

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	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	21
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68908gz16vfje

Table of Contents

Chapter 1 General Description

1.1	Introduction	19
1.2	Features	19
1.2.1	Standard Features	19
1.2.2	Features of the CPU08	21
1.3	MCU Block Diagram	21
1.4	Pin Assignments	21
1.5	Pin Functions	24
1.5.1	Power Supply Pins (V_{DD} and V_{SS})	24
1.5.2	Oscillator Pins (OSC1 and OSC2)	24
1.5.3	External Reset Pin (\overline{RST})	24
1.5.4	External Interrupt Pin (\overline{IRQ})	24
1.5.5	CGM Power Supply Pins (V_{DDA} and V_{SSA})	25
1.5.6	External Filter Capacitor Pin (V_{CGMXFC})	25
1.5.7	ADC Power Supply/Reference Pins (V_{DDAD}/V_{REFH} and V_{SSAD}/V_{REFL})	25
1.5.8	Port A Input/Output (I/O) Pins (PTA7/KBD7–PTA0/KBD0)	25
1.5.9	Port B I/O Pins (PTB7/AD7–PTB0/AD0)	25
1.5.10	Port C I/O Pins (PTC6–PTC0/CANTX)	25
1.5.11	Port D I/O Pins (PTD7/T2CH1–PTD0/ \overline{SS})	25
1.5.12	Port E I/O Pins (PTE5–PTE2, PTE1/RxD, and PTE0/TxD)	26
1.6	Unused Pin Termination	26

Chapter 2 Memory

2.1	Introduction	27
2.2	Unimplemented Memory Locations	27
2.3	Reserved Memory Locations	27
2.4	Input/Output (I/O) Section	27
2.5	Random-Access Memory (RAM)	38
2.6	FLASH Memory (FLASH)	38
2.6.1	Functional Description	38
2.6.2	FLASH Control Register	39
2.6.3	FLASH Page Erase Operation	40
2.6.4	FLASH Mass Erase Operation	41
2.6.5	FLASH Program/Read Operation	41
2.6.6	FLASH Block Protection	44
2.6.7	FLASH Block Protect Register	44
2.6.8	Wait Mode	45
2.6.9	Stop Mode	45

Table of Contents

16.6	Low-Power Modes	225
16.6.1	Wait Mode	225
16.6.2	Stop Mode	226
16.7	SIM Registers	227
16.7.1	Break Status Register	228
16.7.2	SIM Reset Status Register	228
16.7.3	Break Flag Control Register	229

Chapter 17 Serial Peripheral Interface (SPI) Module

17.1	Introduction	231
17.2	Features	231
17.3	Pin Name Conventions	231
17.4	Functional Description	233
17.4.1	Master Mode	233
17.4.2	Slave Mode	235
17.5	Transmission Formats	235
17.5.1	Clock Phase and Polarity Controls	235
17.5.2	Transmission Format When CPHA = 0	236
17.5.3	Transmission Format When CPHA = 1	237
17.5.4	Transmission Initiation Latency	237
17.6	Queuing Transmission Data	239
17.7	Error Conditions	240
17.7.1	Overflow Error	240
17.7.2	Mode Fault Error	241
17.8	Interrupts	243
17.9	Resetting the SPI	244
17.10	Low-Power Modes	244
17.10.1	Wait Mode	244
17.10.2	Stop Mode	244
17.11	SPI During Break Interrupts	245
17.12	I/O Signals	245
17.12.1	MISO (Master In/Slave Out)	245
17.12.2	MOSI (Master Out/Slave In)	246
17.12.3	SPSCK (Serial Clock)	246
17.12.4	\overline{SS} (Slave Select)	246
17.12.5	CGND (Clock Ground)	247
17.13	I/O Registers	247
17.13.1	SPI Control Register	247
17.13.2	SPI Status and Control Register	248
17.13.3	SPI Data Register	250

Chapter 18 Timebase Module (TBM)

18.1	Introduction	251
18.2	Features	251
18.3	Functional Description	251

18.4	Interrupts	251
18.5	TBM Interrupt Rate	252
18.6	Low-Power Modes	253
18.6.1	Wait Mode	253
18.6.2	Stop Mode	253
18.7	Timebase Control Register	254

Chapter 19

Timer Interface Module (TIM)

19.1	Introduction	255
19.2	Features	257
19.3	Pin Name Conventions	257
19.4	Functional Description	257
19.4.1	TIM Counter Prescaler	259
19.4.2	Input Capture	259
19.4.3	Output Compare	260
19.4.3.1	Unbuffered Output Compare	260
19.4.3.2	Buffered Output Compare	260
19.4.4	Pulse Width Modulation (PWM)	261
19.4.4.1	Unbuffered PWM Signal Generation	261
19.4.4.2	Buffered PWM Signal Generation	262
19.4.4.3	PWM Initialization	262
19.5	Interrupts	263
19.6	Low-Power Modes	263
19.6.1	Wait Mode	263
19.6.2	Stop Mode	264
19.7	TIM During Break Interrupts	264
19.8	I/O Signals	264
19.9	I/O Registers	264
19.9.1	TIM Status and Control Register	265
19.9.2	TIM Counter Registers	266
19.9.3	TIM Counter Modulo Registers	267
19.9.4	TIM Channel Status and Control Registers	267
19.9.5	TIM Channel Registers	270

Chapter 20

Development Support

20.1	Introduction	271
20.2	Break Module (BRK)	271
20.2.1	Functional Description	271
20.2.1.1	Flag Protection During Break Interrupts	273
20.2.1.2	TIM During Break Interrupts	273
20.2.1.3	COP During Break Interrupts	273
20.2.2	Break Module Registers	273
20.2.2.1	Break Status and Control Register	274
20.2.2.2	Break Address Registers	274
20.2.2.3	Break Status Register	275
20.2.2.4	Break Flag Control Register	275

