68HC05JB4
68HC705JB4

SPECIFICATION
(General Release)

February 24, 1999

Semiconductor Products Sector

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.

TABLE OF CONTENTS

Section **Page**

SECTION 1
GENERAL DESCRIPTION

SECTION 2
MEMORY

SECTION 3
CENTRAL PROCESSING UNIT

SECTION 4
INTERRUPTS

TABLE OF CONTENTS

Section **Page**

TABLE OF CONTENTS

Section **Page**

SECTION 8
MULTI-FUNCTION TIMER

SECTION 9
16-BIT TIMER

SECTION 10
UNIVERSAL SERIAL BUS MODULE

TABLE OF CONTENTS

Section **Page**

TABLE OF CONTENTS

Section **Page**

SECTION 14
MECHANICAL SPECIFICATIONS

APPENDIX A
MC68HC705JB4

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES

Table **Title** **Page**

SECTION 1
GENERAL DESCRIPTION

The MC68HC05JB4 is a member of the low-cost, high-performance M68HC05 Family of 8-bit microcontroller units (MCUs). The M68HC05 Family is based on the customer-specified integrated circuit (CSIC) design strategy. All MCUs in the family use the popular M68HC05 central processing unit (CPU) and are available with a variety of subsystems, memory sizes and types, and package types.

The MC68HC05JB4 is specifically designed to be used in applications where a low speed (1.5Mbps) Universal Serial Bus (USB) interface is required.

1.1 FEATURES

- Industry standard M68HC05 CPU core
- Memory-mapped input/output (I/O) registers
- 3584 Bytes of user ROM
- 176 Bytes of user RAM (includes 64 byte stack)
- 19 Bidirectional I/O pins with the following added features:
  - PA[0:7]: Software enable Internal pull-up resistor (50kΩ typical)
  - PA[0:3]: Built-in schmitt triggered input level
  - Maskable as extra input sources for IRQ interrupt
  - Maskable Negative-Edge Only or Negative-Edge
  - and Low-Level Interrupt Capability
  - PA4: $\overline{\text{IRQ2}}$ with built-in schmitt triggered input
  - PA[5:7]: 10mA Sink output drive
  - PA[6:7]: Maskable 10mA/25mA sink output drive
  - Software enable Slow Edge Pull Down Devices
  - PB[0:4]: Software enable Internal pull-up resistor (50kΩ typical)
  - Software enable Slow Edge Pull Down Devices
  - PB[0]: ICAP1 with built-in schmitt triggered input
  - PB[3:4]: AD[4:5]
  - PC[0:3]: AD[0:3]

- Fully compliant to Low Speed USB with 3 Endpoints:
  - 1 Control Endpoint (2x8 byte buffer)
  - 2 Interrupt Endpoints (1x8 byte buffer shared)
- 6-channel 8-bit Analog-to-Digital Converter (ADC)
- Multi-function Timer (MFT)
- 16-bit Timer with 1 input capture and 1 output compare
- Low Voltage Reset (LVR) Circuit
- Computer Operating Properly (COP) Watchdog Reset
- Provide a 3.3V ±10% DC Voltage for USB pull-up resistor
- Fully Static Operation with no Minimum Clock Speed
- Illegal Address Reset
- Power-Saving STOP and WAIT Modes
- Available in 28-Pin PDIP and 28-Pin SOIC packages

1.2 MASK OPTIONS

The following mask options are available:

- Enable PA0 to PA3 as extra IRQ interrupt sources.
- External interrupt pins ($\overline{IRQ}$, PA0 to PA3): negative edge-triggered or negative edge- and low level-triggered.
- High current (25mA) output on PA6 and PA7.
- OSC: crystal/ceramic resonator start up delay, 4064 or 128 clock cycles.
- LVR: enabled or disabled.
- COP: enabled or disabled.

1.3 MCU STRUCTURE

Figure 1-1 shows the structure of MC68HC05JB4 MCU.

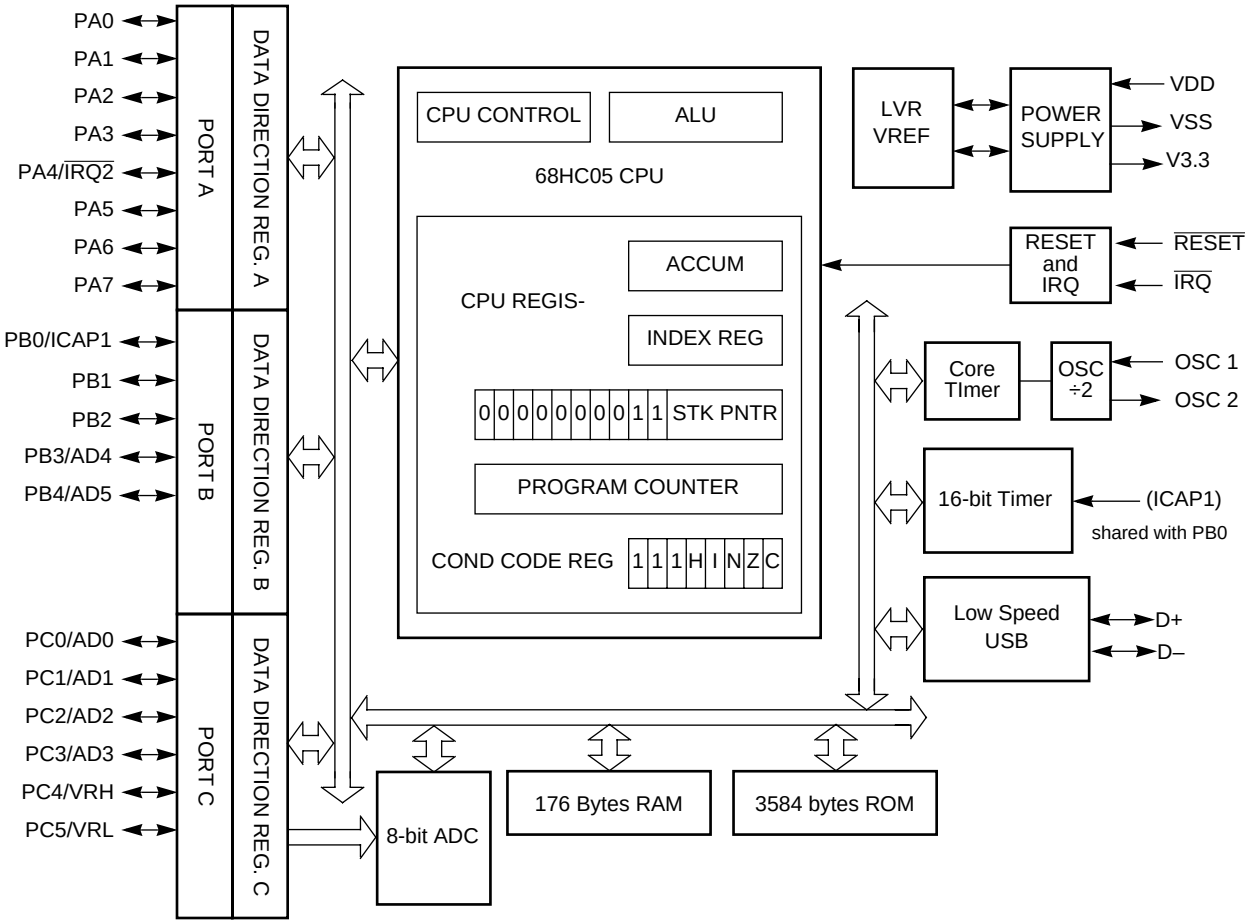


Figure 1-1. MC68HC05JB4 Block Diagram

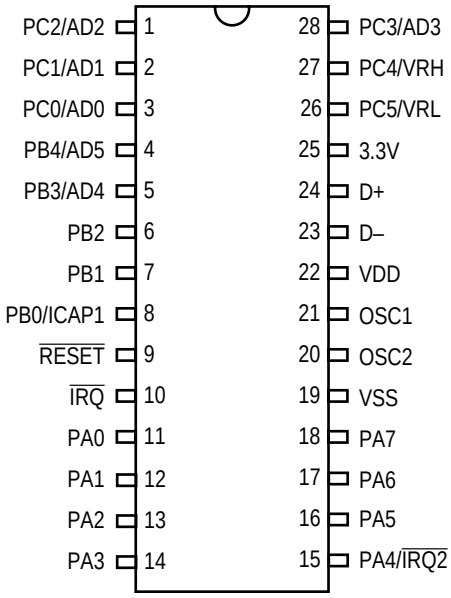


Figure 1-2. MC68HC05JB4 Pin Assignment

1.4 FUNCTIONAL PIN DESCRIPTION

The following paragraphs give a description of the general function of each pin assigned in Fig. 1-2 and Fig. 1-3.

