

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	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	37
Program Memory Size	8KB (8K 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 ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908gz8cfa

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

The address ranges for the user memory and vectors are:

- \$C000–\$FDFF; user memory
- \$FE08; FLASH control register
- \$FF7E; FLASH block protect register
- \$FFD4–\$FFFF; these locations are reserved for user-defined interrupt and reset vectors

Programming tools are available from Freescale Semiconductor. Contact your local representative for more information.

NOTE

A security feature prevents viewing of the FLASH contents.⁽¹⁾

2.6.2 FLASH Control Register

The FLASH control register (FLCR) controls FLASH program and erase operations.

Address:	\$FE08							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
Write:								
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figure 2-3. FLASH Control Register (FLCR)

HVEN — High-Voltage Enable Bit

This read/write bit enables the charge pump to drive high voltages for program and erase operations in the array. HVEN can only be set if either PGM = 1 or ERASE = 1 and the proper sequence for program or erase is followed.

- 1 = High voltage enabled to array and charge pump on
- 0 = High voltage disabled to array and charge pump off

MASS — Mass Erase Control Bit

Setting this read/write bit configures the 16-Kbyte FLASH array for mass erase operation.

- 1 = MASS erase operation selected
- 0 = PAGE erase operation selected

ERASE — Erase Control Bit

This read/write bit configures the memory for erase operation. ERASE is interlocked with the PGM bit such that both bits cannot be equal to 1 or set to 1 at the same time.

- 1 = Erase operation selected
- 0 = Erase operation unselected

PGM — Program Control Bit

This read/write bit configures the memory for program operation. PGM is interlocked with the ERASE bit such that both bits cannot be equal to 1 or set to 1 at the same time.

- 1 = Program operation selected
- 0 = Program operation unselected

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

Memory

10. Clear the PGM bit.⁽¹⁾
11. Wait for a time, t_{NVH} (minimum 5 μ s).
12. Clear the HVEN bit.
13. After time, t_{RCV} (typical 1 μ s), the memory can be accessed in read mode again.

This program sequence is repeated throughout the memory until all data is programmed.

NOTE

Programming and erasing of FLASH locations can not be performed by code being executed from the same FLASH array.

NOTE

While these operations must be performed in the order shown, other unrelated operations may occur between the steps. Care must be taken within the FLASH array memory space such as the COP control register (COPCTL) at \$FFFF.

NOTE

It is highly recommended that interrupts be disabled during program/ erase operations.

NOTE

Do not exceed t_{PROG} maximum or t_{HV} maximum. t_{HV} is defined as the cumulative high voltage programming time to the same row before next erase. t_{HV} must satisfy this condition:

$$t_{NVS} + t_{NVH} + t_{PGS} + (t_{PROG} \times 32) \leq t_{HV} \text{ maximum}$$

Refer to 21.15 Memory Characteristics.

NOTE

The time between programming the FLASH address change (step 7 to step 7), or the time between the last FLASH programmed to clearing the PGM bit (step 7 to step 10) must not exceed the maximum programming time, t_{PROG} maximum.

CAUTION

Be cautious when programming the FLASH array to ensure that non-FLASH locations are not used as the address that is written to when selecting either the desired row address range in step 3 of the algorithm or the byte to be programmed in step 7 of the algorithm. This applies particularly to \$FFD4–\$FFDF.

Chapter 5

Configuration Register (CONFIG)

5.1 Introduction

This section describes the configuration registers, CONFIG1 and CONFIG2. The configuration registers enable or disable these options:

- Stop mode recovery time (32 CGMXCLK cycles or 4096 CGMXCLK cycles)
- COP timeout period ($2^{18} - 2^4$ or $2^{13} - 2^4$ COPCLK cycles)
- STOP instruction
- Computer operating properly module (COP)
- Low-voltage inhibit (LVI) module control and voltage trip point selection
- Enable/disable the oscillator (OSC) during stop mode
- Enable/disable an extra divide by 128 prescaler in timebase module
- Enable for MSCAN

5.2 Functional Description

The configuration registers are used in the initialization of various options. The configuration registers can be written once after each reset. All of the configuration register bits are cleared during reset. Since the various options affect the operation of the microcontroller unit (MCU), it is recommended that these registers be written immediately after reset. The configuration registers are located at \$001E and \$001F and may be read at anytime.

NOTE

On a FLASH device, the options except LVI5OR3 are one-time writable by the user after each reset. The LVI5OR3 bit is one-time writable by the user only after each POR (power-on reset). The CONFIG registers are not in the FLASH memory but are special registers containing one-time writable latches after each reset. Upon a reset, the CONFIG registers default to predetermined settings as shown in Figure 5-1 and Figure 5-2.

Address: \$001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	MSCANEN	TMCLKSEL	OSCENINSTOP	ESCIBDSRC
Write:								
Reset:	0	0	0	0	See note	0	0	1

Note: MSCANEN is only reset via POR (power-on reset).

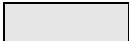
 = Unimplemented

Figure 5-1. Configuration Register 2 (CONFIG2)

Z — Zero Flag

The CPU sets the zero flag when an arithmetic operation, logic operation, or data manipulation produces a result of \$00.

1 = Zero result

0 = Non-zero result

C — Carry/Borrow Flag

The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some instructions — such as bit test and branch, shift, and rotate — also clear or set the carry/borrow flag.

1 = Carry out of bit 7

0 = No carry out of bit 7

7.4 Arithmetic/Logic Unit (ALU)

The ALU performs the arithmetic and logic operations defined by the instruction set.

Refer to the *CPU08 Reference Manual* (document order number CPU08RM/AD) for a description of the instructions and addressing modes and more detail about the architecture of the CPU.