Chapter 1

General Description

1.1 Introduction

The MC68HC908GZ16 is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

Table 1-1. Summary of Device Variations

Device	Memory Size
MC68HC908QZ16	16 Kbytes user FLASH
MC68HC908GZ8	8 Kbytes user FLASH

The information contained in this document pertains to both the MC68HC908GZ16 and the MC68HC908GZ8 with the exceptions shown Appendix A MC68HC908GZ8

1.2 Features

For convenience, features have been organized to reflect:

- Standard features
- Features of the CPU08

1.2.1 Standard Features

Features include:

- High-performance M68HC08 architecture optimized for C-compilers
- Fully upward-compatible object code with M6805, M146805, and M68HC05 Families
- 8-MHz internal bus frequency
- Clock generation module supporting 1-MHz to 8-MHz crystals
- MSCAN08 (implementing 2.0b protocol as defined in BOSCH specification dated September 1991)
- FLASH program memory security⁽¹⁾
- On-chip programming firmware for use with host personal computer which does not require high voltage for entry
- In-system programming (ISP)

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.

3.8.2.3 Left Justified Signed Data Mode

In left justified signed data mode, the ADRH register holds the eight MSBs of the 10-bit result. The only difference from left justified mode is that the AD9 is complemented. The ADRL register holds the two LSBs of the 10-bit result. All other bits read as 0. ADRH and ADRL are updated each time an ADC single channel conversion completes. Reading ADRH latches the contents of ADRL until ADRL is read. All subsequent results will be lost until the ADRH and ADRL reads are completed.

Address:	\$003D							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Write:								
Reset:	Unaffected by reset							
Address:	\$003E							
Read:	AD1	AD0	0	0	0	0	0	0
Write:								
Reset:	Unaffected by reset							
	<div></div> = Unimplemented							

Figure 3-7. ADC Data Register High (ADRH) and Low (ADRL)

3.8.2.4 Eight Bit Truncation Mode

In 8-bit truncation mode, the ADRL register holds the eight MSBs of the 10-bit result. The ADRH register is unused and reads as 0. The ADRL register is updated each time an ADC single channel conversion completes. In 8-bit mode, the ADRL register contains no interlocking with ADRH.

Address:	\$003D								ADRH
	Bit 7	6	5	4	3	2	1	Bit 0	
Read:	0	0	0	0	0	0	0	0	
Write:									
Reset:	Unaffected by reset								
Address:	\$003E								ADRL
Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	
Write:									
Reset:	Unaffected by reset								
	<div></div> = Unimplemented								

Figure 3-8. ADC Data Register High (ADRH) and Low (ADRL)

4.4 I/O Signals

The following paragraphs describe the CGM I/O signals.

4.4.1 Crystal Amplifier Input Pin (OSC1)

The OSC1 pin is an input to the crystal oscillator amplifier.

4.4.2 Crystal Amplifier Output Pin (OSC2)

The OSC2 pin is the output of the crystal oscillator inverting amplifier.

4.4.3 External Filter Capacitor Pin (CGMXFC)

The CGMXFC pin is required by the loop filter to filter out phase corrections. An external filter network is connected to this pin. (See Figure 4-2.)

NOTE

To prevent noise problems, the filter network should be placed as close to the CGMXFC pin as possible, with minimum routing distances and no routing of other signals across the network.

4.4.4 PLL Analog Power Pin (V_{DDA})

V_{DDA} is a power pin used by the analog portions of the PLL. Connect the V_{DDA} pin to the same voltage potential as the V_{DD} pin.

NOTE

Route V_{DDA} carefully for maximum noise immunity and place bypass capacitors as close as possible to the package.

4.4.5 PLL Analog Ground Pin (V_{SSA})

V_{SSA} is a ground pin used by the analog portions of the PLL. Connect the V_{SSA} pin to the same voltage potential as the V_{SS} pin.

NOTE

Route V_{SSA} carefully for maximum noise immunity and place bypass capacitors as close as possible to the package.

4.4.6 Oscillator Enable Signal (SIMOSCEN)

The SIMOSCEN signal comes from the system integration module (SIM) and enables the oscillator and PLL.

4.4.7 Oscillator Stop Mode Enable Bit (OSCSTOPENB)

OSCSTOPENB is a bit in the CONFIG register that enables the oscillator to continue operating during stop mode. If this bit is set, the Oscillator continues running during stop mode. If this bit is not set (default), the oscillator is controlled by the SIMOSCEN signal which will disable the oscillator during stop mode.

LVISTOP — LVI Enable in Stop Mode Bit

When the LVIPWRD bit is clear, setting the LVISTOP bit enables the LVI to operate during stop mode. Reset clears LVISTOP.

- 1 = LVI enabled during stop mode
- 0 = LVI disabled during stop mode

LVIRSTD — LVI Reset Disable Bit

LVIRSTD disables the reset signal from the LVI module. See Chapter 11 Low-Voltage Inhibit (LVI).

- 1 = LVI module resets disabled
- 0 = LVI module resets enabled

LVIPWRD — LVI Power Disable Bit

LVIPWRD disables the LVI module. See Chapter 11 Low-Voltage Inhibit (LVI).

- 1 = LVI module power disabled
- 0 = LVI module power enabled

LVI5OR3 — LVI 5-V or 3-V Operating Mode Bit

LVI5OR3 selects the voltage operating mode of the LVI module (see Chapter 11 Low-Voltage Inhibit (LVI)). The voltage mode selected for the LVI should match the operating V_{DD} (see Chapter 21

Electrical Specifications) for the LVI's voltage trip points for each of the modes.

- 1 = LVI operates in 5-V mode
- 0 = LVI operates in 3-V mode

NOTE

The LVI5OR3 bit is cleared by a power-on reset (POR) only. Other resets will leave this bit unaffected.

SSREC — Short Stop Recovery Bit

SSREC enables the CPU to exit stop mode with a delay of 32 CGMXCLK cycles instead of a 4096-CGMXCLK cycle delay.