1.4.1 V_{DD} AND V_{SS}

Power is supplied to the MCU through V_{DD} and V_{SS}. V_{DD} is the positive supply, and V_{SS} is ground. The MCU operates from a single power supply.

Very fast signal transitions occur on the MCU pins. The short rise and fall times place very high short-duration current demands on the power supply. To prevent noise problems, special care should be taken to provide good power supply bypassing at the MCU by using bypass capacitors with good high-frequency characteristics that are positioned as close to the MCU as possible. Bypassing requirements vary, depending on how heavily the MCU pins are loaded.

1.4.2 OSC1, OSC2

The OSC1 and OSC2 pins are the connections for the on-chip oscillator. The OSC1 and OSC2 pins can accept the following sets of components:

1. A crystal as shown in **Figure 1-3**(a)
2. A ceramic resonator as shown in **Figure 1-3**(a)
3. An external clock signal as shown in **Figure 1-3**(b)

The frequency, f_{OSC}, of the oscillator or external clock source is divided by two to produce the internal operating frequency, f_{OP}. If the internal operating frequency is 3MHz, then the external oscillator frequency will be 6MHz. For LS USB 1.5MHz

frequency clock can be derived from a divided by 4 circuit. The type of oscillator is selected by a mask option. An internal 2MΩ resistor may be selected between OSC1 and OSC2 by a mask option (crystal/ceramic resonator mode only).

Crystal Oscillator

The circuit in **Figure 1-3**(a) shows a typical oscillator circuit for an AT-cut, parallel resonant crystal. The crystal manufacturer's recommendations should be followed, as the crystal parameters determine the external component values required to provide maximum stability and reliable start-up. The load capacitance values used in the oscillator circuit design should include all stray capacitances. The crystal and components should be mounted as close as possible to the pins for start-up stabilization and to minimize output distortion. An internal start-up resistor of approximately 2 MΩ is provided between OSC1 and OSC2 for the crystal type oscillator as a mask option.

(a) Crystal or Ceramic Resonator Connections

(b) External Clock Source Connection

Figure 1-3. Oscillator Connections

Ceramic Resonator Oscillator

In cost-sensitive applications, a ceramic resonator can be used in place of the crystal. The circuit in **Figure 1-3**(a) can be used for a ceramic resonator. The resonator manufacturer's recommendations should be followed, as the resonator parameters determine the external component values required for maximum stability and reliable starting. The load capacitance values used in the oscillator circuit design should include all stray capacitances. The ceramic resonator and components should be mounted as close as possible to the pins for start-up stabilization and to minimize output distortion. An internal start-up resistor of approximately 2 MΩ is provided between OSC1 and OSC2 for the ceramic resonator type oscillator as a mask option.

External Clock

An external clock from another CMOS-compatible device can be connected to the OSC1 input, with the OSC2 input not connected, as shown in **Figure 1-3**(b).This configuration is possible ONLY when the crystal/ceramic resonator mask option is selected.

1.4.3 RESET

This is an I/O pin. This pin can be used as an input to reset the MCU to a known start-up state by pulling it to the low state. The $\overline{\text{RESET}}$ pin contains a steering diode to discharge any voltage on the pin to V_{DD}, when the power is removed. An internal pull-up is also connected between this pin and V_{DD}. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger to improve its noise immunity as an input. This pin is an output pin if LVR triggers an internal reset.

1.4.4 $\overline{\text{IRQ}}$ (MASKABLE INTERRUPT REQUEST)

This input pin drives the asynchronous IRQ interrupt function of the CPU. The IRQ interrupt function has a mask option to provide either only negative edge-sensitive triggering or both negative edge-sensitive and low level-sensitive triggering. If the option is selected to include level-sensitive triggering, the $\overline{\text{IRQ}}$ input requires an external resistor to V_{DD} for "wired-OR" operation, if desired. The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity.

NOTE

Each of the PA0 thru PA3 I/O pins may be connected as an OR function with the IRQ interrupt function by a mask option. This capability allows keyboard scan applications where the transitions or levels on the I/O pins will behave the same as the $\overline{\text{IRQ}}$ pin. The edge or level sensitivity selected by a separate mask option for the $\overline{\text{IRQ}}$ pin also applies to the I/O pins OR'ed to create the IRQ signal.

1.4.5 V3.3

This is an output reference voltage nominally set at 3.3 volt DC.

1.4.6 D+ and D–

These two lines carry the USB differential data. For low speed device such as MC68HC05JB4, a 1.5 kΩ resistor is required to be connected across D– and 3.3V for proper signal termination.

1.4.7 PA0-PA7

These eight I/O lines comprise Port-A. PA0 to PA7 are push-pull pins with internal pull-up resistors. The state of any pin is software programmable and all Port A lines are configured as inputs during power-on or reset. The internal pull-up resistor on PA0-4 is software enable. The PA0 thru PA3 can be connected via an internal NAND gate to the IRQ interrupt function enabled by a mask option. PA5 thru PA7 has built in 10mA pull-down device for direct LED drive. In addition, PA6 and PA7 both have Slow Falling Edge Control which is enabled by software and can sink 25mA current selectable by mask option. PA0 thru PA4 have built-in schmitt triggered input. PA4 can be used as an extra interrupt pin ($\overline{\text{IRQ2}}$) when IRQ2 interrupt is enabled.

1.4.8 PB0-PB4

These five I/O lines comprise Port-B. PB0 to PB4 are push-pull pins with internal pull-up resistors. The state of any pin is software programmable and all Port B lines are configured as inputs during power-on or reset. The internal pull-up resistor is software enable. In addition, all Port-B pins have Slow Falling Edge Control which is enabled by software. PB0 can be used as the input capture pin when the input capture function is enabled on the 16-bit Timer. PB0 has built-in schmitt triggered input. When the ADC function is enabled, PB3 and PB4 can be used as extra two analog input channels (AD4 and AD5 respectively) to the ADC.

1.4.9 PC0-PC5

These six I/O lines comprise Port-C. PC0 to PC5 are push-pull pins. The state of any pin is software programmable and all Port C lines are configured as inputs during power-on or reset. When the ADC function is enabled, PC0 thru PC3 become the four analog input channels to the ADC and PC4 and PC5 become the analog “High” and “Low” reference voltages to the ADC respectively.

SECTION 2
MEMORY

The MC68HC05JB4 has 8k-bytes of addressable memory, with 64 bytes of I/O, 176 bytes of user RAM, and 3584 bytes of user ROM, as shown in **Figure 2-1**.

