3	TRG2	TRG1	TRG0
0x1818	DBGS	AF	BF	ARMF	0	CNT3	CNT2	CNT1	CNT0
0x1819– 0x181F	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
0x1820	FCDIV	DIVLD	PRDIV8	DIV					
0x1821	FOPT	KEYEN	FNORED	EPGMOD	0	0	0	SEC	
0x1822	Reserved	0	0	0	0	0	0	0	0
0x1823	FCNFG	0	EPGSEL	KEYACC	0	0	0	0	0
0x1824	FPROT	EPS		FPS					FPOP
0x1825	FSTAT	FCBEF	FCCF	FPVIOL	FACCERR	0	FBLANK	0	0
0x1826	FCMD	FCMD							
0x1827– 0x183F	Reserved	— —	— —	— —	— —	— —	— —	— —	— —
0x1840	PTAPE	PTAPE7	PTAPE6	0	0	PTAPE3	PTAPE2	PTAPE1	PTAPE0
0x1841	PTASE	PTASE7	PTASE6	0	0	PTASE3	PTASE2	PTASE1	PTASE0
0x1842	PTADS	PTADS7	PTADS6	0	0	PTADS3	PTADS2	PTADS1	PTADS0
0x1843	Reserved	—	—	—	—	—	—	—	—
0x1844	PTASC	0	0	0	0	PTAIF	PTAACK	PTAIE	PTAMOD

4.5 FLASH and EEPROM

The MC9S08EL32 Series and MC9S08SL16 Series includes FLASH and EEPROM memory intended primarily for program and data storage. In-circuit programming allows the operating program and data to be loaded into FLASH and EEPROM, respectively, after final assembly of the application product. It is possible to program the arrays through the single-wire background debug interface. Because no special voltages are needed for erase and programming operations, in-application programming is also possible through other software-controlled communication paths. For a more detailed discussion of in-circuit and in-application programming, refer to the *HCS08 Family Reference Manual, Volume I,* Freescale Semiconductor document order number HCS08RMv1/D.

4.5.1 Features

Features of the FLASH and EEPROM memory include:

- Array size
  - MC9S08EL32: 32,768 bytes of FLASH, 512 bytes of EEPROM
  - MC9S08EL16: 16,384 bytes of FLASH, 512 bytes of EEPROM
  - MC9S08SL16: 16,384 bytes of FLASH, 256 bytes of EEPROM
  - MC9S08SL8: 8,192 bytes of FLASH, 256 bytes of EEPROM
- Sector size: 512 bytes for FLASH, 8 bytes for EEPROM
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 100,000 program/erase cycles at typical voltage and temperature
- Flexible block protection and vector redirection
- Security feature for FLASH, EEPROM, and RAM

4.5.2 Program and Erase Times

Before any program or erase command can be accepted, the FLASH and EEPROM clock divider register (FCDIV) must be written to set the internal clock for the FLASH and EEPROM module to a frequency (f_{FCLK}) between 150 kHz and 200 kHz (see Section 4.5.11.1, “FLASH and EEPROM Clock Divider Register (FCDIV)”). This register can be written only once, so normally this write is performed during reset initialization. FCDIV cannot be written if the access error flag, FACCERR in FSTAT, is set. The user must ensure that FACCERR is not set before writing to the FCDIV register. One period of the resulting clock ($1/f_{FCLK}$) is used by the command processor to time program and erase pulses. An integer number of these timing pulses is used by the command processor to complete a program or erase command.

Table 4-5 shows program and erase times. The bus clock frequency and FCDIV determine the frequency of FCLK (f_{FCLK}). The time for one cycle of FCLK is $t_{FCLK} = 1/f_{FCLK}$. The times are shown as a number of cycles of FCLK and as an absolute time for the case where $t_{FCLK} = 5$ µs. Program and erase times shown include overhead for the command state machine and enabling and disabling of program and erase voltages.

The first byte of a series of sequential bytes being programmed in burst mode will take the same amount of time to program as a byte programmed in standard mode. Subsequent bytes will program in the burst program time provided that the conditions above are met. If the next sequential address is the beginning of a new row, the program time for that byte will be the standard time instead of the burst time. This is because the high voltage to the array must be disabled and then enabled again. If a new burst command has not been queued before the current command completes, then the charge pump will be disabled and high voltage removed from the array.

A flowchart to execute the burst program operation is shown in Figure 4-3.