- 1 = Stop mode recovery after 32 CGMXCLK cycles
- 0 = Stop mode recovery after 4096 CGMXCLK cycles

NOTE

Exiting stop mode by an LVI reset will result in the long stop recovery.

If the system clock source selected is the internal oscillator or the external crystal and the OSCENINSTOP configuration bit is not set, the oscillator will be disabled during stop mode. The short stop recovery does not provide enough time for oscillator stabilization and for this reason the SSREC bit should not be set.

The system stabilization time for power-on reset and long stop recovery (both 4096 CGMXCLK cycles) gives a delay longer than the LVI enable time for these startup scenarios. There is no period where the MCU is not protected from a low-power condition. However, when using the short stop recovery configuration option, the 32-CGMXCLK delay must be greater than the LVI's turn on time to avoid a period in startup where the LVI is not protecting the MCU.

STOP — STOP Instruction Enable Bit

STOP enables the STOP instruction.

- 1 = STOP instruction enabled
- 0 = STOP instruction treated as illegal opcode

10.9 Low-Voltage Inhibit Module (LVI)

10.9.1 Wait Mode

If enabled, the low-voltage inhibit (LVI) module remains active in wait mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of wait mode.

10.9.2 Stop Mode

If enabled, the LVI module remains active in stop mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of stop mode.

10.10 Enhanced Serial Communications Interface Module (ESCI)

10.10.1 Wait Mode

The enhanced serial communications interface (ESCI), or SCI module for short, module remains active in wait mode. Any enabled CPU interrupt request from the SCI module can bring the MCU out of wait mode.

If SCI module functions are not required during wait mode, reduce power consumption by disabling the module before executing the WAIT instruction.

10.10.2 Stop Mode

The SCI module is inactive in stop mode. The STOP instruction does not affect SCI register states. SCI module operation resumes after the MCU exits stop mode.

Because the internal clock is inactive during stop mode, entering stop mode during an SCI transmission or reception results in invalid data.

10.11 Serial Peripheral Interface Module (SPI)

10.11.1 Wait Mode

The serial peripheral interface (SPI) module remains active in wait mode. Any enabled CPU interrupt request from the SPI module can bring the MCU out of wait mode.

If SPI module functions are not required during wait mode, reduce power consumption by disabling the SPI module before executing the WAIT instruction.

10.11.2 Stop Mode

The SPI module is inactive in stop mode. The STOP instruction does not affect SPI register states. SPI operation resumes after an external interrupt. If stop mode is exited by reset, any transfer in progress is aborted, and the SPI is reset.

Low-Voltage Inhibit (LVI)

LVISTOP, LVIPWRD, LVI5OR3, and LVIRSTD are in the configuration register (CONFIG1). See Figure 5-2. Configuration Register 1 (CONFIG1) for details of the LVI's configuration bits. Once an LVI reset occurs, the MCU remains in reset until V_{DD} rises above a voltage, V_{TRIPR} , which causes the MCU to exit reset. See 16.3.2.5 Low-Voltage Inhibit (LVI) Reset for details of the interaction between the SIM and the LVI. The output of the comparator controls the state of the LVIOUT flag in the LVI status register (LVISR).

An LVI reset also drives the \overline{RST} pin low to provide low-voltage protection to external peripheral devices.

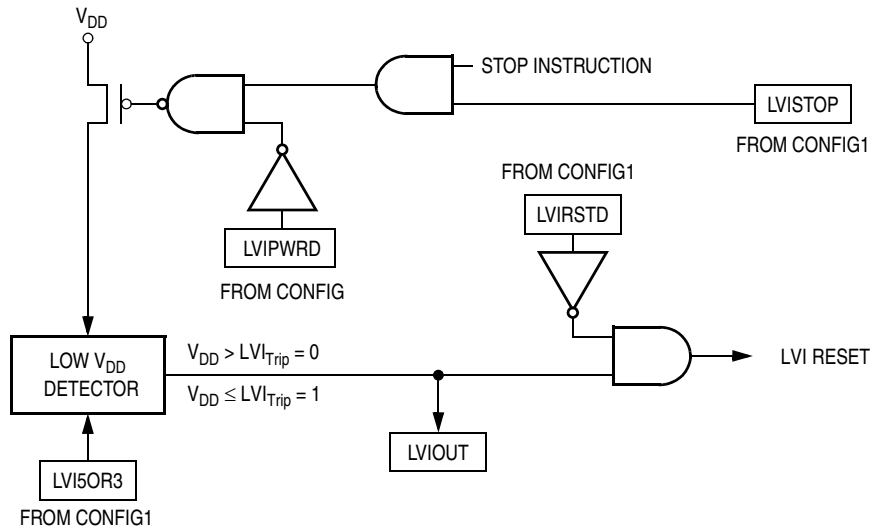


Figure 11-1. LVI Module Block Diagram

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$FE0C	LVI Status Register (LVISR) See page 119.	Read: LVIOUT	0	0	0	0	0	0	0
		Write:							
		Reset:	0	0	0	0	0	0	0
		= Unimplemented							

Figure 11-2. LVI I/O Register Summary

11.3.1 Polled LVI Operation

In applications that can operate at V_{DD} levels below the V_{TRIPF} level, software can monitor V_{DD} by polling the LVIOUT bit. In the configuration register, the LVIPWRD bit must be at logic 0 to enable the LVI module, and the LVIRSTD bit must be at logic 1 to disable LVI resets.

11.3.2 Forced Reset Operation

In applications that require V_{DD} to remain above the V_{TRIPF} level, enabling LVI resets allows the LVI module to reset the MCU when V_{DD} falls below the V_{TRIPF} level. In the configuration register, the LVIPWRD and LVIRSTD bits must be at logic 0 to enable the LVI module and to enable LVI resets.

MSCAN08 Controller (MSCAN08)

A double buffer scheme would de-couple the re-loading of the transmit buffers from the actual message being sent and as such reduces the reactivity requirements on the CPU. Problems may arise if the sending of a message would be finished just while the CPU re-loads the second buffer. In that case, no buffer would then be ready for transmission and the bus would be released.