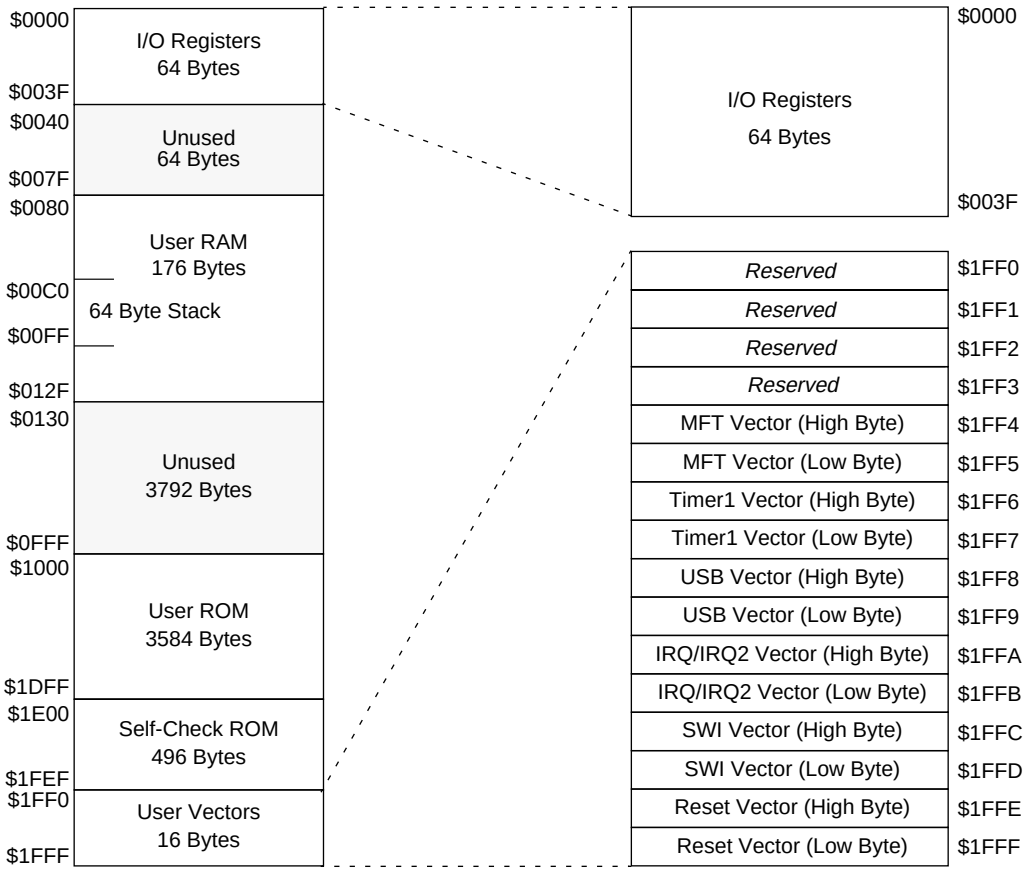


Figure 2-1. MC68HC05JB4 Memory Map

2.1 I/O AND CONTROL REGISTERS

The I/O and Control Registers reside in locations $0000 to $003F. The bit assignments for each register are shown in **Figure 2-2**, **Figure 2-3**, **Figure 2-4**, and **Figure 2-5**. Reading from unused bits will return unknown states, and writing to unused bits will be ignored.

2.2 RAM

The user RAM consists of 176 bytes (including the stack) at locations $0080 to $012F. The stack begins at address $00FF and proceeds down to $00C0. Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

2.3 ROM

There are a total of 4k bytes of ROM on chip. This includes 3584 bytes of user ROM with locations $1000 to $1DFF for user program storage and 16 bytes for user vectors at locations $1FF0 to $1FFF. Also, 496 bytes of Self-check ROM on chip at locations $1E00 to $1FEF.

2.4 I/O REGISTERS SUMMARY

ADDR	REGISTER	R/W	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
$0000	Port A Data PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		W								
$0001	Port B Data PORTB	R	0	0	0	PB4	PB3	PB2	PB1	PB0
		W								
$0002	Port C Data PORTC	R	0	0	PC5	PC4	PC3	PC2	PC1	PC0
		W								
$0003	*Unused*	*R*								
		W								
$0004	Port A Data Direction DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
$0005	Port B Data Direction DDRB	R	SLOWEA	SLOWEB	0	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		W								
$0006	Port C Data Direction DDRC	R	0	0	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		W								
$0007	*Unused*	*R*								
		W								
$0008	MFT Ctrl/Status TCSR	R	TOF	RTIF	TOFE	RTIE	0	0	RT1	RT0
		W					TOFR	RTIFR		
$0009	MFT Counter TCNT	R	TMR7	TMR6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0
		W								
$000A	IRQ Control/Status ICSR	R	IRQE	IRQ2E	0	0	IRQF	IRQ2F	0	0
		W							IRQR	IRQ2R
$000B	*Unused*	*R*								
		W								
$000C	*Unused*	*R*								
		W								
$000D	*Unused*	*R*								
		W								
$000E	ADC Control/Status ADSCR	R	COCO	ADRC	ADON	0	CH3	CH2	CH1	CH0
		W								
$000F	ADC Data ADDR	R	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
		W								

Figure 2-2. MC68HC05JB4 I/O Registers $0000-$000F

unused bits reserved bits

ADDR	REGISTER	R/W	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
$0010	Port A Pull-Up	R								
	PURA	W	PURA7	PURA6	PURA5	PURA4	PURA3	PURA2	PURA1	PURA0
$0011	Port B Pull-Up	R								
	PURB	W				PURB4	PURB3	PURB2	PURB1	PURB0
$0012	Timer1 Control	R	ICIE	OCIE	TOIE	0	0	0	IEDG	0
	TCR	W								
$0013	Timer1 Status	R	ICF	OCF	TOF	0	0	0	0	0
	TSR	W								
$0014	Input Capture MSB	R	ICH7	ICH6	ICH5	ICH4	ICH3	ICH2	ICH1	ICH0
	ICH	W								
$0015	Input Capture LSB	R	ICL7	ICL6	ICL5	ICL4	ICL3	ICL2	ICL1	ICL0
	ICL	W								
$0016	Output Compare MSB	R	OCH7	OCH6	OCH5	OCH4	OCH3	OCH2	OCH1	OCH0
	OCH	W								
$0017	Output Compare LSB	R	OCL7	OCL6	OCL5	OCL4	OCL3	OCL2	OCL1	OCL0
	OCL	W								
$0018	Timer1 Counter MSB	R	TCNTH7	TCNTH6	TCNTH5	TCNTH4	TCNTH3	TCNTH2	TCNTH1	TCNTH0
	TCNTH	W								
$0019	Timer1 Counter LSB	R	TCNTL7	TCNTL6	TCNTL5	TCNTL4	TCNTL3	TCNTL2	TCNTL1	TCNTL0
	TCNTL	W								
$001A	Alter. Counter MSB	R	ACNTH7	ACNTH6	ACNTH5	ACNTH4	ACNTH3	ACNTH2	ACNTH1	ACNTH0
	ACNTH	W								
$001B	Alter. Counter LSB	R	ACNTL7	ACNTL6	ACNTL5	ACNTL4	ACNTL3	ACNTL2	ACNTL1	ACNTL0
	ACNTL	W								
$001C	*Unused*	*R*								
		W								
$001D	*Unused*	*R*								
		W								
$001E	*Unused*	*R*								
		W								
$001F	*Unused*	*R*								
		W								