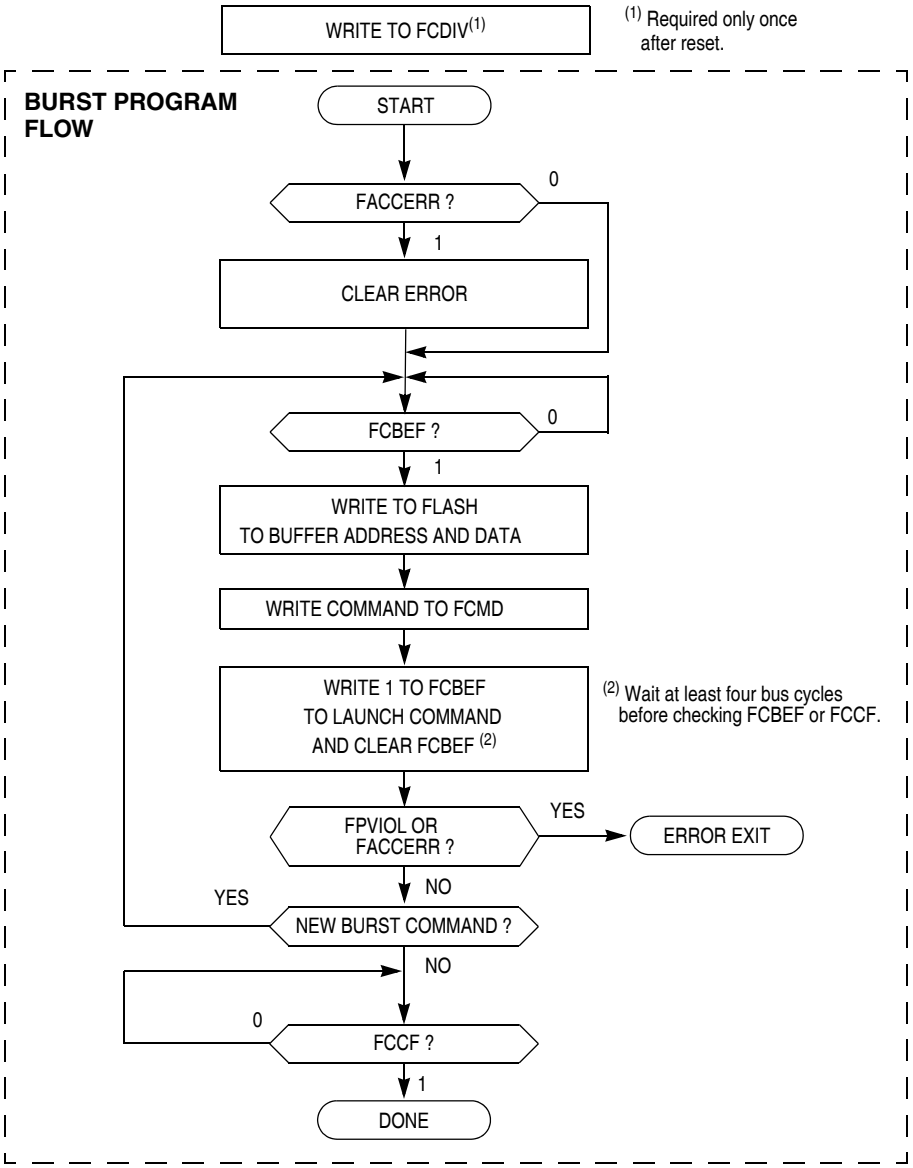


Figure 4-3. Burst Program Flowchart

Table 5-2. Vector Summary

Vector Priority	Vector Number	Address (High/Low)	Vector Name	Module	Source	Enable	Description
Lowest	31	0xFFC0/0xFFC1	Vacmp2	ACMP2	ACF	ACIE	Analog comparator 2
	30	0xFFC2/0xFFC3	Vacmp1	ACMP1	ACF	ACIE	Analog comparator 1
	29	0xFFC4/0xFFC5	—	—	—	—	—
	28	0xFFC6/0xFFC7	—	—	—	—	—
	27	0xFFC8/0xFFC9	—	—	—	—	—
	26	0xFFCA/0xFFCB	—	—	—	—	—
	25	0xFFCC/0xFFCD	Vrtc	RTC	RTIF	RTIE	Real-time interrupt
	24	0xFFCE/0xFFCF	Viic	IIC	IICIS	IICIE	IIC control
	23	0xFFD0/0xFFD1	Vadc	ADC	COCO	AIEN	ADC
	22	0xFFD2/0xFFD3	Vportc	Port C	PTCIF	PTCIE	Port C Pins
	21	0xFFD4/0xFFD5	Vportb	Port B	PTBIF	PTBIE	Port B Pins
	20	0xFFD6/0xFFD7	Vporta	Port A	PTAIF	PTAIE	Port A Pins
	19	0xFFD8/0xFFD9	Vslic	SLIC	SLCF	SLCIE	SLIC
	18	0xFFDA/0xFFDB	Vscitx	SCI	TDRE, TC	TIE, TCIE	SCI transmit
	17	0xFFDC/0xFFDD	Vscirx	SCI	IDLE, LBKDIF, RDRF, RXEDGIF	ILIE, LBKDIE, RIE, RXEDGIE	SCI receive
	16	0xFFDE/0xFFDF	Vscierr	SCI	OR, NF, FE, PF	ORIE, NFIE, FEIE, PFIE	SCI error
	15	0xFFE0/0xFFE1	Vspi	SPI	SPIF, MODF, SPTEF	SPIE, SPIE, SPTIE	SPI
	14	0xFFE2/0xFFE3	Vtpm2ovf	TPM2	TOF	TOIE	TPM2 overflow
	13	0xFFE4/0xFFE5	Vtpm2ch1	TPM2	CH1F	CH1IE	TPM2 channel 1
	12	0xFFE6/0xFFE7	Vtpm2ch0	TPM2	CH0F	CH0IE	TPM2 channel 0
	11	0xFFE8/0xFFE9	Vtpm1ovf	TPM1	TOF	TOIE	TPM1 overflow
	10	0xFFEA/0xFFEB	—	—	—	—	—
	9	0xFFEC/0xFFED	—	—	—	—	—
	8	0xFFEE/0xFFEF	Vtpm1ch3	TPM1	CH3F	CH3IE	TPM1 channel 3
	7	0xFFF0/0xFFF1	Vtpm1ch2	TPM1	CH2F	CH2IE	TPM1 channel 2
	6	0xFFF2/0xFFF3	Vtpm1ch1	TPM1	CH1F	CH1IE	TPM1 channel 1
	5	0xFFF4/0xFFF5	Vtpm1ch0	TPM1	CH0F	CH0IE	TPM1 channel 0
	4	0xFFF6/0xFFF7	—	—	—	—	—
	3	0xFFF8/0xFFF9	Vlvd	System control	LVWF	LVWIE	Low-voltage warning
	2	0xFFFA/0xFFFB	—	—	—	—	—
	1	0xFFFC/0xFFFD	Vswi	Core	SWI Instruction	—	Software interrupt
Highest	0	0xFFFE/0xFFFF	Vreset	System control	COP, LVD, $\overline{\text{RESET}}$ pin, Illegal opcode, Illegal address	COPT LVDRE — — —	Watchdog timer Low-voltage detect External pin Illegal opcode Illegal address