At least three transmit buffers are required to meet the first of the above requirements under all circumstances. The MSCAN08 has three transmit buffers.

The second requirement calls for some sort of internal prioritization which the MSCAN08 implements with the “local priority” concept described in 12.4.2 Receive Structures.

12.4.2 Receive Structures

The received messages are stored in a 2-stage input first in first out (FIFO). The two message buffers are mapped using a “ping pong” arrangement into a single memory area (see Figure 12-3). While the background receive buffer (RxBG) is exclusively associated to the MSCAN08, the foreground receive buffer (RxFG) is addressable by the central processor unit (CPU08). This scheme simplifies the handler software, because only one address area is applicable for the receive process.

Both buffers have a size of 13 bytes to store the CAN control bits, the identifier (standard or extended), and the data content. For details, see 12.12 Programmer’s Model of Message Storage.

The receiver full flag (RXF) in the MSCAN08 receiver flag register (CRFLG), signals the status of the foreground receive buffer. When the buffer contains a correctly received message with matching identifier, this flag is set. See 12.13.5 MSCAN08 Receiver Flag Register (CRFLG)

On reception, each message is checked to see if it passes the filter (for details see 12.5 Identifier Acceptance Filter) and in parallel is written into RxBG. The MSCAN08 copies the content of RxBG into RxFG⁽¹⁾, sets the RXF flag, and generates a receive interrupt to the CPU⁽²⁾. The user’s receive handler has to read the received message from RxFG and to reset the RXF flag to acknowledge the interrupt and to release the foreground buffer. A new message which can follow immediately after the IFS field of the CAN frame, is received into RxBG. The overwriting of the background buffer is independent of the identifier filter function.

When the MSCAN08 module is transmitting, the MSCAN08 receives its own messages into the background receive buffer, RxBG. It does NOT overwrite RxFG, generate a receive interrupt or acknowledge its own messages on the CAN bus. The exception to this rule is in loop-back mode (see 12.13.2 MSCAN08 Module Control Register 1), where the MSCAN08 treats its own messages exactly like all other incoming messages. The MSCAN08 receives its own transmitted messages in the event that it loses arbitration. If arbitration is lost, the MSCAN08 must be prepared to become the receiver.

An overrun condition occurs when both the foreground and the background receive message buffers are filled with correctly received messages with accepted identifiers and another message is correctly received from the bus with an accepted identifier. The latter message will be discarded and an error interrupt with overrun indication will be generated if enabled. The MSCAN08 is still able to transmit messages with both receive message buffers filled, but all incoming messages are discarded.

1. Only if the RXF flag is not set.

2. The receive interrupt will occur only if not masked. A polling scheme can be applied on RXF also.

To protect the CAN bus system from fatal consequences resulting from violations of the above rule, the MSCAN08 drives the CAN_{TX} pin into recessive state.

In power-down mode, no registers can be accessed.

MSCAN08 bus activity can wake the MCU from CPU stop/MSCAN08 power-down mode. However, until the oscillator starts up and synchronization is achieved the MSCAN08 will not respond to incoming data.

12.8.4 CPU Wait Mode

The MSCAN08 module remains active during CPU wait mode. The MSCAN08 will stay synchronized to the CAN bus and generates transmit, receive, and error interrupts to the CPU, if enabled. Any such interrupt will bring the MCU out of wait mode.

12.8.5 Programmable Wakeup Function

The MSCAN08 can be programmed to apply a low-pass filter function to the CAN_{RX} input line while in internal sleep mode (see information on control bit WUPM in 12.13.2 MSCAN08 Module Control Register 1). This feature can be used to protect the MSCAN08 from wakeup due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference within noisy environments.

12.9 Timer Link

The MSCAN08 will generate a timer signal whenever a valid frame has been received. Because the CAN specification defines a frame to be valid if no errors occurred before the EOF field has been transmitted successfully, the timer signal will be generated right after the EOF. A pulse of one bit time is generated. As the MSCAN08 receiver engine also receives the frames being sent by itself, a timer signal also will be generated after a successful transmission.

The previously described timer signal can be routed into the on-chip timer interface module (TIM). This signal is connected to channel 0 of timer interface module 1 (TIM1) under the control of the timer link enable (TLNKEN) bit in CMCR0.

After the timer has been programmed to capture rising edge events, it can be used under software control to generate 16-bit time stamps which can be stored with the received message.

12.10 Clock System

Figure 12-8 shows the structure of the MSCAN08 clock generation circuitry and its interaction with the clock generation module (CGM). With this flexible clocking scheme the MSCAN08 is able to handle CAN bus rates ranging from 10 kbps up to 1 Mbps.

The clock source bit (CLKSRC) in the MSCAN08 module control register (CMCR1) (see 12.13.1 MSCAN08 Module Control Register 0) defines whether the MSCAN08 is connected to the output of the crystal oscillator or to the PLL output.

The clock source has to be chosen such that the tight oscillator tolerance requirements (up to 0.4%) of the CAN protocol are met.

Table 12-7. Baud Rate Prescaler

BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	Prescaler Value (P)
0	0	0	0	0	0	1
0	0	0	0	0	1	2
0	0	0	0	1	0	3
0	0	0	0	1	1	4
:	:	:	:	:	:	:
:	:	:	:	:	:	:
1	1	1	1	1	1	64

NOTE

The CBTR0 register can be written only if the SFTRES bit in the MSCAN08 module control register is set.

12.13.4 MSCAN08 Bus Timing Register 1

Address:	\$0503						
	Bit 7	6	5	4	3	2	Bit 0
Read:	SAMP	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11
Write:							
Reset:	0	0	0	0	0	0	0

Figure 12-19. Bus Timing Register 1 (CBTR1)

SAMP — Sampling

This bit determines the number of serial bus samples to be taken per bit time. If set, three samples per bit are taken, the regular one (sample point) and two preceding samples, using a majority rule. For higher bit rates, SAMP should be cleared, which means that only one sample will be taken per bit.