7.5 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power-consumption standby modes.

7.5.1 Wait Mode

The WAIT instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling interrupts. After exit from wait mode by interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

7.5.2 Stop Mode

The STOP instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling external interrupts. After exit from stop mode by external interrupt, the I bit remains clear. After exit by reset, the I bit is set.
- Disables the CPU clock

After exiting stop mode, the CPU clock begins running after the oscillator stabilization delay.

7.6 CPU During Break Interrupts

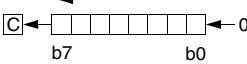
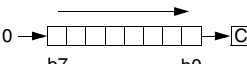
If a break module is present on the MCU, the CPU starts a break interrupt by:

- Loading the instruction register with the SWI instruction
- Loading the program counter with \$FFFC:\$FFFD or with \$FEFC:\$FEFD in monitor mode

The break interrupt begins after completion of the CPU instruction in progress. If the break address register match occurs on the last cycle of a CPU instruction, the break interrupt begins immediately.

A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the MCU to normal operation if the break interrupt has been deasserted.

Table 7-1. Instruction Set Summary (Sheet 4 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z				
JMP <i>opr</i> JMP <i>opr</i> JMP <i>opr</i> ,X JMP <i>opr</i> ,X JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff	2 3 4 3 2
JSR <i>opr</i> JSR <i>opr</i> JSR <i>opr</i> ,X JSR <i>opr</i> ,X JSR ,X	Jump to Subroutine	PC ← (PC) + <i>n</i> (<i>n</i> = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Unconditional Address	–	–	–	–	–	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	4 5 6 5 4
LDA # <i>opr</i> LDA <i>opr</i> LDA <i>opr</i> LDA <i>opr</i> ,X LDA <i>opr</i> ,X LDA ,X LDA <i>opr</i> ,SP LDA <i>opr</i> ,SP	Load A from M	A ← (M)	0	–	–	↑	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A6 B6 C6 D6 E6 F6 9EE6 9ED6	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LDHX # <i>opr</i> LDHX <i>opr</i>	Load H:X from M	H:X ← (M:M + 1)	0	–	–	↑	↑	IMM DIR	45 55	ii jj dd	3 4
LDX # <i>opr</i> LDX <i>opr</i> LDX <i>opr</i> LDX <i>opr</i> ,X LDX <i>opr</i> ,X LDX ,X LDX <i>opr</i> ,SP LDX <i>opr</i> ,SP	Load X from M	X ← (M)	0	–	–	↑	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	AE BE CE DE EE FE 9EEE 9EDE	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
LSL <i>opr</i> LSLA LSLX LSL <i>opr</i> ,X LSL ,X LSL <i>opr</i> ,SP	Logical Shift Left (Same as ASL)		↑	–	–	↑	↑	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
LSR <i>opr</i> LSRA LSRX LSR <i>opr</i> ,X LSR ,X LSR <i>opr</i> ,SP	Logical Shift Right		↑	–	–	0	↑	DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff ff	4 1 1 4 3 5
MOV <i>opr</i> , <i>opr</i> MOV <i>opr</i> ,X+ MOV # <i>opr</i> , <i>opr</i> MOV X+, <i>opr</i>	Move	(M) _{Destination} ← (M) _{Source} H:X ← (H:X) + 1 (IX+D, DIX+)	0	–	–	↑	↑	DD DIX+ IMD IX+D	4E 5E 6E 7E	dd dd dd ii dd dd	5 4 4 4
MUL	Unsigned multiply	X:A ← (X) × (A)	–	0	–	–	–	INH	42		5
NEG <i>opr</i> NEGA NEGX NEG <i>opr</i> ,X NEG ,X NEG <i>opr</i> ,SP	Negate (Two's Complement)	M ← –(M) = \$00 – (M) A ← –(A) = \$00 – (A) X ← –(X) = \$00 – (X) M ← –(M) = \$00 – (M) M ← –(M) = \$00 – (M)	↑	–	–	↑	↑	DIR INH INH IX1 IX SP1	30 40 50 60 70 9E60	dd ff ff ff	4 1 1 4 3 5
NOP	No Operation	None	–	–	–	–	–	INH	9D		1
NSA	Nibble Swap A	A ← (A[3:0]:A[7:4])	–	–	–	–	–	INH	62		3
ORA # <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ,X ORA <i>opr</i> ,X ORA ,X ORA <i>opr</i> ,SP ORA <i>opr</i> ,SP	Inclusive OR A and M	A ← (A) (M)	0	–	–	↑	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	AA BA CA DA EA FA 9EEA 9EDA	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
PSHA	Push A onto Stack	Push (A); SP ← (SP) – 1	–	–	–	–	–	INH	87		2
PSHH	Push H onto Stack	Push (H); SP ← (SP) – 1	–	–	–	–	–	INH	8B		2
PSHX	Push X onto Stack	Push (X); SP ← (SP) – 1	–	–	–	–	–	INH	89		2

MSCAN08 Controller (MSCAN08)