Figure 2-3. MC68HC05JB4 I/O Registers $0010-$001F

ADDR	REGISTER	R/W	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
$0020	USB Endpoint 0 Data 0	R	UE0RD7	UE0RD6	UE0RD5	UE0RD4	UE0RD3	UE0RD2	UE0RD1	UE0RD0
	UD0R0	W	UE0TD7	UE0TD6	UE0TD5	UE0TD4	UE0TD3	UE0TD2	UE0TD1	UE0TD0
$0021	USB Endpoint 0 Data 1	R	UE0RD7	UE0RD6	UE0RD5	UE0RD4	UE0RD3	UE0RD2	UE0RD1	UE0RD0
	UD0R1	W	UE0TD7	UE0TD6	UE0TD5	UE0TD4	UE0TD3	UE0TD2	UE0TD1	UE0TD0
$0022	USB Endpoint 0 Data 2	R	UE0RD7	UE0RD6	UE0RD5	UE0RD4	UE0RD3	UE0RD2	UE0RD1	UE0RD0
	UD0R2	W	UE0TD7	UE0TD6	UE0TD5	UE0TD4	UE0TD3	UE0TD2	UE0TD1	UE0TD0
$0023	USB Endpoint 0 Data 3	R	UE0RD7	UE0RD6	UE0RD5	UE0RD4	UE0RD3	UE0RD2	UE0RD1	UE0RD0
	UD0R3	W	UE0TD7	UE0TD6	UE0TD5	UE0TD4	UE0TD3	UE0TD2	UE0TD1	UE0TD0
$0024	USB Endpoint 0 Data 4	R	UE0RD7	UE0RD6	UE0RD5	UE0RD4	UE0RD3	UE0RD2	UE0RD1	UE0RD0
	UD0R4	W	UE0TD7	UE0TD6	UE0TD5	UE0TD4	UE0TD3	UE0TD2	UE0TD1	UE0TD0
$0025	USB Endpoint 0 Data 5	R	UE0RD7	UE0RD6	UE0RD5	UE0RD4	UE0RD3	UE0RD2	UE0RD1	UE0RD0
	UD0R5	W	UE0TD7	UE0TD6	UE0TD5	UE0TD4	UE0TD3	UE0TD2	UE0TD1	UE0TD0
$0026	USB Endpoint 0 Data 6	R	UE0RD7	UE0RD6	UE0RD5	UE0RD4	UE0RD3	UE0RD2	UE0RD1	UE0RD0
	UD0R6	W	UE0TD7	UE0TD6	UE0TD5	UE0TD4	UE0TD3	UE0TD2	UE0TD1	UE0TD0
$0027	USB Endpoint 0 Data 7	R	UE0RD7	UE0RD6	UE0RD5	UE0RD4	UE0RD3	UE0RD2	UE0RD1	UE0RD0
	UD0R7	W	UE0TD7	UE0TD6	UE0TD5	UE0TD4	UE0TD3	UE0TD2	UE0TD1	UE0TD0
$0028	USB Endpoint 1 Data 0	R	UE1RD7	UE1RD6	UE1RD5	UE1RD4	UE1RD3	UE1RD2	UE1RD1	UE1RD0
	UD1R0	W	UE1TD7	UE1TD6	UE1TD5	UE1TD4	UE1TD3	UE1TD2	UE1TD1	UE1TD0
$0029	USB Endpoint 1 Data 1	R	UE1RD7	UE1RD6	UE1RD5	UE1RD4	UE1RD3	UE1RD2	UE1RD1	UE1RD0
	UD1R1	W	UE1TD7	UE1TD6	UE1TD5	UE1TD4	UE1TD3	UE1TD2	UE1TD1	UE1TD0
$002A	USB Endpoint 1 Data 2	R	UE1RD7	UE1RD6	UE1RD5	UE1RD4	UE1RD3	UE1RD2	UE1RD1	UE1RD0
	UD1R2	W	UE1TD7	UE1TD6	UE1TD5	UE1TD4	UE1TD3	UE1TD2	UE1TD1	UE1TD0
$002B	USB Endpoint 1 Data 3	R	UE1RD7	UE1RD6	UE1RD5	UE1RD4	UE1RD3	UE1RD2	UE1RD1	UE1RD0
	UD1R3	W	UE1TD7	UE1TD6	UE1TD5	UE1TD4	UE1TD3	UE1TD2	UE1TD1	UE1TD0
$002C	USB Endpoint 1 Data 4	R	UE1RD7	UE1RD6	UE1RD5	UE1RD4	UE1RD3	UE1RD2	UE1RD1	UE1RD0
	UD1R4	W	UE1TD7	UE1TD6	UE1TD5	UE1TD4	UE1TD3	UE1TD2	UE1TD1	UE1TD0
$002D	USB Endpoint 1 Data 5	R	UE1RD7	UE1RD6	UE1RD5	UE1RD4	UE1RD3	UE1RD2	UE1RD1	UE1RD0
	UD1R5	W	UE1TD7	UE1TD6	UE1TD5	UE1TD4	UE1TD3	UE1TD2	UE1TD1	UE1TD0
$002E	USB Endpoint 1 Data 6	R	UE1RD7	UE1RD6	UE1RD5	UE1RD4	UE1RD3	UE1RD2	UE1RD1	UE1RD0
	UD1R6	W	UE1TD7	UE1TD6	UE1TD5	UE1TD4	UE1TD3	UE1TD2	UE1TD1	UE1TD0
$002F	USB Endpoint 1 Data 7	R	UE1RD7	UE1RD6	UE1RD5	UE1RD4	UE1RD3	UE1RD2	UE1RD1	UE1RD0
	UD1R7	W	UE1TD7	UE1TD6	UE1TD5	UE1TD4	UE1TD3	UE1TD2	UE1TD1	UE1TD0

unused bits reserved bits

Figure 2-4. MC68HC05JB4 I/O Registers $0020-$002F

ADDR	REGISTER	R/W	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
$0030	*Unused*	*R*								
		W								
$0031	*Unused*	*R*								
		W								
$0032	*Unused*	*R*								
		W								
$0033	*Unused*	*R*								
		W								
$0034	*Unused*	*R*								
		W								
$0035	*Unused*	*R*								
		W								
$0036	*Unused*	*R*								
		W								
$0037	USB Control 2	R		0	TX1ST		ENABLE2	ENABLE1	STALL2	STALL1
	UCR2	W		TX1STR						
$0038	USB Address	R	USBEN	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0
	UADR	W								
$0039	USB Interrupt 0	R	TXD0F	RXD0F	RSTF	SUSPND	TXD0IE	RXD0IE	0	0
	UIR0	W	0	0	0				TXD0FR	RXD0FR
$003A	USB Interrupt 1	R	TXD1F	RXD1F	RESUMF	0	TXD1IE	EOPIE	0	0
	UIR1	W	0	0	0	RESUMFR			TXD1FR	EOPFR
$003B	USB Control 0	R	T0SEQ	STALL0	TX0E	RX0E	TP0SIZ3	TP0SIZ2	TP0SIZ1	TP0SIZ0
	UCR0	W								
$003C	USB Control 1	R	T1SEQ	ENDADD	TX1E	FRESUM	TP1SIZ3	TP1SIZ2	TP1SIZ1	TP1SIZ0
	UCR1	W								
$003D	USB Status	R	RSEQ	SETUP			RPSIZ3	RPSIZ2	RPSIZ1	RPSIZ0
	USR	W								
$003E	*Reserved*	*R*								
		W								
$003F	*Reserved*	*R*								
		W								

unused bits reserved bits

Figure 2-5. MC68HC05JB4 I/O Registers $0030-$003F

ADDR	REGISTER	R/W	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
$1FF0	COP Register	R	0	0	0	0	0	0	0	0
	COPR	W								COPR

Figure 2-6. COP Register (COPR)

SECTION 3
CENTRAL PROCESSING UNIT

The MC68HC05JB4 has an 8k-bytes memory map. The stack has only 64 bytes. Therefore, the stack pointer has been reduced to only 6 bits and will only decrement down to $00C0 and then wrap-around to $00FF. All other instructions and registers behave as described in this chapter.

3.1 REGISTERS

The MCU contains five registers which are hard-wired within the CPU and are not part of the memory map. These five registers are shown in **Figure 3-1** and are described in the following paragraphs.

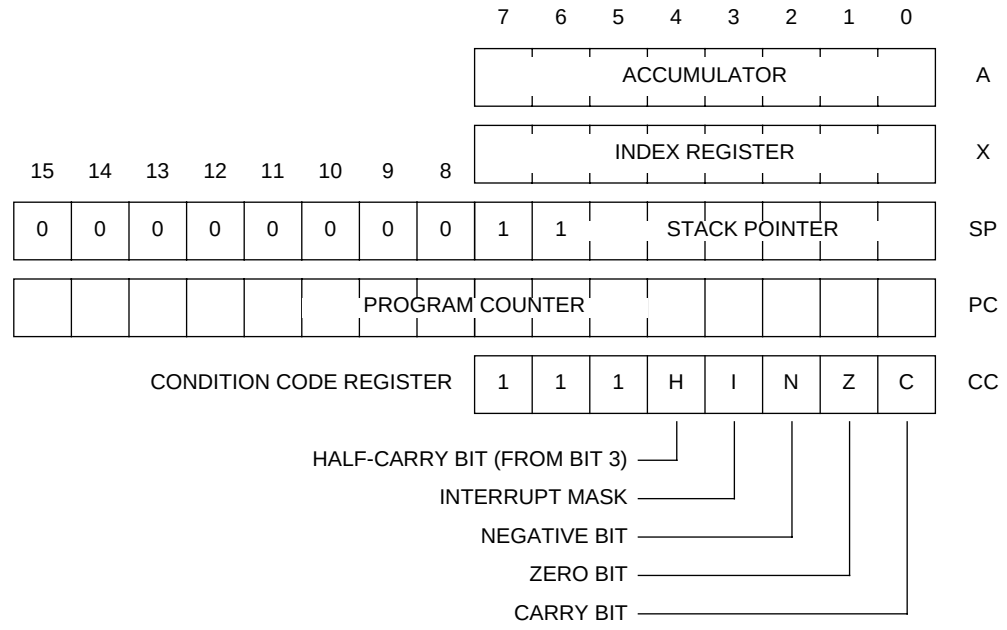


Figure 3-1. MC68HC05 Programming Model

3.2 ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register as shown in **Figure 3-1**. The CPU uses the accumulator to hold operands and results of arithmetic calculations or non-arithmetic operations. The accumulator is not affected by a reset of the device.