1 = Three samples per bit⁽¹⁾

0 = One sample per bit

TSEG22–TSEG10 — Time Segment

Time segments within the bit time fix the number of clock cycles per bit time and the location of the sample point. Time segment 1 (TSEG1) and time segment 2 (TSEG2) are programmable as shown in Table 12-8.

The bit time is determined by the oscillator frequency, the baud rate prescaler, and the number of time quanta (T_q) clock cycles per bit as shown in Table 12-4).

$$\text{Bit time} = \frac{\text{Pres value}}{f_{\text{MSCANCLK}}} \cdot \text{number of time quanta}$$

NOTE

The CBTR1 register can only be written if the SFTRES bit in the MSCAN08 module control register is set.

1. In this case PHASE_SEG1 must be at least 2 time quanta.

TWRNIF — Transmitter Warning Interrupt Flag

This flag is set when the MSCAN08 goes into warning status due to the transmit error counter (TEC) exceeding 96 and neither one of the error interrupt flags or the bus-off interrupt flag is set⁽¹⁾. If not masked, an error interrupt is pending while this flag is set.

- 1 = MSCAN08 has gone into transmitter warning status.
- 0 = No transmitter warning status has been reached.

RERRIF — Receiver Error Passive Interrupt Flag

This flag is set when the MSCAN08 goes into error passive status due to the receive error counter exceeding 127 and the bus-off interrupt flag is not set⁽²⁾. If not masked, an error interrupt is pending while this flag is set.

- 1 = MSCAN08 has gone into receiver error passive status.
- 0 = No receiver error passive status has been reached.

TERRIF — Transmitter Error Passive Interrupt Flag

This flag is set when the MSCAN08 goes into error passive status due to the transmit error counter exceeding 127 and the bus-off interrupt flag is not set⁽³⁾. If not masked, an error interrupt is pending while this flag is set.

- 1 = MSCAN08 went into transmit error passive status.
- 0 = No transmit error passive status has been reached.

BOFFIF — Bus-Off Interrupt Flag

This flag is set when the MSCAN08 goes into bus-off status, due to the transmit error counter exceeding 255. It cannot be cleared before the MSCAN08 has monitored 128 times 11 consecutive 'recessive' bits on the bus. If not masked, an error interrupt is pending while this flag is set.

- 1 = MSCAN08 has gone into bus-off status.
- 0 = No bus-off status has been reached.

OVRIF — Overrun Interrupt Flag

This flag is set when a data overrun condition occurs. If not masked, an error interrupt is pending while this flag is set.

- 1 = A data overrun has been detected since last clearing the flag.
- 0 = No data overrun has occurred.

RXF — Receive Buffer Full

The RXF flag is set by the MSCAN08 when a new message is available in the foreground receive buffer. This flag indicates whether the buffer is loaded with a correctly received message. After the CPU has read that message from the receive buffer the RXF flag must be cleared to release the buffer. A set RXF flag prohibits the exchange of the background receive buffer into the foreground buffer. If not masked, a receive interrupt is pending while this flag is set.

- 1 = The receive buffer is full. A new message is available.
- 0 = The receive buffer is released (not full).

NOTE

To ensure data integrity, no registers of the receive buffer shall be read while the RXF flag is cleared.

1. Condition to set the flag: $TWRNIF = (96 \rightarrow TEC) \& \overline{RERRIF} \& \overline{TERRIF} \& \overline{BOFFIF}$

2. Condition to set the flag: $RERRIF = (127 \rightarrow REC \rightarrow 255) \& \overline{BOFFIF}$

3. Condition to set the flag: $TERRIF = (128 \rightarrow TEC \rightarrow 255) \& \overline{BOFFIF}$

Table 15-6. ESCI LIN Control Bits

LINT	LINR	M	Functionality
0	0	X	Normal ESCI functionality
0	1	0	13-bit break detect enabled for LIN receiver
0	1	1	14-bit break detect enabled for LIN receiver
1	0	0	13-bit generation enabled for LIN transmitter
1	0	1	14-bit generation enabled for LIN transmitter
1	1	0	13-bit break detect/11-bit generation enabled for LIN
1	1	1	14-bit break detect/12-bit generation enabled for LIN

In LIN (version 1.2) systems, the master node transmits a break character which will appear as 11.05–14.95 dominant bits to the slave node. A data character of 0x00 sent from the master might appear as 7.65–10.35 dominant bit times. This is due to the oscillator tolerance requirement that the slave node must be within $\pm 15\%$ of the master node's oscillator. Since a slave node cannot know if it is running faster or slower than the master node (prior to synchronization), the LINR bit allows the slave node to differentiate between a 0x00 character of 10.35 bits and a break character of 11.05 bits. The break symbol length must be verified in software in any case, but the LINR bit serves as a filter, preventing false detections of break characters that are really 0x00 data characters.

SCP1 and SCP0 — ESCI Baud Rate Register Prescaler Bits

These read/write bits select the baud rate register prescaler divisor as shown in Table 15-7. Reset clears SCP1 and SCP0.

Table 15-7. ESCI Baud Rate Prescaling

SCP[1:0]	Baud Rate Register Prescaler Divisor (BPD)
0 0	1
0 1	3
1 0	4
1 1	13

SCR2–SCR0 — ESCI Baud Rate Select Bits

These read/write bits select the ESCI baud rate divisor as shown in Table 15-8. Reset clears SCR2–SCR0.