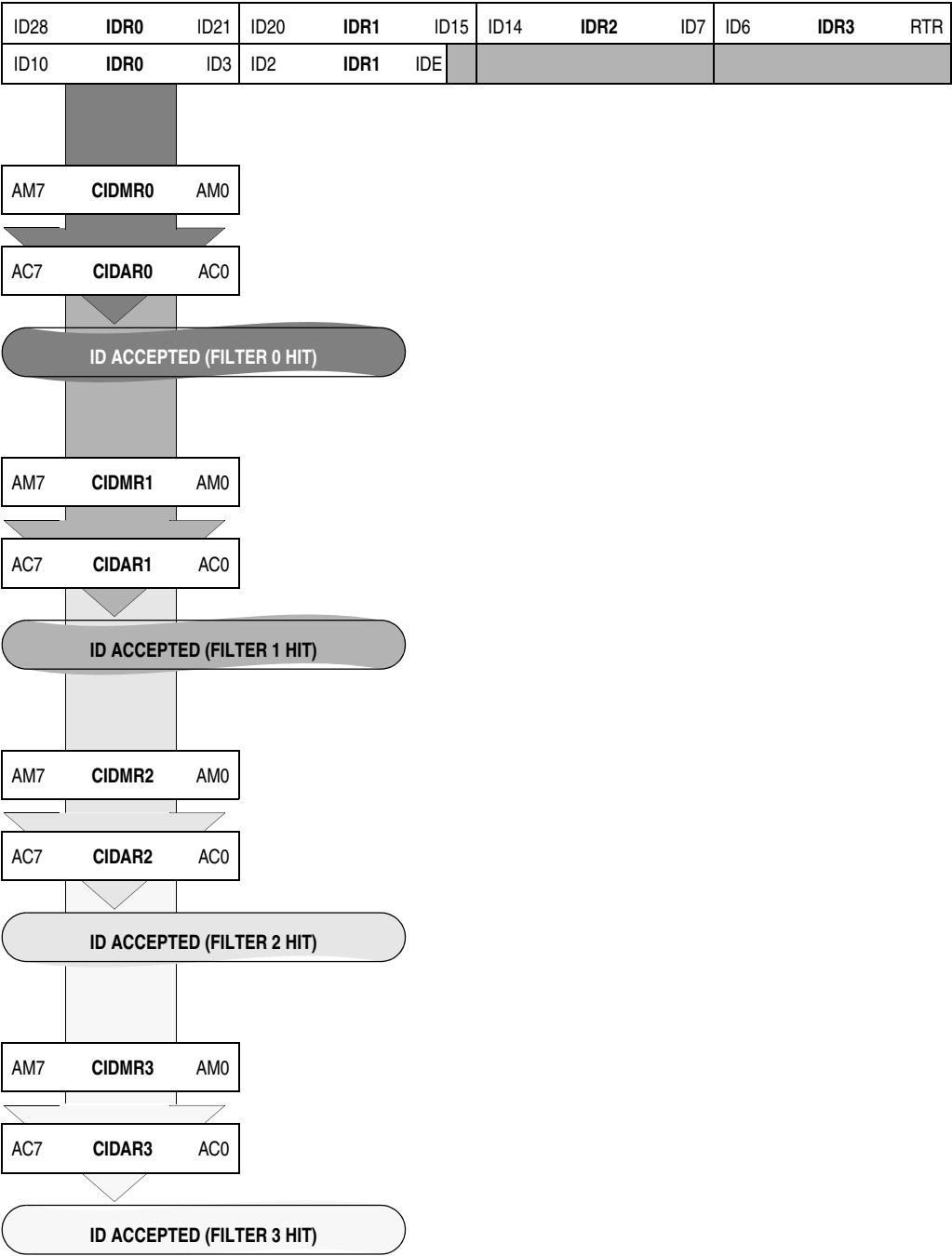


Figure 12-6. Quadruple 8-Bit Maskable Acceptance Filters

Table 13-1. Port Control Register Bits Summary

Port	Bit	DDR	Module Control		Pin
A	0	DDRA0	KBD	KBIE0	PTA0/KBD0
	1	DDRA1		KBIE1	PTA1/KBD1
	2	DDRA2		KBIE2	PTA2/KBD2
	3	DDRA3		KBIE3	PTA3/KBD3
	4	DDRA4		KBIE4	PTA4/KBD4
	5	DDRA5		KBIE5	PTA5/KBD5
	6	DDRA6		KBIE6	PTA6/KBD6
	7	DDRA7		KBIE7	PTA7/KBD7
B	0	DDRB0	ADC	ADCH4–ADCH0	PTB0/AD0
	1	DDRB1			PTB1/AD1
	2	DDRB2			PTB2/AD2
	3	DDRB3			PTB3/AD3
	4	DDRB4			PTB4/AD4
	5	DDRB5			PTB5/AD5
	6	DDRB6			PTB6/AD6
	7	DDRB7			PTB7/AD7
C	0	DDRC0	MSCAN08	CANEN	PTC0
	1	DDRC1			PTC1
	2	DDRC2			PTC2
	3	DDRC3			PTC3
	4	DDRC4			PTC4
	5	DDRC5			PTC5
	6	DDRC6			PTC6
D	0	DDRD0	SPI	SPE	PTD0/ \overline{SS}
	1	DDRD1			PTD1/MISO
	2	DDRD2			PTD2/MOSI
	3	DDRD3			PTD3/SPSCK
	4	DDRD4	TIM1	ELS0B:ELS0A	PTD4/T1CH0
	5	DDRD5		ELS1B:ELS1A	PTD5/T1CH1
	6	DDRD6	TIM2	ELS0B:ELS0A	PTD6/T2CH0
	7	DDRD7		ELS1B:ELS1A	PTD7/T2CH1
E	0	DDRE0	SCI	ENSCI	PTE0/TxD
	1	DDRE1			PTE1/RxD
	2	DDRE2			PTE2
	3	DDRE3			PTE3
	4	DDRE4			PTE4
	5	DDRE5			PTE5

PTAPUE7–PTAPUE0 — Port A Input Pullup Enable Bits

These writable bits are software programmable to enable pullup devices on an input port bit.