5.6 Low-Voltage Detect (LVD) System

The MC9S08EL32 Series and MC9S08SL16 Series includes a system to protect against low voltage conditions in order to protect memory contents and control MCU system states during supply voltage variations. The system is comprised of a power-on reset (POR) circuit and a LVD circuit with trip voltages

5.7.6 System Power Management Status and Control 1 Register (SPMSC1)

This high page register contains status and control bits to support the low voltage detect function, and to enable the bandgap voltage reference for use by the ADC module.

	7	6	5	4	3	2	1	0
R	LVWF[1]	0	LVWIE	LVDRE[2]	LVDSE[2]	LVDE[2]	0	BGBE
W		LVWACK						
Reset:	0	0	0	1	1	1	0	0

= Unimplemented or Reserved

1 LVWF will be set in the case when V_{Supply} transitions below the trip point or after reset and V_{Supply} is already below V_{LVW}

2 This bit can be written only one time after reset. Additional writes are ignored.

Figure 5-8. System Power Management Status and Control 1 Register (SPMSC1)

Table 5-9. SPMSC1 Register Field Descriptions

Field	Description
7 LVWF	**Low-Voltage Warning Flag** — The LVWF bit indicates the low voltage warning status. 0 Low voltage warning is not present. 1 Low voltage warning is present or was present.
6 LVWACK	**Low-Voltage Warning Acknowledge** — The LVWF bit indicates the low voltage warning status.Writing a 1 to LVWACK clears LVWF to a 0 if a low voltage warning is not present.
5 LVWIE	**Low-Voltage Warning Interrupt Enable** — This bit enables hardware interrupt requests for LVWF. 0 Hardware interrupt disabled (use polling). 1 Request a hardware interrupt when LVWF = 1.
4 LVDRE	**Low-Voltage Detect Reset Enable** — This write-once bit enables LVD events to generate a hardware reset (provided LVDE = 1). 0 LVD events do not generate hardware resets. 1 Force an MCU reset when an enabled low-voltage detect event occurs.
3 LVDSE	**Low-Voltage Detect Stop Enable** — Provided LVDE = 1, this read/write bit determines whether the low-voltage detect function operates when the MCU is in stop mode. 0 Low-voltage detect disabled during stop mode. 1 Low-voltage detect enabled during stop mode.
2 LVDE	**Low-Voltage Detect Enable** — This write-once bit enables low-voltage detect logic and qualifies the operation of other bits in this register. 0 LVD logic disabled. 1 LVD logic enabled.
0 BGBE	**Bandgap Buffer Enable** — This bit enables an internal buffer for the bandgap voltage reference for use by the ADC module on one of its internal channels. 0 Bandgap buffer disabled. 1 Bandgap buffer enabled.

Chapter 6
Parallel Input/Output Control

This section explains software controls related to parallel input/output (I/O) and pin control. The MC9S08EL32 has three parallel I/O ports which include a total of 22 I/O pins. See Chapter 2, “Pins and Connections,” for more information about pin assignments and external hardware considerations of these pins.

Many of these pins are shared with on-chip peripherals such as timer systems, communication systems, or keyboard interrupts as shown in Table 2-1. The peripheral modules have priority over the general-purpose I/O functions so that when a peripheral is enabled, the I/O functions associated with the shared pins are disabled.

After reset, the shared peripheral functions are disabled and the pins are configured as inputs (PTxDDn = 0). The pin control functions for each pin are configured as follows: slew rate control enabled (PTxSEn = 1), low drive strength selected (PTxDSn = 0), and internal pull-ups disabled (PTxPEn = 0).

NOTE

Not all general-purpose I/O pins are available on all packages. To avoid extra current drain from floating input pins, the user’s reset initialization routine in the application program must either enable on-chip pull-up devices or change the direction of unconnected pins to outputs so the pins do not float.

6.1 Port Data and Data Direction

Reading and writing of parallel I/Os are performed through the port data registers. The direction, either input or output, is controlled through the port data direction registers. The parallel I/O port function for an individual pin is illustrated in the block diagram shown in Figure 6-1.

The data direction control bit (PTxDDn) determines whether the output buffer for the associated pin is enabled, and also controls the source for port data register reads. The input buffer for the associated pin is always enabled unless the pin is enabled as an analog function or is an output-only pin.

When a shared digital function is enabled for a pin, the output buffer is controlled by the shared function. However, the data direction register bit will continue to control the source for reads of the port data register.

When a shared analog function is enabled for a pin, both the input and output buffers are disabled. A value of 0 is read for any port data bit where the bit is an input (PTxDDn = 0) and the input buffer is disabled. In general, whenever a pin is shared with both an alternate digital function and an analog function, the analog function has priority such that if both the digital and analog functions are enabled, the analog function controls the pin.