Table 15-8. ESCI Baud Rate Selection

SCR[2:1:0]	Baud Rate Divisor (BD)
0 0 0	1
0 0 1	2
0 1 0	4
0 1 1	8
1 0 0	16
1 0 1	32
1 1 0	64
1 1 1	128

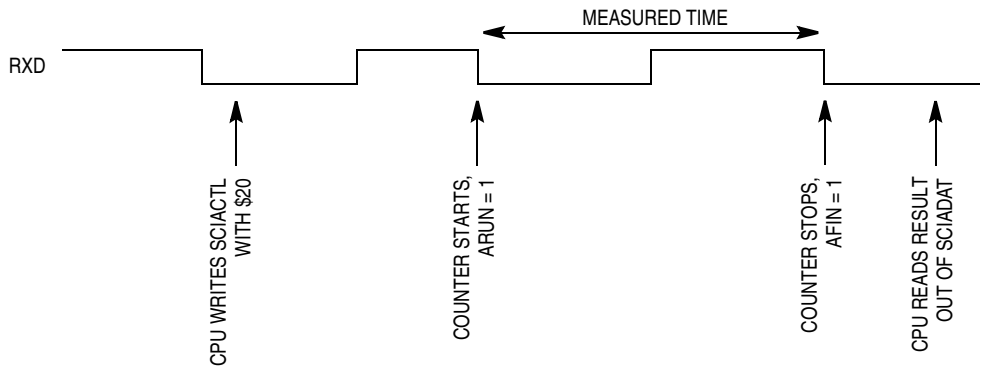


Figure 15-21. Bit Time Measurement with ACLK = 0

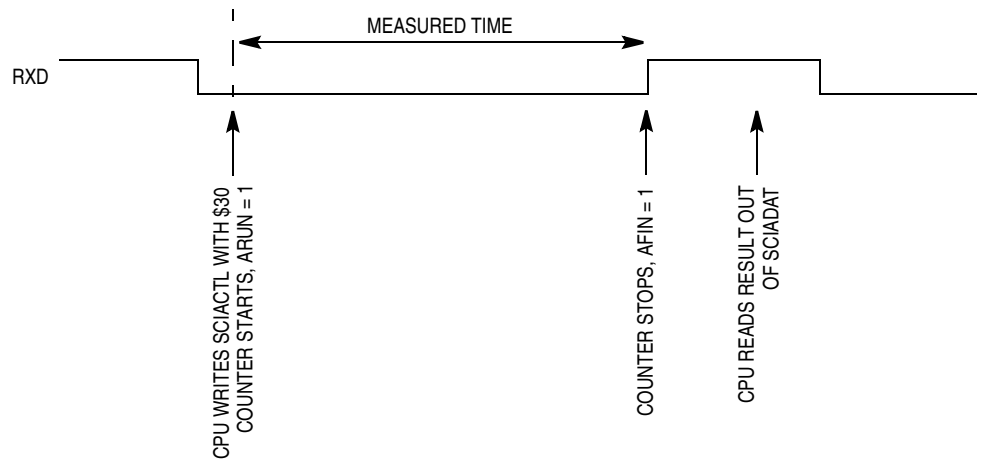


Figure 15-22. Bit Time Measurement with ACLK = 1, Scenario A

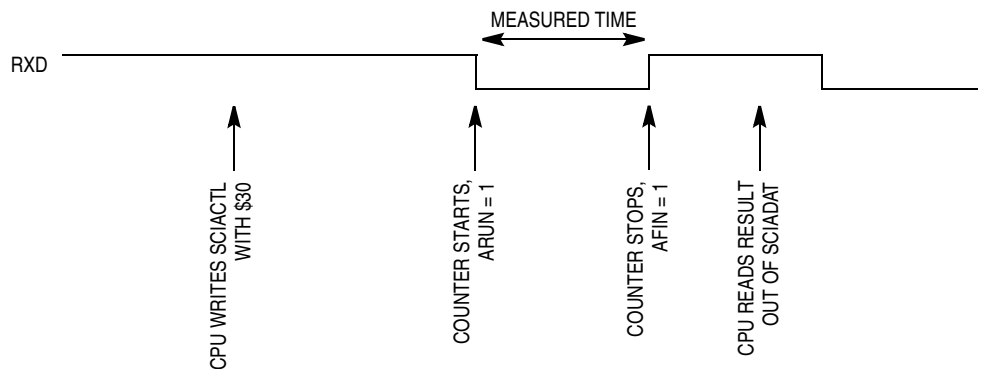


Figure 15-23. Bit Time Measurement with ACLK = 1, Scenario B

17.6 Queuing Transmission Data

The double-buffered transmit data register allows a data byte to be queued and transmitted. For an SPI configured as a master, a queued data byte is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag (SPTE) indicates when the transmit data buffer is ready to accept new data. Write to the transmit data register only when the SPTE bit is high. Figure 17-9 shows the timing associated with doing back-to-back transmissions with the SPI (SPSCK has CPHA:CPOL = 1:0).