- 1 = Corresponding port A pin configured to have internal pullup
- 0 = Corresponding port A pin has internal pullup disconnected

13.4 Port B

Port B is an 8-bit special-function port that shares all eight of its pins with the analog-to-digital converter (ADC) module.

13.4.1 Port B Data Register

The port B data register (PTB) contains a data latch for each of the eight port pins.

Address:	\$0001							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Write:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
Reset:	Unaffected by reset							
Alternative Function:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Figure 13-6. Port B Data Register (PTB)

PTB7–PTB0 — Port B Data Bits

These read/write bits are software-programmable. Data direction of each port B pin is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.

AD7–AD0 — Analog-to-Digital Input Bits

AD7–AD0 are pins used for the input channels to the analog-to-digital converter module. The channel select bits in the ADC status and control register define which port B pin will be used as an ADC input and overrides any control from the port I/O logic by forcing that pin as the input to the analog circuitry. See Chapter 3 Analog-to-Digital Converter (ADC) for more information.

NOTE

Care must be taken when reading port B while applying analog voltages to AD7–AD0 pins. If the appropriate ADC channel is not enabled, excessive current drain may occur if analog voltages are applied to the PTBx/ADx pin, while PTB is read as a digital input. Those ports not selected as analog input channels are considered digital I/O ports.

13.4.2 Data Direction Register B

Data direction register B (DDRB) determines whether each port B pin is an input or an output. Writing a logic 1 to a DDRB bit enables the output buffer for the corresponding port B pin; a logic 0 disables the output buffer.

Resets and Interrupts

14.3.3.1 Interrupt Status Register 1

Address: \$FE04

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 14-6. Interrupt Status Register 1 (INT1)

IF6–IF1 — Interrupt Flags 6–1

These flags indicate the presence of interrupt requests from the sources shown in Table 14-2.

1 = Interrupt request present

0 = No interrupt request present

Bit 1 and Bit 0 — Always read 0

14.3.3.2 Interrupt Status Register 2

Address: \$FE05

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 14-7. Interrupt Status Register 2 (INT2)

IF14–IF7 — Interrupt Flags 14–7

These flags indicate the presence of interrupt requests from the sources shown in Table 14-2.

1 = Interrupt request present

0 = No interrupt request present

14.3.3.3 Interrupt Status Register 3

Address: \$FE06

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	IF20	IF19	IF18	IF17	IF16	IF15
Write:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 14-8. Interrupt Status Register 3 (INT3)

IF20–IF15 — Interrupt Flags 20–15

This flag indicates the presence of an interrupt request from the source shown in Table 14-2.

1 = Interrupt request present

0 = No interrupt request present

Bits 7–6 — Always read 0

15.4.1 Data Format

The SCI uses the standard non-return-to-zero mark/space data format illustrated in Figure 15-4.

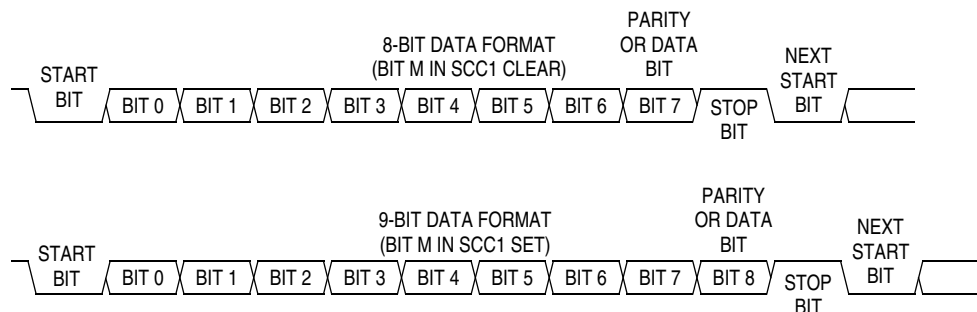


Figure 15-4. SCI Data Formats

15.4.2 Transmitter

Figure 15-5 shows the structure of the SCI transmitter and the registers are summarized in Figure 15-3. The baud rate clock source for the ESCI can be selected via the configuration bit, ESCIBDSRC.