3.3 INDEX REGISTER (X)

The index register shown in **Figure 3-1** is an 8-bit register that can perform two functions:

- Indexed addressing
- Temporary storage

In indexed addressing with no offset, the index register contains the low byte of the operand address, and the high byte is assumed to be $00. In indexed addressing with an 8-bit offset, the CPU finds the operand address by adding the index register content to an 8-bit immediate value. In indexed addressing with a 16-bit offset, the CPU finds the operand address by adding the index register content to a 16-bit immediate value.

The index register can also serve as an auxiliary accumulator for temporary storage. The index register is not affected by a reset of the device.

3.4 STACK POINTER (SP)

The stack pointer shown in **Figure 3-1** is a 16-bit register. In MCU devices with memory space less than 64k-bytes the unimplemented upper address lines are ignored. The stack pointer contains the address of the next free location on the stack. During a reset or the reset stack pointer (RSP) instruction, the stack pointer is set to $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled off the stack.

When accessing memory, the ten most significant bits are permanently set to 0000000011. The six least significant register bits are appended to these ten fixed bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64($C0) locations. If 64 locations are exceeded, the stack pointer wraps around and overwrites the previously stored information. A subroutine call occupies two locations on the stack and an interrupt uses five locations.

3.5 PROGRAM COUNTER (PC)

The program counter shown in **Figure 3-1** is a 16-bit register. In MCU devices with memory space less than 64k-bytes the unimplemented upper address lines are ignored. The program counter contains the address of the next instruction or operand to be fetched.

Normally, the address in the program counter increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

3.6 CONDITION CODE REGISTER (CCR)

The CCR shown in **Figure 3-1** is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. The fifth bit is the interrupt mask. These bits can be individually tested by a program, and specific actions can be taken as a result of their states. The condition code register should be thought of as having three additional upper bits that are always ones. Only the interrupt mask is affected by a reset of the device. The following paragraphs explain the functions of the lower five bits of the condition code register.

3.6.1 Half Carry Bit (H-Bit)

When the half-carry bit is set, it means that a carry occurred between bits 3 and 4 of the accumulator during the last ADD or ADC (add with carry) operation. The half-carry bit is required for binary-coded decimal (BCD) arithmetic operations.

3.6.2 Interrupt Mask (I-Bit)

When the interrupt mask is set, the internal and external interrupts are disabled. Interrupts are enabled when the interrupt mask is cleared. When an interrupt occurs, the interrupt mask is automatically set after the CPU registers are saved on the stack, but before the interrupt vector is fetched. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. Normally, the interrupt is processed as soon as the interrupt mask is cleared.

A return from interrupt (RTI) instruction pulls the CPU registers from the stack, restoring the interrupt mask to its state before the interrupt was encountered. After any reset, the interrupt mask is set and can only be cleared by the Clear I-Bit (CLI), or WAIT instructions.

3.6.3 Negative Bit (N-Bit)

The negative bit is set when the result of the last arithmetic operation, logical operation, or data manipulation was negative. (Bit 7 of the result was a logical one.)

The negative bit can also be used to check an often tested flag by assigning the flag to bit 7 of a register or memory location. Loading the accumulator with the contents of that register or location then sets or clears the negative bit according to the state of the flag.

3.6.4 Zero Bit (Z-Bit)

The zero bit is set when the result of the last arithmetic operation, logical operation, data manipulation, or data load operation was zero.

3.6.5 Carry/Borrow Bit (C-Bit)

The carry/borrow bit is set when a carry out of bit 7 of the accumulator occurred during the last arithmetic operation, logical operation, or data manipulation. The carry/borrow bit is also set or cleared during bit test and branch instructions and during shifts and rotates. This bit is neither set by an INC nor by a DEC instruction.

SECTION 4
INTERRUPTS

The MCU can be interrupted in seven different ways:

- Non-maskable Software Interrupt Instruction (SWI)
- External Asynchronous Interrupt ($\overline{IRQ}$)
- External Asynchronous Interrupt ($\overline{IRQ2}$)
- Optional External Interrupt via $\overline{IRQ}$ on PA0-PA3 (by a mask option)
- USB Interrupt
- Timer1 Interrupt (16-bit Timer)
- Multi-Function Timer Interrupt

4.1 INTERRUPT VECTORS

Table 4-1. Reset/Interrupt Vector Addresses

Function	Source	Control Bit	Global Hardware Mask	Local Software Mask	Priority (1 = Highest)	Vector Address
Reset	Power-On Logic $\overline{RESET}$ Pin Low Voltage Reset Illegal Address Reset	—	—	—	1	$1FFE–$1FFF
	COP Watchdog					
Software Interrupt (SWI)	User Code	—	—	—	Same Priority As Instruction	$1FFC–$1FFD
External Interrupt (IRQ)	$\overline{IRQ}$ Pin $\overline{IRQ2}$ Pin	—	I Bit	IRQE Bit IRQ2E Bit	2	$1FFA–$1FFB
USB Interrupts	TXD0F TXD1F RESUMP	—	I Bit	TXD0IE TXD1IE —	3	$1FF8–$1FF9
Timer1 Interrupts	ICF Bit OCF Bit TOF Bit	—	I Bit	ICIE Bit OCIE Bit TOIE Bit	4	$1FF6–$1FF7
MFT Interrupts	TOF Bit RTIF	—	I Bit	TOFE Bit RTIE Bit	5	$1FF4–$1FF5
Reserved						$1FF2–$1FF3
Reserved						$1FF0–$1FF1

NOTE

If more than one interrupt request is pending, the CPU fetches the vector of the higher priority interrupt first. A higher priority interrupt does not actually interrupt a lower priority interrupt service routine unless the lower priority interrupt service routine clears the I bit.

4.2 INTERRUPT PROCESSING

The CPU does the following actions to begin servicing an interrupt:

- Stores the CPU registers on the stack in the order shown in **Figure 4-1**.
- Sets the I bit in the condition code register to prevent further interrupts.
- Loads the program counter with the contents of the appropriate interrupt vector locations as shown in **Table 4-1**.

The return from interrupt (RTI) instruction causes the CPU to recover its register contents from the stack as shown in **Figure 4-1**. The sequence of events caused by an interrupt are shown in the flow chart in **Figure 4-2**.