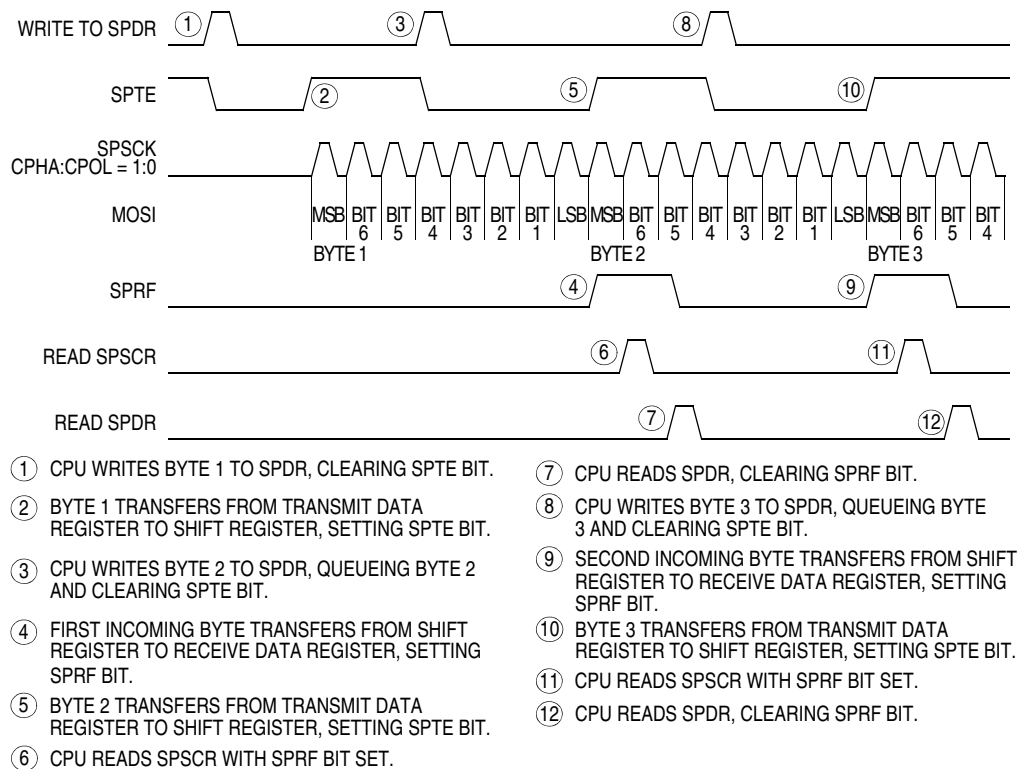


Figure 17-9. SPRF/SPTE CPU Interrupt Timing

The transmit data buffer allows back-to-back transmissions without the slave precisely timing its writes between transmissions as in a system with a single data buffer. Also, if no new data is written to the data buffer, the last value contained in the shift register is the next data word to be transmitted.

For an idle master or idle slave that has no data loaded into its transmit buffer, the SPTE is set again no more than two bus cycles after the transmit buffer empties into the shift register. This allows the user to queue up a 16-bit value to send. For an already active slave, the load of the shift register cannot occur until the transmission is completed. This implies that a back-to-back write to the transmit data register is not possible. The SPTE indicates when the next write can occur.

PS[2:0] — Prescaler Select Bits

These read/write bits select one of the seven prescaler outputs as the input to the TIM counter as Table 19-2 shows. Reset clears the PS[2:0] bits.

Table 19-2. Prescaler Selection

PS2	PS1	PS0	TIM Clock Source
0	0	0	Internal bus clock ÷ 1
0	0	1	Internal bus clock ÷ 2
0	1	0	Internal bus clock ÷ 4
0	1	1	Internal bus clock ÷ 8
1	0	0	Internal bus clock ÷ 16
1	0	1	Internal bus clock ÷ 32
1	1	0	Internal bus clock ÷ 64
1	1	1	Not available

19.9.2 TIM Counter Registers

The two read-only TIM counter registers contain the high and low bytes of the value in the TIM counter. Reading the high byte (TCNTH) latches the contents of the low byte (TCNTL) into a buffer. Subsequent reads of TCNTH do not affect the latched TCNTL value until TCNTL is read. Reset clears the TIM counter registers. Setting the TIM reset bit (TRST) also clears the TIM counter registers.

NOTE

If you read TCNTH during a break interrupt, be sure to unlatch TCNTL by reading TCNTL before exiting the break interrupt. Otherwise, TCNTL retains the value latched during the break.

Address: T1CNTH, \$0021 and T2CNTH, \$002C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	14	13	12	11	10	9	Bit 8
Write:								
Reset:	0	0	0	0	0	0	0	0
	= Unimplemented							

Figure 19-6. TIM Counter Registers High (TCNTH)

Address: T1CNTL, \$0022 and T2CNTL, \$002D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	0	0	0	0	0	0	0	0
	= Unimplemented							

Figure 19-7. TIM Counter Registers Low (TCNTL)

20.2.2.1 Break Status and Control Register

The break status and control register (BRKSCR) contains break module enable and status bits.

Address: \$FE0B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BRKE	BRKA	0	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figure 20-3. Break Status and Control Register (BRKSCR)

BRKE — Break Enable Bit

This read/write bit enables breaks on break address register matches. Clear BRKE by writing a logic 0 to bit 7. Reset clears the BRKE bit.

- 1 = Breaks enabled on 16-bit address match
- 0 = Breaks disabled

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a logic 1 to BRKA generates a break interrupt. Clear BRKA by writing a logic 0 to it before exiting the break routine. Reset clears the BRKA bit.

- 1 = Break address match
- 0 = No break address match

20.2.2.2 Break Address Registers

The break address registers (BRKH and BRKL) contain the high and low bytes of the desired breakpoint address. Reset clears the break address registers.

Address: \$FE09

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 20-4. Break Address Register High (BRKH)

Address: \$FE0A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 20-5. Break Address Register Low (BRKL)