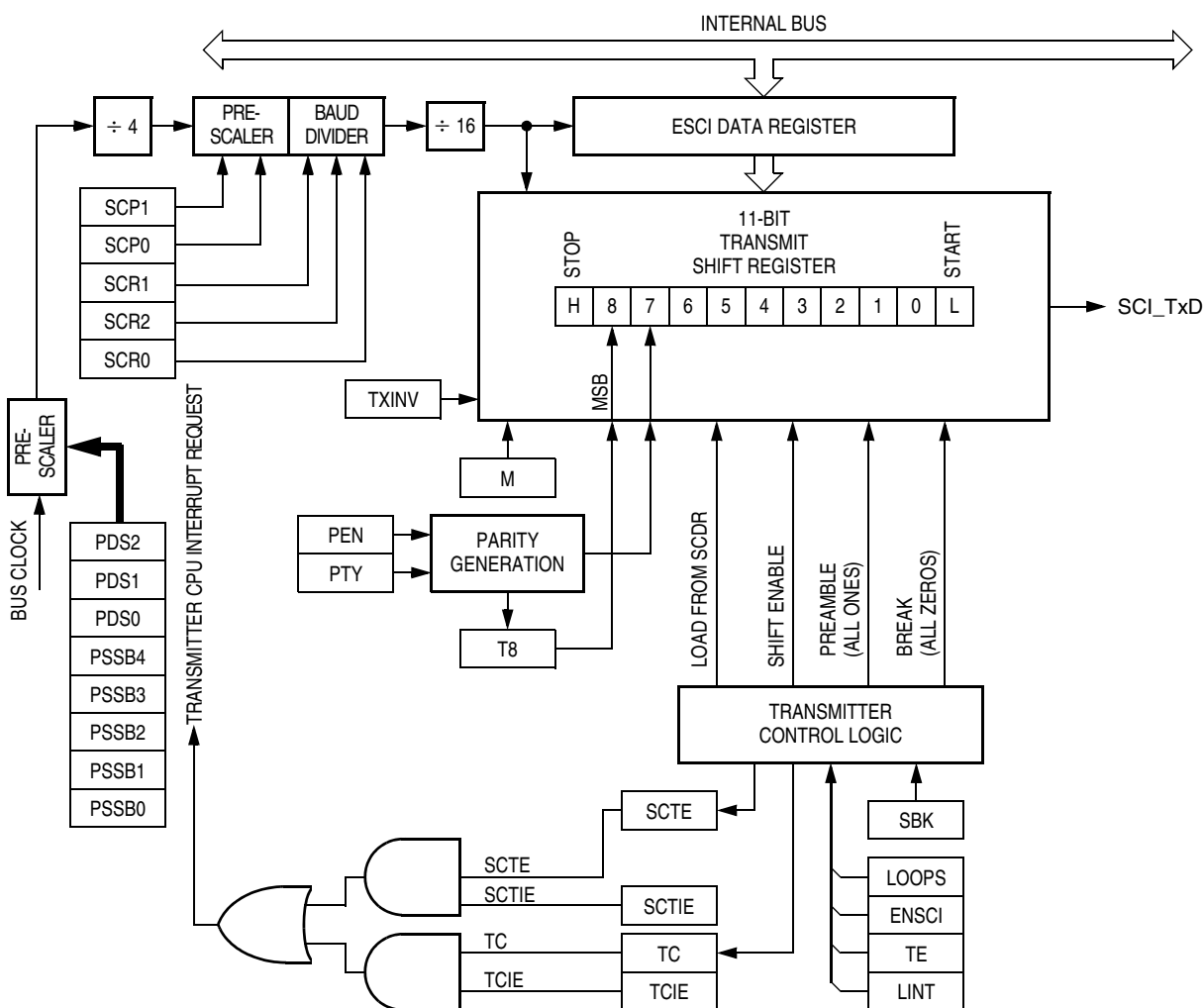


Figure 15-5. ESCI Transmitter

15.9 ESCI Arbiter

The ESCI module comprises an arbiter module designed to support software for communication tasks as bus arbitration, baud rate recovery and break time detection. The arbiter module consists of an 9-bit counter with 1-bit overflow and control logic. The CPU can control operation mode via the ESCI arbiter control register (SCIACTL).

15.9.1 ESCI Arbiter Control Register

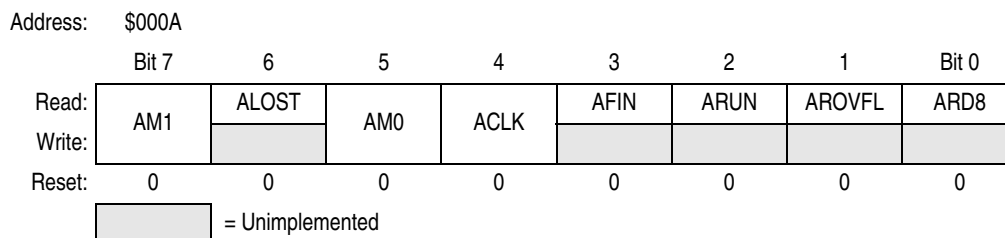


Figure 15-19. ESCI Arbiter Control Register (SCIACTL)

AM1 and AM0 — Arbiter Mode Select Bits

These read/write bits select the mode of the arbiter module as shown in Table 15-12. Reset clears AM1 and AM0.

Table 15-12. ESCI Arbiter Selectable Modes

AM[1:0]	ESCI Arbiter Mode
0 0	Idle / counter reset
0 1	Bit time measurement
1 0	Bus arbitration
1 1	Reserved / do not use

Alost — Arbitration Lost Flag

This read-only bit indicates loss of arbitration. Clear Alost by writing a logic 0 to AM1. Reset clears Alost.

ACLK — Arbiter Counter Clock Select Bit

This read/write bit selects the arbiter counter clock source. Reset clears ACLK.

- 1 = Arbiter counter is clocked with one quarter of the ESCI input clock generated by the ESCI prescaler
- 0 = Arbiter counter is clocked with the bus clock divided by four

NOTE

For ACLK = 1, the Arbiter input clock is driven from the ESCI prescaler. The prescaler can be clocked by either the bus clock or CGMXCLK depending on the state of the ESCIBDSRC bit in CONFIG2.

AFIN— Arbiter Bit Time Measurement Finish Flag

This read-only bit indicates bit time measurement has finished. Clear AFIN by writing any value to SCIACTL. Reset clears AFIN.

- 1 = Bit time measurement has finished
- 0 = Bit time measurement not yet finished

System Integration Module (SIM)

The SIM is responsible for:

- Bus clock generation and control for CPU and peripherals:
 - Stop/wait/reset/break entry and recovery
 - Internal clock control
- Master reset control, including power-on reset (POR) and computer operating properly (COP) timeout
- Interrupt control:
 - Acknowledge timing
 - Arbitration control timing
 - Vector address generation
- CPU enable/disable timing
- Modular architecture expandable to 128 interrupt sources

Table 16-1 shows the internal signal names used in this section.