6.5.3.5 Port C Drive Strength Selection Register (PTCDS)

	7	6	5	4	3	2	1	0
R W	PTCDS7	PTCDS6	PTCDS5	PTCDS4	PTCDS3	PTCDS2	PTCDS1	PTCDS0
Reset:	0	0	0	0	0	0	0	0

Figure 6-23. Drive Strength Selection for Port C Register (PTCDS)

Table 6-21. PTCDS Register Field Descriptions

Field	Description
7:0 PTCDS[7:0]	**Output Drive Strength Selection for Port C Bits** — Each of these control bits selects between low and high output drive for the associated PTC pin. For port C pins that are configured as inputs, these bits have no effect. 0 Low output drive strength selected for port C bit n. 1 High output drive strength selected for port C bit n.

6.5.3.6 Port C Interrupt Status and Control Register (PTCSC)

	7	6	5	4	3	2	1	0
R	0	0	0	0	PTCIF	0	PTCIE	PTCMOD
W						PTCACK		
Reset:	0	0	0	0	0	0	0	0

Figure 6-24. Port C Interrupt Status and Control Register (PTCSC)

Table 6-22. PTCSC Register Field Descriptions

Field	Description
3 PTCIF	**Port C Interrupt Flag** — PTCIF indicates when a port D interrupt is detected. Writes have no effect on PTCIF. 0 No port C interrupt detected. 1 Port C interrupt detected.
2 PTCACK	**Port C Interrupt Acknowledge** — Writing a 1 to PTCACK is part of the flag clearing mechanism. PTCACK always reads as 0.
1 PTCIE	**Port C Interrupt Enable** — PTCIE determines whether a port C interrupt is requested. 0 Port C interrupt request not enabled. 1 Port C interrupt request enabled.
0 PTCMOD	**Port C Detection Mode** — PTCMOD (along with the PTCES bits) controls the detection mode of the port C interrupt pins. 0 Port C pins detect edges only. 1 Port C pins detect both edges and levels.

7.3.5 Extended Addressing Mode (EXT)

In extended addressing mode, the full 16-bit address of the operand is located in the next two bytes of program memory after the opcode (high byte first).

7.3.6 Indexed Addressing Mode

Indexed addressing mode has seven variations including five that use the 16-bit H:X index register pair and two that use the stack pointer as the base reference.

7.3.6.1 Indexed, No Offset (IX)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair as the address of the operand needed to complete the instruction.

7.3.6.2 Indexed, No Offset with Post Increment (IX+)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair as the address of the operand needed to complete the instruction. The index register pair is then incremented (H:X = H:X + 0x0001) after the operand has been fetched. This addressing mode is only used for MOV and CBEQ instructions.

7.3.6.3 Indexed, 8-Bit Offset (IX1)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair plus an unsigned 8-bit offset included in the instruction as the address of the operand needed to complete the instruction.

7.3.6.4 Indexed, 8-Bit Offset with Post Increment (IX1+)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair plus an unsigned 8-bit offset included in the instruction as the address of the operand needed to complete the instruction. The index register pair is then incremented (H:X = H:X + 0x0001) after the operand has been fetched. This addressing mode is used only for the CBEQ instruction.

7.3.6.5 Indexed, 16-Bit Offset (IX2)

This variation of indexed addressing uses the 16-bit value in the H:X index register pair plus a 16-bit offset included in the instruction as the address of the operand needed to complete the instruction.

7.3.6.6 SP-Relative, 8-Bit Offset (SP1)

This variation of indexed addressing uses the 16-bit value in the stack pointer (SP) plus an unsigned 8-bit offset included in the instruction as the address of the operand needed to complete the instruction.

7.3.6.7 SP-Relative, 16-Bit Offset (SP2)

This variation of indexed addressing uses the 16-bit value in the stack pointer (SP) plus a 16-bit offset included in the instruction as the address of the operand needed to complete the instruction.

7.4 Special Operations

The CPU performs a few special operations that are similar to instructions but do not have opcodes like other CPU instructions. In addition, a few instructions such as STOP and WAIT directly affect other MCU circuitry. This section provides additional information about these operations.

7.4.1 Reset Sequence

Reset can be caused by a power-on-reset (POR) event, internal conditions such as the COP (computer operating properly) watchdog, or by assertion of an external active-low reset pin. When a reset event occurs, the CPU immediately stops whatever it is doing (the MCU does not wait for an instruction boundary before responding to a reset event). For a more detailed discussion about how the MCU recognizes resets and determines the source, refer to the Resets, Interrupts, and System Configuration chapter.

The reset event is considered concluded when the sequence to determine whether the reset came from an internal source is done and when the reset pin is no longer asserted. At the conclusion of a reset event, the CPU performs a 6-cycle sequence to fetch the reset vector from 0xFFFE and 0xFFFF and to fill the instruction queue in preparation for execution of the first program instruction.

7.4.2 Interrupt Sequence

When an interrupt is requested, the CPU completes the current instruction before responding to the interrupt. At this point, the program counter is pointing at the start of the next instruction, which is where the CPU should return after servicing the interrupt. The CPU responds to an interrupt by performing the same sequence of operations as for a software interrupt (SWI) instruction, except the address used for the vector fetch is determined by the highest priority interrupt that is pending when the interrupt sequence started.

The CPU sequence for an interrupt is:

1. Store the contents of PCL, PCH, X, A, and CCR on the stack, in that order.
2. Set the I bit in the CCR.
3. Fetch the high-order half of the interrupt vector.
4. Fetch the low-order half of the interrupt vector.
5. Delay for one free bus cycle.
6. Fetch three bytes of program information starting at the address indicated by the interrupt vector to fill the instruction queue in preparation for execution of the first instruction in the interrupt service routine.

After the CCR contents are pushed onto the stack, the I bit in the CCR is set to prevent other interrupts while in the interrupt service routine. Although it is possible to clear the I bit with an instruction in the

10.6.2 Sources of Error

Several sources of error exist for A/D conversions. These are discussed in the following sections.