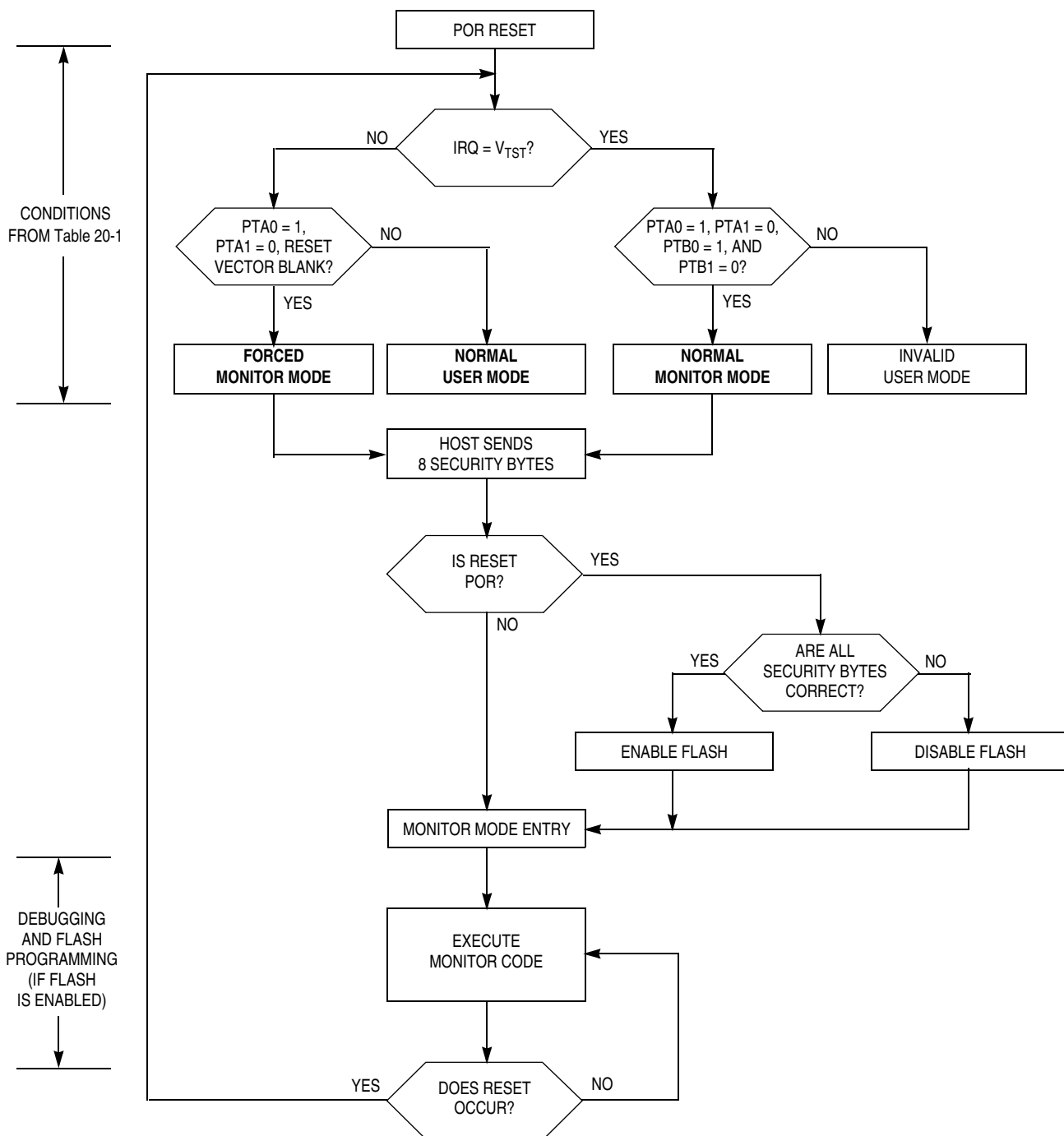


Figure 20-8. Simplified Monitor Mode Entry Flowchart

21.6 3.3-Vdc Electrical Characteristics

Characteristic ⁽¹⁾	Symbol	Min	Typ ⁽²⁾	Max	Unit
Output high voltage ($I_{Load} = -0.6$ mA) all I/O pins	V_{OH}	$V_{DD} - 0.3$	—	—	V
($I_{Load} = -4.0$ mA) all I/O pins	V_{OH}	$V_{DD} - 1.0$	—	—	V
($I_{Load} = -10.0$ mA) pins PTC0–PTC4 only	V_{OH}	$V_{DD} - 1.0$	—	—	V
Maximum combined I_{OH} for port PTA7–PTA3, port PTC0–PTC1, port E, port PTD0–PTD3	I_{OH1}	—	—	30	mA
Maximum combined I_{OH} for port PTA2–PTA0, port B, port PTC2–PTC6, port PTD4–PTD7	I_{OH2}	—	—	30	mA
Maximum total I_{OH} for all port pins	I_{OHT}	—	—	60	mA
Output low voltage ($I_{Load} = 1.6$ mA) all I/O pins	V_{OL}	—	—	0.3	V
($I_{Load} = 10$ mA) all I/O pins	V_{OL}	—	—	1.0	V
($I_{Load} = 20$ mA) pins PTC0–PTC4 only	V_{OL}	—	—	0.8	V
Maximum combined I_{OH} for port PTA7–PTA3, port PTC0–PTC1, port E, port PTD0–PTD3	I_{OL1}	—	—	30	mA
Maximum combined I_{OH} for port PTA2–PTA0, port B, port PTC2–PTC6, port PTD4–PTD7	I_{OL2}	—	—	30	mA
Maximum total I_{OL} for all port pins	I_{OLT}	—	—	60	mA
Input high voltage All ports, \overline{IRQ} , \overline{RST} , OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input low voltage All ports, \overline{IRQ} , \overline{RST} , OSC1	V_{IL}	V_{SS}	—	$0.3 \times V_{DD}$	V
V_{DD} supply current Run ⁽³⁾	I_{DD}	—	8	12	mA
Wait ⁽⁴⁾		—	3	6	mA
Stop ⁽⁵⁾		—	0.5	6	μ A
Stop with TBM enabled ⁽⁶⁾		—	500	700	μ A
Stop with LVI and TBM enabled ⁽⁶⁾		—	700	900	μ A
Stop with LVI		—	200	300	μ A
DC injection current ^{(7) (8) (9) (10)} Single pin limit $V_{in} > V_{DD}$	I_{IC}	0	—	2	mA
$V_{in} < V_{SS}$		0	—	–0.2	
Total MCU limit, includes sum of all stressed pins $V_{in} > V_{DD}$		0	—	25	
$V_{in} < V_{SS}$		0	—	–5	
I/O ports Hi-Z leakage current ⁽¹¹⁾	I_{IL}	0	—	± 10	μ A
Input current	I_{In}	0	—	± 1	μ A
Pullup resistors (as input only) Ports PTA7/KBD7–PTA0/KBD0, PTC6–PTC0, PTD7/T2CH1–PTD0/ \overline{SS}	R_{PU}	20	45	65	k Ω
Capacitance Ports (as input or output)	C_{Out} C_{In}	— —	— —	12 8	pF

Continued on next page