Table 16-1. Signal Name Conventions

Signal Name	Description
CGMXCLK	Buffered version of OSC1 from clock generator module (CGM)
CGMVCLK	PLL output
CGMOUT	PLL-based or OSC1-based clock output from CGM module (Bus clock = CGMOUT divided by two)
IAB	Internal address bus
IDB	Internal data bus
PORRST	Signal from the power-on reset module to the SIM
IRST	Internal reset signal
R/W	Read/write signal

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$FE00	SIM Break Status Register (SBSR) See page 228.	Read:	R	R	R	R	R	R	SBSW	R
		Write:							Note ⁽¹⁾	
		Reset:	0	0	0	0	0	0	0	0
1. Writing a logic 0 clears SBSW.										
\$FE01	SIM Reset Status Register (SRSR) See page 228.	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
\$FE03	SIM Break Flag Control Register (SBFCR) See page 229.	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
\$FE04	Interrupt Status Register 1 (INT1) See page 224.	Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE05	Interrupt Status Register 2 (INT2) See page 224.	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE06	Interrupt Status Register 3 (INT3) See page 224.	Read:	0	0	IF20	IF19	IF18	IF17	IF16	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
<div><div></div> = Unimplemented</div> <div><div>R</div> = Reserved</div>										

= Unimplemented R = Reserved

Figure 16-2. SIM I/O Register Summary

16.3.2.6 Monitor Mode Entry Module Reset (MODRST)

The monitor mode entry module reset (MODRST) asserts its output to the SIM when monitor mode is entered in the condition where the reset vectors are erased (\$FF) (see 20.3.1.1 Normal Monitor Mode). When MODRST gets asserted, an internal reset occurs. The SIM actively pulls down the \overline{RST} pin for all internal reset sources.

16.4 SIM Counter

The SIM counter is used by the power-on reset module (POR) and in stop mode recovery to allow the oscillator time to stabilize before enabling the internal bus (IBUS) clocks. The SIM counter is 13 bits long.

16.4.1 SIM Counter During Power-On Reset

The power-on reset module (POR) detects power applied to the MCU. At power-on, the POR circuit asserts the signal PORRST. Once the SIM is initialized, it enables the clock generation module (CGM) to drive the bus clock state machine.

16.4.2 SIM Counter During Stop Mode Recovery

The SIM counter also is used for stop mode recovery. The STOP instruction clears the SIM counter. After an interrupt, break, or reset, the SIM senses the state of the short stop recovery bit, SSREC, in the mask option register. If the SSREC bit is a logic 1, then the stop recovery is reduced from the normal delay of 4096 CGMXCLK cycles down to 32 CGMXCLK cycles. This is ideal for applications using canned oscillators that do not require long startup times from stop mode. External crystal applications should use the full stop recovery time, that is, with SSREC cleared.

16.4.3 SIM Counter and Reset States

External reset has no effect on the SIM counter. See 16.6.2 Stop Mode for details. The SIM counter is free-running after all reset states. See 16.3.2 Active Resets from Internal Sources for counter control and internal reset recovery sequences.

16.5 Exception Control

Normal, sequential program execution can be changed in three different ways:

- Interrupts:
 - Maskable hardware CPU interrupts
 - Non-maskable software interrupt instruction (SWI)
- Reset
- Break interrupts

16.5.1 Interrupts

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. Figure 16-8 shows interrupt entry timing. Figure 16-9 shows interrupt recovery timing.

17.4.2 Slave Mode

The SPI operates in slave mode when the SPMSTR bit is clear. In slave mode, the SPSCCK pin is the input for the serial clock from the master MCU. Before a data transmission occurs, the \overline{SS} pin of the slave SPI must be at logic 0. \overline{SS} must remain low until the transmission is complete. See 17.7.2 Mode Fault Error.

In a slave SPI module, data enters the shift register under the control of the serial clock from the master SPI module. After a byte enters the shift register of a slave SPI, it transfers to the receive data register, and the SPRF bit is set. To prevent an overflow condition, slave software then must read the receive data register before another full byte enters the shift register.

The maximum frequency of the SPSCCK for an SPI configured as a slave is the bus clock speed (which is twice as fast as the fastest master SPSCCK clock that can be generated). The frequency of the SPSCCK for an SPI configured as a slave does not have to correspond to any SPI baud rate. The baud rate only controls the speed of the SPSCCK generated by an SPI configured as a master. Therefore, the frequency of the SPSCCK for an SPI configured as a slave can be any frequency less than or equal to the bus speed.

When the master SPI starts a transmission, the data in the slave shift register begins shifting out on the MISO pin. The slave can load its shift register with a new byte for the next transmission by writing to its transmit data register. The slave must write to its transmit data register at least one bus cycle before the master starts the next transmission. Otherwise, the byte already in the slave shift register shifts out on the MISO pin. Data written to the slave shift register during a transmission remains in a buffer until the end of the transmission.

When the clock phase bit (CPHA) is set, the first edge of SPSCCK starts a transmission. When CPHA is clear, the falling edge of \overline{SS} starts a transmission. See 17.5 Transmission Formats.

NOTE

SPSCCK must be in the proper idle state before the slave is enabled to prevent SPSCCK from appearing as a clock edge.

17.5 Transmission Formats

During an SPI transmission, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). A serial clock synchronizes shifting and sampling on the two serial data lines. A slave select line allows selection of an individual slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. On a master SPI device, the slave select line can optionally be used to indicate multiple-master bus contention.

17.5.1 Clock Phase and Polarity Controls

Software can select any of four combinations of serial clock (SPSCCK) phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, which selects an active high or low clock and has no significant effect on the transmission format.