10.6.2.1 Sampling Error

For proper conversions, the input must be sampled long enough to achieve the proper accuracy. Given the maximum input resistance of approximately 7kΩ and input capacitance of approximately 5.5 pF, sampling to within 1/4LSB (at 10-bit resolution) can be achieved within the minimum sample window (3.5 cycles @ 8 MHz maximum ADCK frequency) provided the resistance of the external analog source (R_{AS}) is kept below 5 kΩ.

Higher source resistances or higher-accuracy sampling is possible by setting ADLSMP (to increase the sample window to 23.5 cycles) or decreasing ADCK frequency to increase sample time.

10.6.2.2 Pin Leakage Error

Leakage on the I/O pins can cause conversion error if the external analog source resistance (R_{AS}) is high. If this error cannot be tolerated by the application, keep R_{AS} lower than V_{DDAD} / (2^N*I_{LEAK}) for less than 1/4LSB leakage error (N = 8 in 8-bit mode or 10 in 10-bit mode).

10.6.2.3 Noise-Induced Errors

System noise which occurs during the sample or conversion process can affect the accuracy of the conversion. The ADC accuracy numbers are guaranteed as specified only if the following conditions are met:

- There is a 0.1 µF low-ESR capacitor from V_{REFH} to V_{REFL}.
- There is a 0.1 µF low-ESR capacitor from V_{DDAD} to V_{SSAD}.
- If inductive isolation is used from the primary supply, an additional 1 µF capacitor is placed from V_{DDAD} to V_{SSAD}.
- V_{SSAD} (and V_{REFL}, if connected) is connected to V_{SS} at a quiet point in the ground plane.
- Operate the MCU in wait or stop3 mode before initiating (hardware triggered conversions) or immediately after initiating (hardware or software triggered conversions) the ADC conversion.
  - For software triggered conversions, immediately follow the write to the ADCSC1 with a WAIT instruction or STOP instruction.
  - For stop3 mode operation, select ADACK as the clock source. Operation in stop3 reduces V_{DD} noise but increases effective conversion time due to stop recovery.
- There is no I/O switching, input or output, on the MCU during the conversion.

There are some situations where external system activity causes radiated or conducted noise emissions or excessive V_{DD} noise is coupled into the ADC. In these situations, or when the MCU cannot be placed in wait or stop3 or I/O activity cannot be halted, these recommended actions may reduce the effect of noise on the accuracy:

- Place a 0.01 µF capacitor (C_{AS}) on the selected input channel to V_{REFL} or V_{SSAD} (this will improve noise issues but will affect sample rate based on the external analog source resistance).

Table 11-7. IICS Field Descriptions

Field	Description
7 TCF	**Transfer Complete Flag.** This bit is set on the completion of a byte transfer. This bit is only valid during or immediately following a transfer to the IIC module or from the IIC module.The TCF bit is cleared by reading the IICD register in receive mode or writing to the IICD in transmit mode. 0 Transfer in progress 1 Transfer complete
6 IAAS	**Addressed as a Slave.** The IAAS bit is set when the calling address matches the programmed slave address or when the GCAEN bit is set and a general call is received. Writing the IICC register clears this bit. 0 Not addressed 1 Addressed as a slave
5 BUSY	**Bus Busy.** The BUSY bit indicates the status of the bus regardless of slave or master mode. The BUSY bit is set when a start signal is detected and cleared when a stop signal is detected. 0 Bus is idle 1 Bus is busy
4 ARBL	**Arbitration Lost.** This bit is set by hardware when the arbitration procedure is lost. The ARBL bit must be cleared by software by writing a 1 to it. 0 Standard bus operation 1 Loss of arbitration
2 SRW	**Slave Read/Write.** When addressed as a slave, the SRW bit indicates the value of the R/W command bit of the calling address sent to the master. 0 Slave receive, master writing to slave 1 Slave transmit, master reading from slave
1 IICIF	**IIC Interrupt Flag.** The IICIF bit is set when an interrupt is pending. This bit must be cleared by software, by writing a 1 to it in the interrupt routine. One of the following events can set the IICIF bit: • One byte transfer completes • Match of slave address to calling address • Arbitration lost 0 No interrupt pending 1 Interrupt pending
0 RXAK	**Receive Acknowledge**. When the RXAK bit is low, it indicates an acknowledge signal has been received after the completion of one byte of data transmission on the bus. If the RXAK bit is high it means that no acknowledge signal is detected. 0 Acknowledge received 1 No acknowledge received

11.3.5 IIC Data I/O Register (IICD)

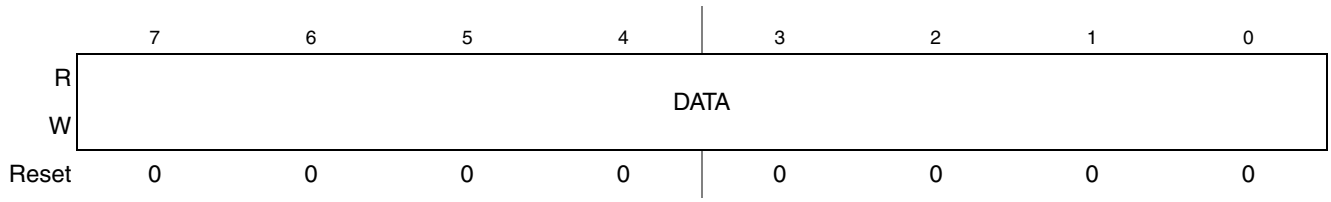


Figure 11-7. IIC Data I/O Register (IICD)

The SLIC clock is the same as the CPU bus clock. The module is designed to provide better than 1% bit rate accuracy at the lowest value of the SLIC clock frequency and the accuracy improves as the SLIC clock frequency is increased. For this reason, it is advantageous to choose the fastest SLIC clock which is still within the acceptable operating range of the SLIC.Because the SLIC may be used with MCUs with internal oscillators, the tolerance of the oscillator must be taken into account to ensure that SLIC clock frequency does not exceed the bounds of the SLIC clock operating range. This is especially important if the user wishes to use the oscillator untrimmed, where process variations might result in MCU frequency offsets of ±25%.