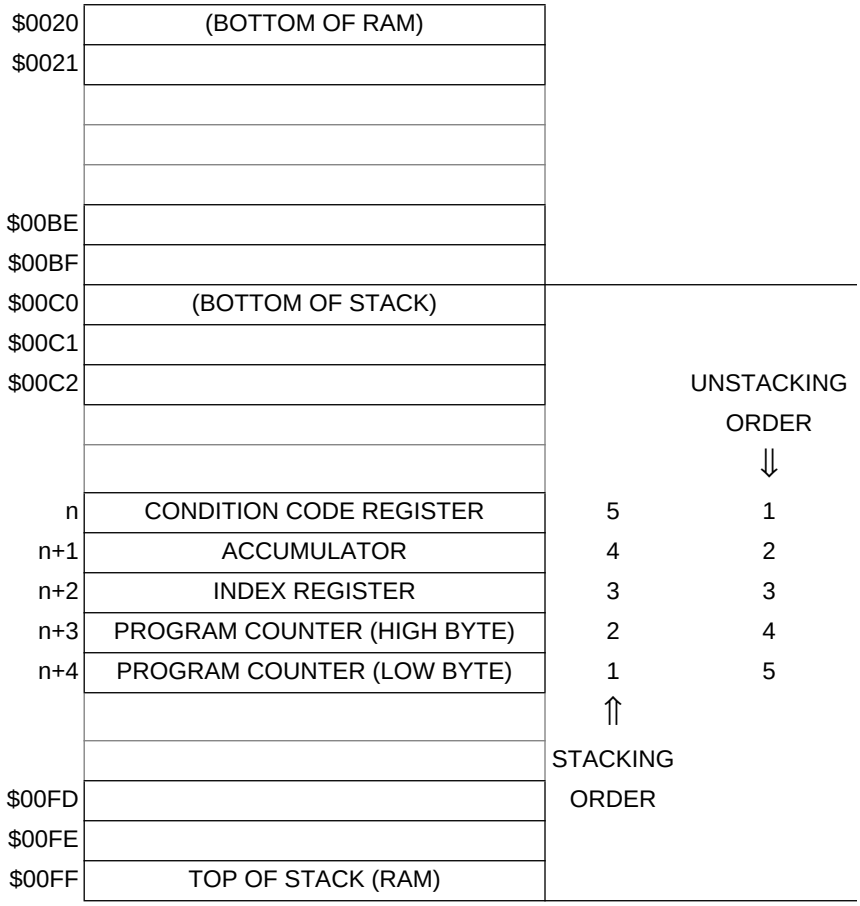


Figure 4-1. Interrupt Stacking Order

I

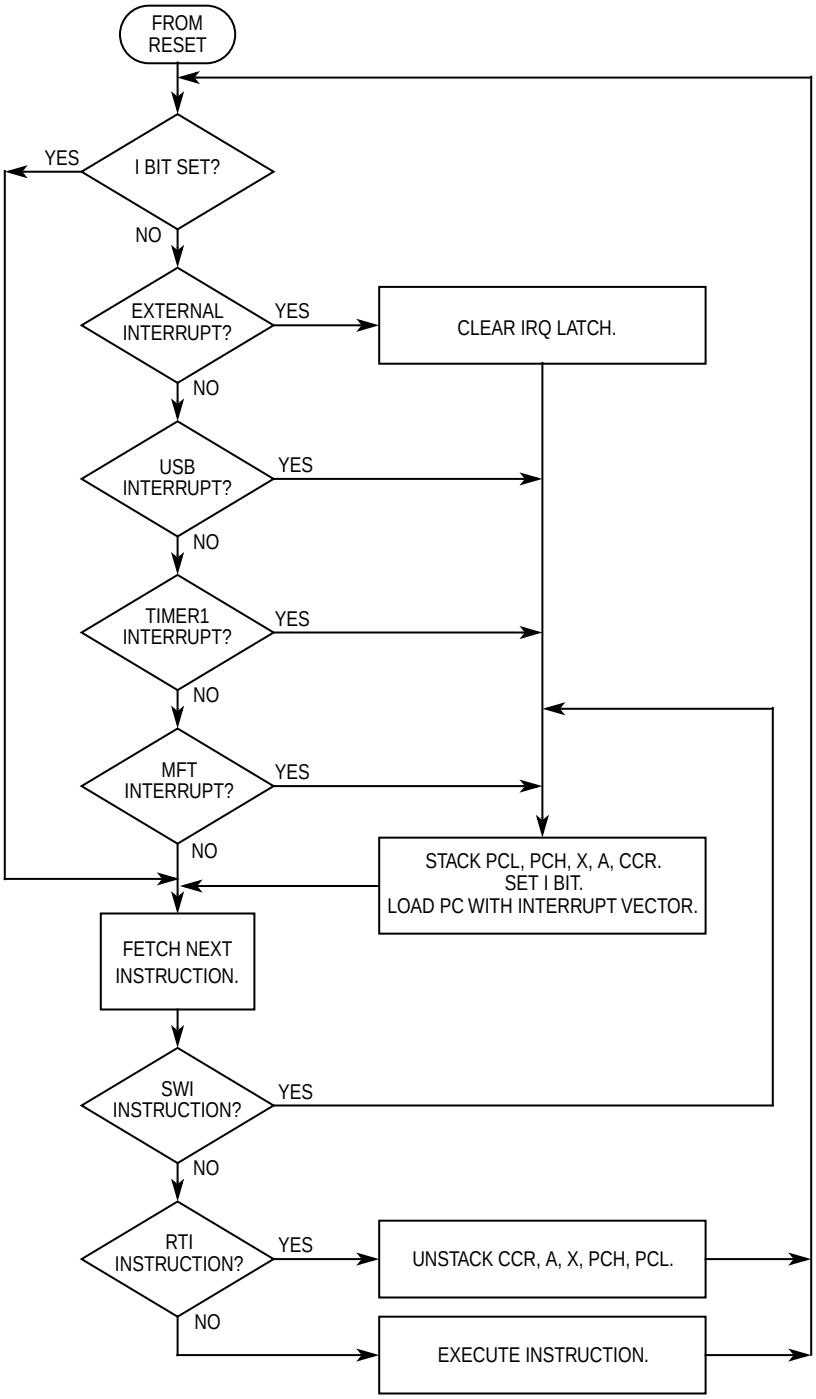


Figure 4-2. Interrupt Flowchart

4.3 RESET INTERRUPT SEQUENCE

The RESET function is not in the strictest sense an interrupt; however, it is acted upon in a similar manner as shown in **Figure 4-2**. A low level input on the $\overline{\text{RESET}}$ pin or an internally generated RST signal causes the program to vector to its starting address which is specified by the contents of memory locations $1FFE and $1FFF. The I-bit in the condition code register is also set.

4.4 SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction and a non-maskable interrupt since it is executed regardless of the state of the I-bit in the CCR. As with any instruction, interrupts pending during the previous instruction will be serviced before the SWI opcode is fetched. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD.

4.5 HARDWARE INTERRUPTS

All hardware interrupts except RESET are maskable by the I-bit in the CCR. If the I-bit is set, all hardware interrupts (internal and external) are disabled. Clearing the I-bit enables the hardware interrupts. There are two types of hardware interrupts which are explained in the following sections.

4.5.1 External Interrupt $\overline{\text{IRQ}}$

The $\overline{\text{IRQ}}$ pin provides an asynchronous interrupt to the CPU. A block diagram of the IRQ logic is shown in **Figure 4-3**.

The $\overline{\text{IRQ}}$ pin is one source of an IRQ interrupt and a mask option can also enable the four lower Port A pins (PA0 to PA3) to act as other IRQ interrupt sources.

Refer to **Figure 4-3** for the following descriptions. IRQ interrupt source comes from IRQ latch. The IRQ latch will be set on the falling edge of the $\overline{\text{IRQ}}$ pin or on any falling edge of PA0-3 pins if PA0-3 interrupts have been enabled. If ‘edge-only’ sensitivity is chosen by a mask option, only the IRQ latch output can activate an IRQF flag which creates a request to the CPU to generate the IRQ interrupt sequence. This makes the IRQ interrupt sensitive to the following cases:

1. Falling edge on the $\overline{\text{IRQ}}$ pin.
2. Falling edge on any PA0-PA3 pin with IRQ enabled (via mask option).

If level sensitivity is chosen, the active high state of the signal to the clock input of the IRQ latch can also activate an IRQF flag which creates an IRQ request to the CPU to generate the IRQ interrupt sequence. This makes the IRQ interrupt sensitive to the following cases:

1. Low level on the $\overline{\text{IRQ}}$ pin.
2. Falling edge on the $\overline{\text{IRQ}}$ pin.
3. Low level on any PA0- PA3 pin with IRQ enabled (via mask option).
4. Falling edge on any PA0- PA3 pin with IRQ enabled (via mask option).

The IRQE enable bit controls whether an active IRQF flag can generate an IRQ interrupt sequence. This interrupt is serviced by the interrupt service routine located at the address specified by the contents of $1FFA and $1FFB.

If IRQF is set, the only way to clear this flag is by writing a logic one to the IRQR acknowledge bit in the ICSR. As long as the output state of the IRQF flag bit is active the CPU will continuously re-enter the IRQ interrupt sequence until the active state is removed or the IRQE enable bit is cleared.