The clock phase (CPHA) control bit selects one of two fundamentally different transmission formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

NOTE

Before writing to the CPOL bit or the CPHA bit, disable the SPI by clearing the SPI enable bit (SPE).

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).

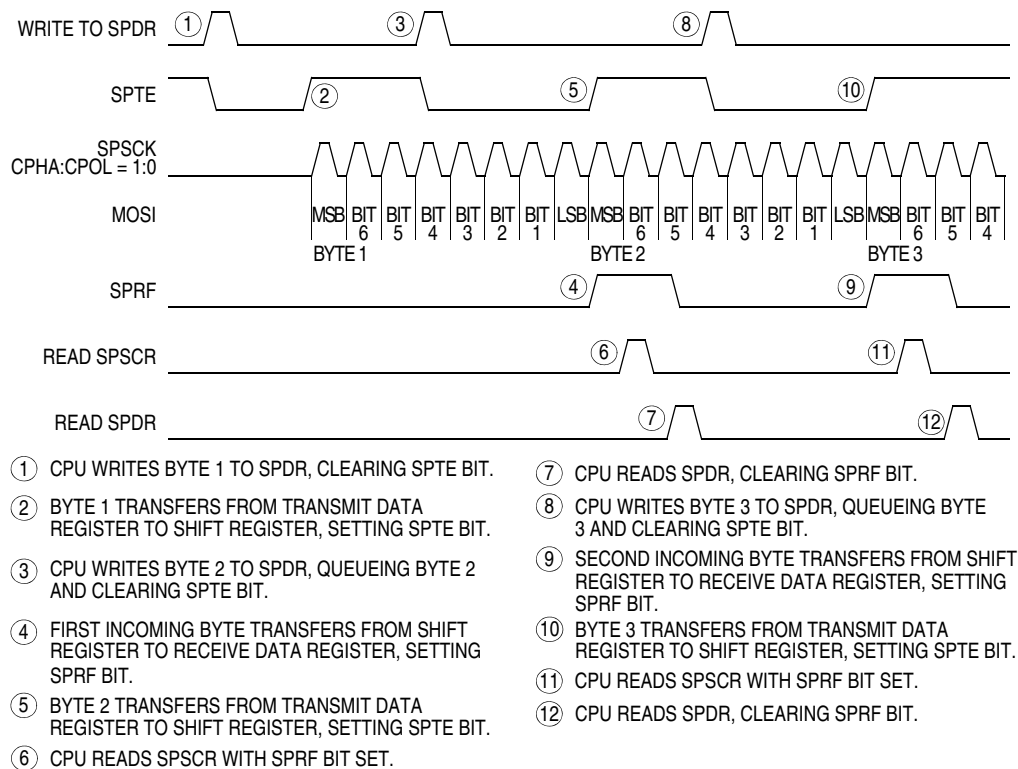


Figure 17-9. SPRF/SPTE CPU Interrupt Timing

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.

Timer Interface Module (TIM)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0020	Timer 1 Status and Control Register (T1SC) See page 265.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
\$0021	Timer 1 Counter Register High (T1CNTH) See page 266.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0022	Timer 1 Counter Register Low (T1CNTL) See page 266.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0023	Timer 1 Counter Modulo Register High (T1MODH) See page 267.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
\$0024	Timer 1 Counter Modulo Register Low (T1MODL) See page 267.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
\$0025	Timer 1 Channel 0 Status and Control Register (T1SC0) See page 267.	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
\$0026	Timer 1 Channel 0 Register High (T1CH0H) See page 270.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
\$0027	Timer 1 Channel 0 Register Low (T1CH0L) See page 270.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
\$0028	Timer 1 Channel 1 Status and Control Register (T1SC1) See page 267.	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
\$0029	Timer 1 Channel 1 Register High (T1CH1H) See page 270.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
\$002A	Timer 1 Channel 1 Register Low (T1CH1L) See page 270.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
\$002B	Timer 2 Status and Control Register (T2SC) See page 265.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
\$002C	Timer 2 Counter Register High (T2CNTH) See page 266.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0

□ = Unimplemented

Figure 19-3. TIM I/O Register Summary (Sheet 1 of 2)

19.9.1 TIM Status and Control Register

The TIM status and control register (TSC):

- Enables TIM overflow interrupts
- Flags TIM overflows
- Stops the TIM counter
- Resets the TIM counter
- Prescales the TIM counter clock

Address: T1SC, \$0020 and T2SC, \$002B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
Write:	0			TRST				
Reset:	0	0	1	0	0	0	0	0


 = Unimplemented

Figure 19-5. TIM Status and Control Register (TSC)

TOF — TIM Overflow Flag Bit

This read/write flag is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. Clear TOF by reading the TIM status and control register when TOF is set and then writing a logic 0 to TOF. If another TIM overflow occurs before the clearing sequence is complete, then writing logic 0 to TOF has no effect. Therefore, a TOF interrupt request cannot be lost due to inadvertent clearing of TOF. Reset clears the TOF bit. Writing a logic 1 to TOF has no effect.

1 = TIM counter has reached modulo value

0 = TIM counter has not reached modulo value

TOIE — TIM Overflow Interrupt Enable Bit

This read/write bit enables TIM overflow interrupts when the TOF bit becomes set. Reset clears the TOIE bit.

1 = TIM overflow interrupts enabled

0 = TIM overflow interrupts disabled

TSTOP — TIM Stop Bit

This read/write bit stops the TIM counter. Counting resumes when TSTOP is cleared. Reset sets the TSTOP bit, stopping the TIM counter until software clears the TSTOP bit.

1 = TIM counter stopped

0 = TIM counter active

NOTE

Do not set the TSTOP bit before entering wait mode if the TIM is required to exit wait mode.