The acceptable range of SLIC clock frequencies is 2 to 20 MHz to guarantee LIN operations with greater than 1.5% accuracy across the 1–20 kbps range of LIN bit rates. The user must ensure that the fastest possible SLIC clock frequency never exceeds 20 MHz or that the slowest possible SLIC clock never falls below 2 MHz under worst case conditions. This would include, for example, oscillator frequency variations due to untrimmed oscillator tolerance, temperature variation, or supply voltage variation.

To initialize the SLIC module into LIN operating mode, the user must perform the following steps prior to needing to receive any LIN message traffic. These steps assume the MCU has been reset either by a power-on reset (POR) or any other MCU reset mechanism.

The steps for SLIC Initialization for LIN operation are:

1. Write SLCC1 to clear INITREQ.
2. When INITACK = 0, write SLCC1 & SLCC2 with desired values for:
   a) SLCWCM — Wait clock mode.
3. Write SLCC2 to set up prescalers for:
   a) RXFP — Digital receive filter clock prescaler.
4. Enable the SLIC module by writing SLCC2:
   a) SLCE = 1 to place SLIC module into run mode.
   b) BTM = 0 to disable byte transfer mode.
5. Write SLCC1 to enable SLIC interrupts (if desired).

12.6.6.2 Byte Transfer Mode Initialization

Bit rate synchronization is handled automatically in LIN mode, using the synchronization data contained in each LIN message to derive the desired bit rate. In byte transfer mode (BTM = 1); however, the user must set up the bit rate for communications using SLCBT.

More information on byte transfer mode is described in Section 12.6.16, “Byte Transfer Mode Operation,” including the performance parameters on recommended maximum speeds, bit time resolution, and oscillator tolerance requirements.

After the desired settings of bit time are determined, the SLIC Initialization for BTM operation is virtually identical to that of LIN operation.

The steps are:

1. Write SLCC1 to clear INITREQ.

Table 12-14. Digital Receive Filter Absolute Cutoff (Ideal Conditions)[1]

SLIC clock (MHz)	Max Bit Rate (kbps)	Min Pulse Width Allowed (μs)	Max Bit Rate (kbps)	Min Pulse Width Allowed (μs)	Max Bit Rate (kbps)	Min Pulse Width Allowed (μs)	Max Bit Rate (kbps)	Min Pulse Width Allowed (μs)
	RXFP = ÷8		RXFP = ÷7		RXFP = ÷6		RXFP = ÷5	
2	15,625	64.00	17,857	56.00	20,833	48.00	25,000	40.00
	RXFP = ÷4		RXFP = ÷3		RXFP = ÷2		RXFP = ÷1	
20	312,500	3.20	416,667	2.40	625,000	1.60	1,250,000	0.80
18	281,250	3.56	375,000	2.67	562,500	1.78	1,125,000	0.89
16	250,000	4.00	333,333	3.00	500,000	2.00	1,000,000	1.00
14	218,750	4.57	291,667	3.43	437,500	2.29	875,000	1.14
12	187,500	5.33	250,000	4.00	375,000	2.67	750,000	1.33
10	156,250	6.40	208,333	4.80	312,500	3.20	625,000	1.60
8	125,000	8.00	166,667	6.00	250,000	4.00	500,000	2.00
6	93,750	10.67	125,000	8.00	187,500	5.33	375,000	2.67
4	62,500	16.00	83,333	12.00	125,000	8.00	250,000	4.00
2	31,250	32.00	41,667	24.00	62,500	16.00	125,000	8.00

1 Bit rates over 120,000 bits per second are not recommended for LIN communications, as physical layer delay between the TX and RX pins can cause the stop bit of a byte to be mis-sampled as the last data bit. This could result in a byte framing error.

12.6.15 Bit Error Detection and Physical Layer Delay

The bit error detection circuitry of the SLIC module monitors the received bits to determine if they match the state of the corresponding transmitted bits. The sampling of the receive line takes place near the end of the bit being transmitted, so as long as the total physical layer delay does not exceed 75% of one bit time, bit error detection will work properly. For normal LIN bus speeds (<= 20 kbps), the physical layer delay in the system is typically significantly lower than 75% of a bit time and bit error detection should remain enabled by the user.

If the physical layer delay begins to exceed 75% of one bit time, the received bits begin to significantly lag behind the transmitted bits. In this case, it's possible for the bit error detection circuitry to falsely sample the delayed 'previous' bit on the receive pin rather than the current bit. It is the responsibility of the user to determine if the total physical layer delay is large enough to require disabling the bit error detection circuitry. This should only be required at speeds higher than allowed in normal LIN operations.