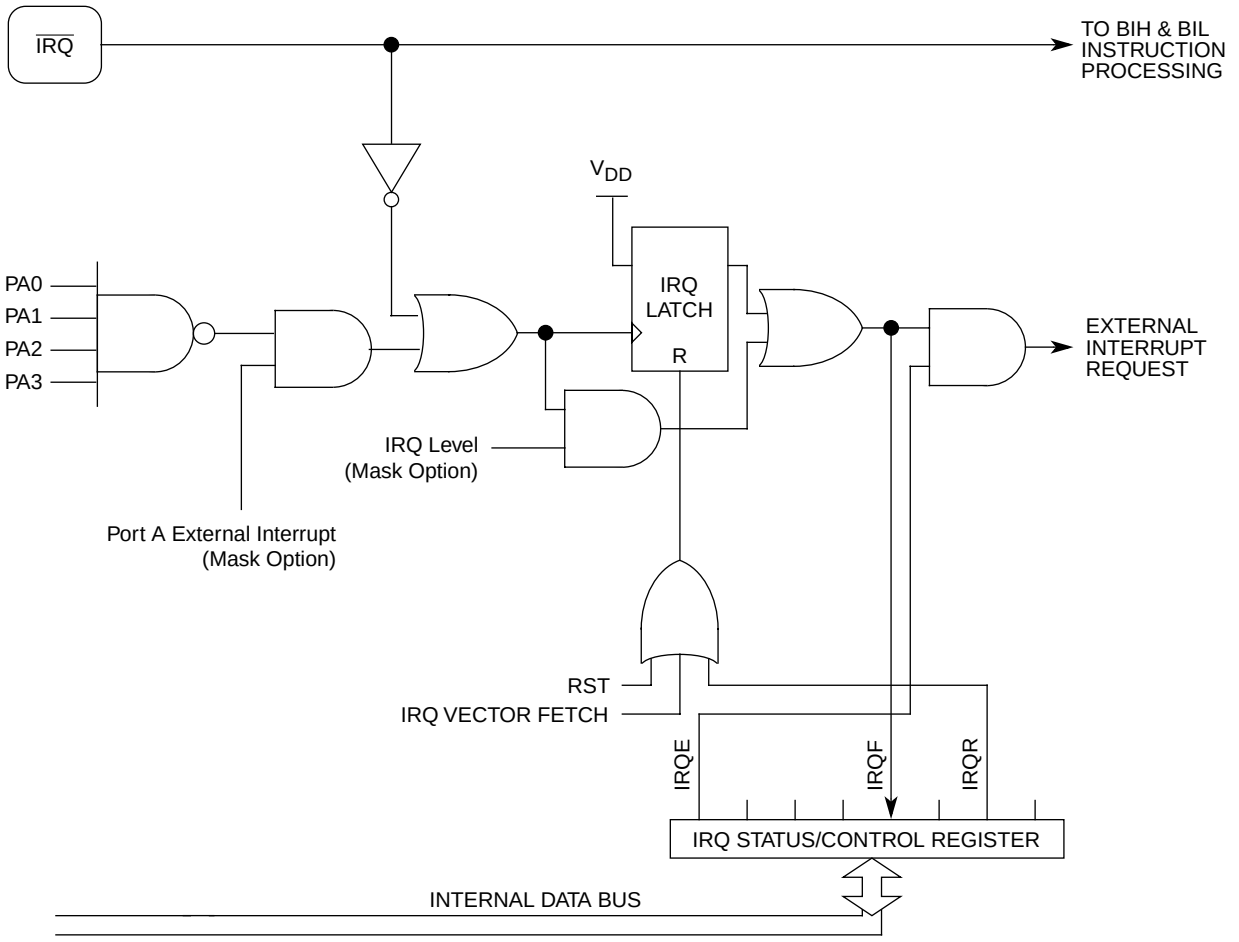


Figure 4-3. External Interrupt (IRQ) Logic

4.5.2 External Interrupt $\overline{\text{IRQ2}}$

The $\overline{\text{IRQ2}}$/PA4 pin provides an asynchronous interrupt to the CPU. When a negative-edge is detected by the schmitt trigger input, an IRQ2 interrupt will be generated if the IRQ2E-bit in the ICSR register is set. This interrupt is serviced by the interrupt service routine located at the address specified by the contents of $1FFA and $1FFB. A block diagram of the IRQ2 function is shown in **Figure 4-4**.

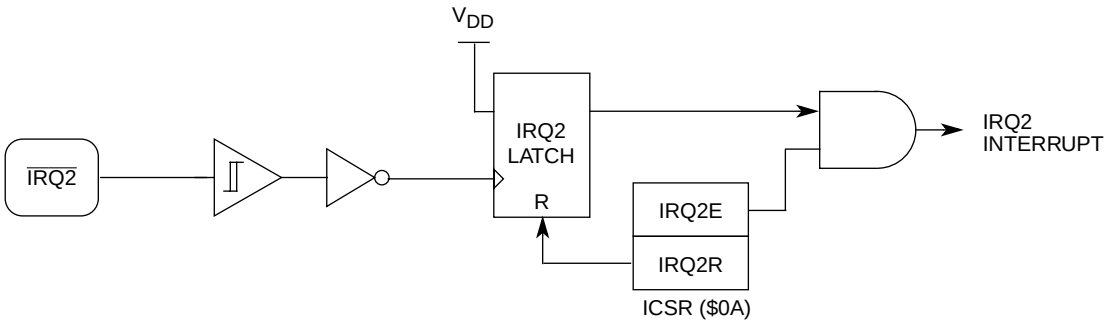


Figure 4-4. External Interrupt (IRQ2) Logic

4.5.3 IRQ Control/Status Register (ICSR) - $0A

The IRQ interrupt function is controlled by the ICSR located at $000A. All unused bits in the ICSR will read as logic zeros. The IRQF bit is cleared and IRQE bit is set by reset.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ICSR	R	IRQE	IRQ2E	0	0	IRQF	IRQ2F	0	0
$000A	W							IRQR	IRQ2R
reset:		1	0	0	0	0	0	0	0

Figure 4-5. IRQ Control and Status Register (ICSR)

IRQR — IRQ Interrupt Acknowledge

The IRQR acknowledge bit clears an IRQ interrupt by clearing the IRQ latch. The IRQR acknowledge bit will always read as a logic zero.

1 = Writing a logic one to the IRQR acknowledge bit will clear the IRQ latch.

0 = Writing a logic zero to the IRQR acknowledge bit will have no effect on the IRQ latch.

IRQF — IRQ Interrupt Request Flag

Writing to the IRQF flag bit will have no effect on it. If the additional setting of IRQF flag bit is not cleared in the IRQ service routine and the IRQE enable bit remains set the CPU will re-enter the IRQ interrupt sequence continuously until either the IRQF flag bit or the IRQE enable bit is clear. The IRQF latch is cleared by reset.

1 = Indicates that an IRQ request is pending.

0 = Indicates that no IRQ request triggered by pins PA0-3 or $\overline{IRQ}$ is pending. The IRQF flag bit can be cleared by writing a logic one to the IRQR acknowledge bit to clear the IRQ latch and also conditioning the external IRQ sources to be inactive (if the level sensitive interrupts are enabled via mask option). Doing so before exiting the service routine will mask out additional occurrences of the IRQF.

IRQE — IRQ Interrupt Enable

The IRQE bit enables/disables the IRQF flag bit to initiate an IRQ interrupt sequence.

1 = Enables $\overline{IRQ}$ interrupt, that is, the IRQF flag bit can generate an interrupt sequence. Reset sets the IRQE enable bit, thereby enabling IRQ interrupts once the I-bit is cleared. Execution of the STOP or WAIT instructions causes the IRQE bit to be set in order to allow the external IRQ to exit these modes.

0 = The IRQF flag bit cannot generate an interrupt sequence.

IRQ2R — IRQ2 Interrupt Acknowledge

The IRQ2R acknowledge bit clears an IRQ2 interrupt by clearing the IRQ2 latch. The IRQ2R acknowledge bit will always read as a logic zero.

1 = Writing a logic one to the IRQ2R acknowledge bit will clear the IRQ2 latch.

0 = Writing a logic zero to the IRQ2R acknowledge bit will have no effect on the IRQ2 latch.

IRQ2F — IRQ2 Interrupt Request Flag

Writing to the IRQ2F flag bit will have no effect on it. If the additional setting of IRQ2F flag bit is not cleared in the IRQ2 service routine and the IRQ2E enable bit remains set the CPU will re-enter the IRQ2 interrupt sequence continuously until either the IRQ2F flag bit or the IRQ2E enable bit is clear. The IRQ2F latch is cleared by reset.

1 = Indicates that an IRQ2 request is pending.

0 = Indicates that no IRQ2 request triggered by pins PA4. The IRQ2F flag bit can be cleared by writing a logic one to the IRQ2R acknowledge bit to clear the IRQ2 latch.

IRQ2E - IRQ2 Interrupt Enable

The IRQ2E bit enables/disables the IRQ2F flag bit to initiate an IRQ2 interrupt sequence.

1 = Enables $\overline{IRQ2}$ interrupt, that is, the IRQ2F flag bit can generate an interrupt sequence. Reset clears the IRQ2E enable bit.

0 = The IRQ2F flag bit cannot generate an interrupt sequence.