TRST — TIM Reset Bit

Setting this write-only bit resets the TIM counter and the TIM prescaler. Setting TRST has no effect on any other registers. Counting resumes from \$0000. TRST is cleared automatically after the TIM counter is reset and always reads as logic 0. Reset clears the TRST bit.

1 = Prescaler and TIM counter cleared

0 = No effect

NOTE

Setting the TSTOP and TRST bits simultaneously stops the TIM counter at a value of \$0000.

Table 19-3. Mode, Edge, and Level Selection

MSxB:MSxA	ELSxB:ELSxA	Mode	Configuration
X0	00	Output preset	Pin under port control; initial output level high
X1	00		Pin under port control; initial output level low
00	01	Input capture	Capture on rising edge only
00	10		Capture on falling edge only
00	11		Capture on rising or falling edge
01	01	Output compare or PWM	Toggle output on compare
01	10		Clear output on compare
01	11		Set output on compare
1X	01	Buffered output compare or buffered PWM	Toggle output on compare
1X	10		Clear output on compare
1X	11		Set output on compare

NOTE

Before enabling a TIM channel register for input capture operation, make sure that the PTD/TCHx pin is stable for at least two bus clocks.

TOVx — Toggle On Overflow Bit

When channel x is an output compare channel, this read/write bit controls the behavior of the channel x output when the TIM counter overflows. When channel x is an input capture channel, TOVx has no effect. Reset clears the TOVx bit.

1 = Channel x pin toggles on TIM counter overflow.

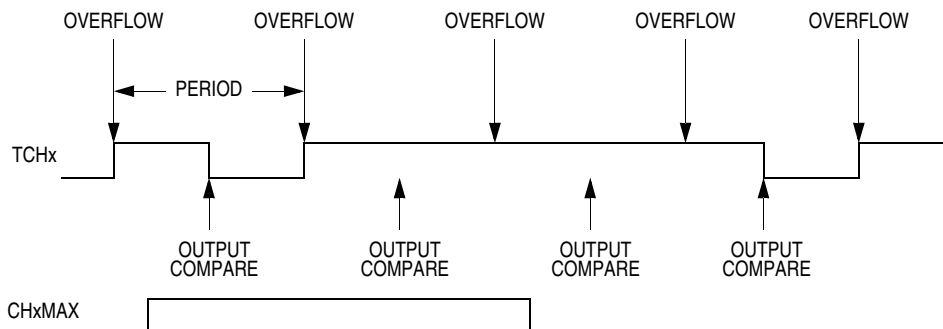
0 = Channel x pin does not toggle on TIM counter overflow.

NOTE

When TOVx is set, a TIM counter overflow takes precedence over a channel x output compare if both occur at the same time.

CHxMAX — Channel x Maximum Duty Cycle Bit

When the TOVx bit is at logic 1, setting the CHxMAX bit forces the duty cycle of buffered and unbuffered PWM signals to 100%. As Figure 19-12 shows, the CHxMAX bit takes effect in the cycle after it is set or cleared. The output stays at the 100% duty cycle level until the cycle after CHxMAX is cleared.


Figure 19-12. CHxMAX Latency

21.5 5-Vdc Electrical Characteristics

Characteristic ⁽¹⁾	Symbol	Min	Typ ⁽²⁾	Max	Unit
Output high voltage ($I_{Load} = -2.0$ mA) all I/O pins ($I_{Load} = -10.0$ mA) all I/O pins ($I_{Load} = -20.0$ mA) pins PTC0–PTC4 only Maximum combined I_{OH} for port PTA7–PTA3, port PTC0–PTC1, port E, port PTD0–PTD3 Maximum combined I_{OH} for port PTA2–PTA0, port B, port PTC2–PTC6, port PTD4–PTD7 Maximum total I_{OH} for all port pins	V_{OH} V_{OH} V_{OH} I_{OH1} I_{OH2} I_{OHT}	$V_{DD} - 0.8$ $V_{DD} - 1.5$ $V_{DD} - 1.5$ — — —	— — — — — —	— — — 50 50 100	V V V mA mA mA
Output low voltage ($I_{Load} = 1.6$ mA) all I/O pins ($I_{Load} = 10$ mA) all I/O pins ($I_{Load} = 20$ mA) pins PTC0–PTC4 only Maximum combined I_{OH} for port PTA7–PTA3, port PTC0–PTC1, port E, port PTD0–PTD3 Maximum combined I_{OH} for port PTA2–PTA0, port B, port PTC2–PTC6, port PTD4–PTD7 Maximum total I_{OL} for all port pins	V_{OL} V_{OL} V_{OL} I_{OL1} I_{OL2} I_{OLT}	— — — — — —	— — — — — —	0.4 1.5 1.5 50 50 100	V V V mA mA 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}	—	$0.2 \times V_{DD}$	V
V_{DD} supply current Run ⁽³⁾ Wait ⁽⁴⁾ Stop ⁽⁵⁾ Stop with TBM enabled ⁽⁶⁾ Stop with LVI and TBM enabled ⁽⁶⁾ Stop with LVI	I_{DD}	— — — — — —	20 6 0.6 1 1.25 250	30 12 10 1.25 1.6 350	mA mA μ A mA mA μ A
DC injection current ^{(7) (8) (9) (10)} Single pin limit $V_{in} > V_{DD}$ $V_{in} < V_{SS}$ Total MCU limit, includes sum of all stressed pins $V_{in} > V_{DD}$ $V_{in} < V_{SS}$	I_{IC}	0 0 0 0	— — — —	2 –0.2 25 –5	mA
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/CAN _{TX} , 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