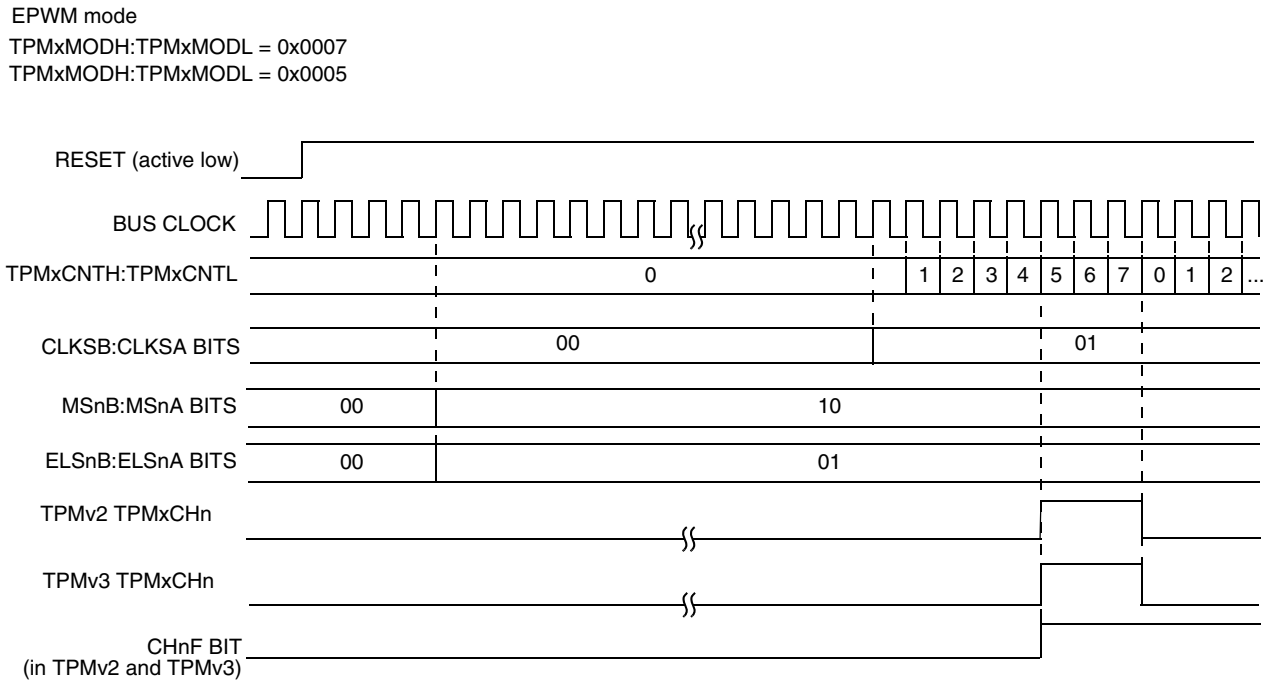


Figure 0-2. Generation of low-true EPWM signal by TPM v2 and v3 after the reset

The following procedure can be used in TPM v3 (when the channel pin is also a port pin) to emulate the high-true EPWM generated by TPM v2 after the reset.

...

configure the channel pin as output port pin and set the output pin;

configure the channel to generate the EPWM signal but keep ELSnB:ELSnA as 00;

configure the other registers (TPMxMODH, TPMxMODL, TPMxCnVH, TPMxCnVL, ...);

configure CLKSB:CLKSA bits (TPM v3 starts to generate the high-true EPWM signal, however TPM does not control the channel pin, so the EPWM signal is not available);

wait until the TOF is set (or use the TOF interrupt);

enable the channel output by configuring ELSnB:ELSnA bits (now EPWM signal is available);

...

1 Junction temperature is a function of die size, on-chip power dissipation, package thermal resistance, mounting site (board) temperature, ambient temperature, air flow, power dissipation of other components on the board, and board thermal resistance.
2 Junction to Ambient Natural Convection

The average chip-junction temperature (T_J) in °C can be obtained from:

$$T_J = T_A + (P_D \times \theta_{JA}) \quad \textit{Eqn. A-1}$$

where:

T_A = Ambient temperature, °C
θ_{JA} = Package thermal resistance, junction-to-ambient, °C/W
$P_D = P_{int} + P_{I/O}$
$P_{int} = I_{DD} \times V_{DD}$, Watts — chip internal power
$P_{I/O}$ = Power dissipation on input and output pins — user determined

For most applications, $P_{I/O} << P_{int}$ and can be neglected. An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad \textit{Eqn. A-2}$$

Solving Equation A-1 and Equation A-2 for K gives:

$$K = P_D \times (T_A + 273°C) + \theta_{JA} \times (P_D)^2 \quad \textit{Eqn. A-3}$$

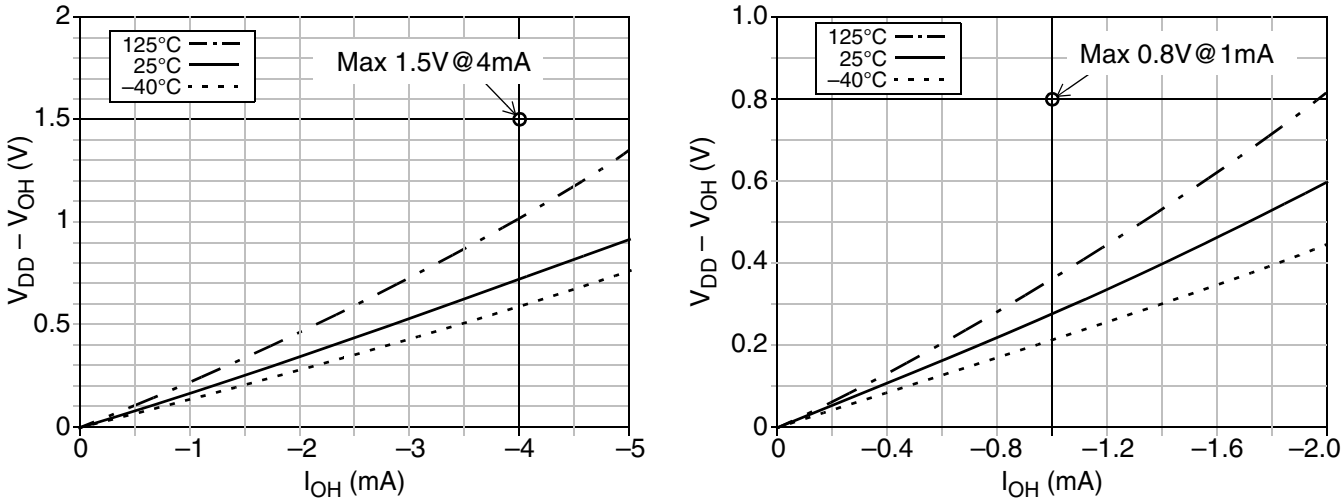
where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving Equation A-1 and Equation A-2 iteratively for any value of T_A.

A.5 ESD Protection and Latch-Up Immunity

Although damage from electrostatic discharge (ESD) is much less common on these devices than on early CMOS circuits, normal handling precautions should be used to avoid exposure to static discharge. Qualification tests are performed to ensure that these devices can withstand exposure to reasonable levels of static without suffering any permanent damage.

All ESD testing is in conformity with AEC-Q100 Stress Test Qualification for Automotive Grade Integrated Circuits. During the device qualification ESD stresses were performed for the human body model (HBM) and the charge device model (CDM).

A device is defined as a failure if after exposure to ESD pulses the device no longer meets the device specification. Complete DC parametric and functional testing is performed per the applicable device specification at room temperature followed by hot temperature, unless specified otherwise in the device specification.



a) V_{DD} = 5V, Low Drive

b) V_{DD} = 3V, Low Drive

Figure A-4. Typical $V_{DD} - V_{OH}$ vs I_{OH}, Low Drive Strength

A.7 Supply Current Characteristics

This section includes information about power supply current in various operating modes.

Table A-7. Supply Current Characteristics

Num	C	Parameter	Symbol	V_{DD} (V)	Typ[1]	Max[2]	Unit
1	C	Run supply current[3] measured at (CPU clock = 4 MHz, f_{Bus} = 2 MHz)	RI_{DD}	5	1.7	2.5	mA
	C			3	1.7	2.4	
2	P	Run supply current[3] measured at (CPU clock = 16 MHz, f_{Bus} = 8 MHz)	RI_{DD}	5	5.1	8.5	mA
	C			3	5.0	8.4	
3	C	Run supply current[4] measured at (CPU clock = 32 MHz, f_{Bus} = 16MHz)	RI_{DD}	5	7.8	15	mA
	C			3	7.7	14	
4		Stop3 mode supply current					
	C	–40°C (C, V, & M suffix)	$S3I_{DD}$	5	1.0	–	µA
	P	25°C (All parts)			1.0	–	
	P[5]	85°C (C suffix only)			6.8	40.0	
	P[5]	105°C (V suffix only)			15.6	50.0	
	P[5]	125°C (M suffix only)			42	75.0	
	C	–40°C (C,V, & M suffix)		3	0.9	–	µA
	P	25°C (All parts)			0.9	–	
	P[5]	85°C (C suffix only)			6.0	35.0	
	P[5]	105°C (V suffix only)			13.1	45.0	
	P[5]	125°C (M suffix only)			38	70.0	

Table A-10. Analog Comparator Electrical Specifications (continued)

Num	C	Rating	Symbol	Min	Typical	Max	Unit
4	D	Analog input offset voltage	V_{AIO}		20	40	mV
5	D	Analog Comparator hysteresis	V_H	3.0	6.0	20.0	mV
6	D	Analog input leakage current	I_{ALKG}	—	—	1.0	μA
7	D	Analog Comparator initialization delay	t_{AINIT}	—	—	1.0	μs

A.11 ADC Characteristics

Table A-11. ADC Operating Conditions

Num	Characteristic	Conditions	Symb	Min	Typ[1]	Max	Unit	Comment
1	Supply voltage	Absolute	V_{DDAD}	2.7	—	5.5	V	
2	Input Voltage		V_{ADIN}	V_{REFL}	—	V_{REFH}	V	
3	Input Capacitance		C_{ADIN}	—	4.5	5.5	pF	
4	Input Resistance		R_{ADIN}	—	3	5	kΩ	
5	Analog Source Resistance	10 bit mode f_{ADCK} > 4MHz f_{ADCK} < 4MHz	R_{AS}	 — —	 — —	 5 10	kΩ	External to MCU
6		8 bit mode (all valid f_{ADCK})		—	—	10		
7	ADC Conversion Clock Freq.	High Speed (ADLPC=0)	f_{ADCK}	0.4	—	8.0	MHz	
8		Low Power (ADLPC=1)		0.4	—	4.0		

[1] Typical values assume $V_{DDAD} = V_{DD} = 5.0V$, Temp = 25°C, f_{ADCK}=1.0MHz unless otherwise stated. Typical values are for reference only and are not tested in production.

B.2 Mechanical Drawings

The latest package outline drawings are available on the product summary pages on http://www.freescale.com. Table B-2 lists the document numbers per package type. Use these numbers in the web page's keyword search engine to find the latest package outline drawings.

Table B-2. Package Descriptions

Pin Count	Type	Abbreviation	Designator	Document No.
28	Thin shrink small outline package	TSSOP	TL	98ARS23923W
20	Thin shrink small outline package	TSSOP	TJ	98ASH70169A