4.5.4 Port A External Interrupts (PA0-PA3, by mask option)

The IRQ interrupt can also be triggered by the inputs on the PA0 to PA3 port pins if enabled by a single mask option. If enabled, the lower four bits of Port A can activate the IRQ interrupt function, and the interrupt operation will be the same as for inputs to the $\overline{IRQ}$ pin. This mask option of PA0-3 interrupt allow all of these input pins to be OR'ed with the input present on the $\overline{IRQ}$ pin. All PA0 to PA3 pins must be selected as a group as an additional IRQ interrupt. All the PA0-3 interrupt sources are also controlled by the IRQE enable bit.

NOTE

The BIH and BIL instructions will only apply to the level on the $\overline{IRQ}$ pin itself, and not to the output of the logic OR function with the PA0 to PA3 pins. The state of the individual Port A pins can be checked by reading the appropriate Port A pins as inputs.

NOTE

If enabled, the PA0 to PA3 pins will cause an IRQ interrupt only when the corresponding pin is configured as input.

4.5.5 Timer1 Interrupt (TIMER1)

The TIMER1 interrupt is generated by the 16-bit timer when either an overflow or an input capture or output compare has occurred as described in the section on 16-bit timer. The interrupt flags and enable bits for the Timer1 interrupts are located in the Timer1 Control & Status Register (TSR) located at $0012, $0013. The I-bit in the CCR must be clear in order for the TIMER1 interrupt to be enabled. Either of these three interrupts will vector to the same interrupt service routine located at the address specified by the contents of memory locations $1FF6 and $1FF7.

4.5.6 USB Interrupt (USB)

The USB interrupt is generated by the USB module as described in the section on Universal Serial Bus. The interrupt enable bits for the USB interrupt are located at bit3-bit2 of UIR0 REG and bit3-bit2 of UIR1 REG. Also Once the device goes into Suspend Mode, any bus activities will cause the USB to generate an interrupt to CPU to come out from the Suspend mode. The I-bit in the CCR must be clear in order for the USB interrupt to be enabled. Either of these two interrupts will vector to the same interrupt service routine located at the address specified by the contents of memory locations $1FF8 and $1FF9.

4.5.7 MFT Interrupt (MFT)

The MFT interrupt is generated by the MFT module as described in the section on Multi-function Timer. These interrupts will vector to the same interrupt service routine located at the address specified by the contents of memory locations $1FF4 and $1FF5.

SECTION 5
RESETS

This section describes the six reset sources and how they initialize the MCU. A reset immediately stops the operation of the instruction being executed, initializes certain control bits, and loads the program counter with a user defined reset vector address. The following conditions produce a reset:

- Initial power up of device (power on reset).
- A logic zero applied to the $\overline{\text{RESET}}$ pin (external reset).
- Timeout of the COP watchdog (COP reset).
- Low voltage applied to the device (LVR reset).
- Fetch of an opcode from an address not in the memory map (illegal address reset).
- Detection of USB reset signal (USB reset).

Figure 5-1 shows a block diagram of the reset sources and their interaction.

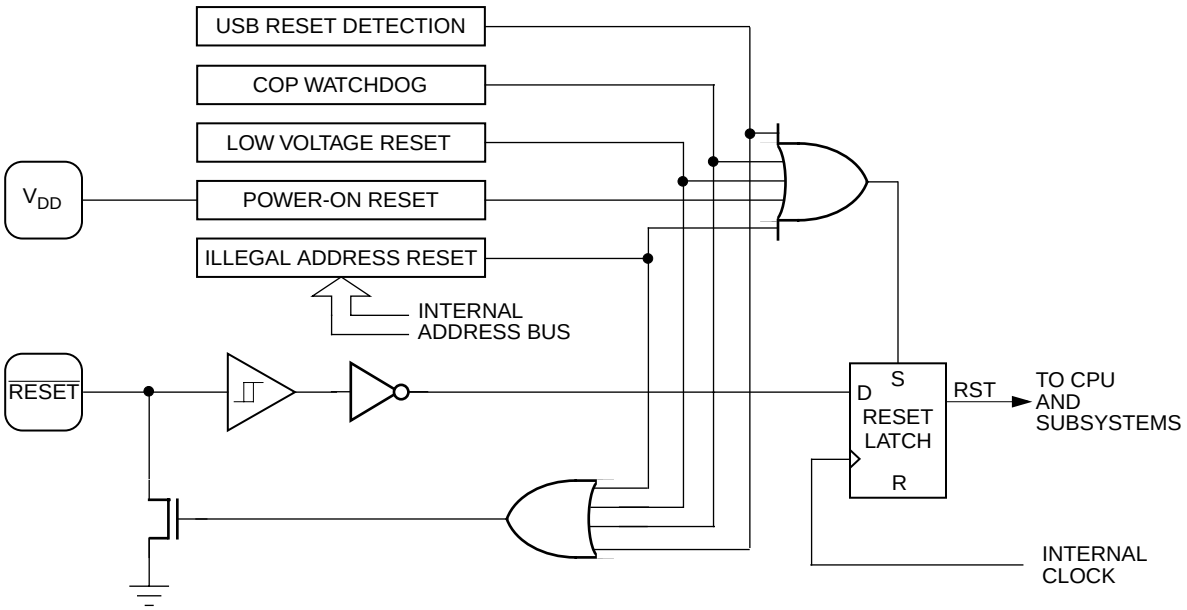


Figure 5-1. Reset Sources

5.1 POWER-ON RESET

A positive transition on the V_{DD} pin generates a power-on reset. The power-on reset is strictly for conditions during powering up and cannot be used to detect drops in power supply voltage.

A 4064 t_{CYC} (internal clock cycle) delay after the oscillator becomes active allows the clock generator to stabilize. If the $\overline{\text{RESET}}$ pin is at logic zero at the end of the multiple t_{CYC} time, the MCU remains in the reset condition until the signal on the $\overline{\text{RESET}}$ pin goes to a logic one.

5.2 EXTERNAL RESET

A logic zero applied to the $\overline{\text{RESET}}$ pin for $1.5t_{CYC}$ generates an external reset. This pin is connected to a Schmitt trigger input gate to provide and upper and lower threshold voltage separated by a minimum amount of hysteresis. The external reset occurs whenever the $\overline{\text{RESET}}$ pin is pulled below the lower threshold and remains in reset until the $\overline{\text{RESET}}$ pin rises above the upper threshold. This active low input will generate the internal RST signal that resets the CPU and peripherals.

The $\overline{\text{RESET}}$ pin can also act as an open drain output. It will be pulled to a low state by an internal pulldown device that is activated by three internal reset sources. This RESET pulldown device will only be asserted for 3 - 4 cycles of the internal clock, f_{OP}, or as long as the internal reset source is asserted. When the external $\overline{\text{RESET}}$ pin is asserted, the pulldown device will not be turned on.

NOTE

Do not connect the $\overline{\text{RESET}}$ pin directly to V_{DD}, as this may overload some power supply designs when the internal pulldown on the $\overline{\text{RESET}}$ pin activates.

5.3 INTERNAL RESETS

The five internally generated resets are the initial power-on reset function, the COP Watchdog timer reset, the low voltage reset, and the illegal address detector. Only the COP Watchdog timer reset, low voltage reset and illegal address detector will also assert the pulldown device on the $\overline{\text{RESET}}$ pin for the duration of the reset function or 3 - 4 internal clock cycles, whichever is longer.

5.3.1 Power-On Reset (POR)

The internal POR is generated on power-up to allow the clock oscillator to stabilize. The POR is strictly for power turn-on conditions and is not able to detect a drop in the power supply voltage (brown-out). There is an oscillator stabilization delay of 4064 internal processor bus clock cycles after the oscillator becomes active.

The POR will generate the RST signal which will reset the CPU. If any other reset function is active at the end of the 4064 cycle delay, the RST signal will remain in the reset condition until the other reset condition(s) end.

POR will not activate the pulldown device on the $\overline{\text{RESET}}$ pin. V_{DD} must drop below V_{POR} in order for the internal POR circuit to detect the next rise of V_{DD}.

5.3.2 USB Reset

The USB reset is generated by a detection on the USB bus reset signal. For MC68HC05JB4, seeing a single-end zero on its upstream port for 4 to 8 bit times will set RSTF bit in UIR0 register. The detections will also generate the RST signal to reset the CPU and other peripherals in the MCU.

5.3.3 Computer Operating Properly (COP) Reset

The COP watchdog is enabled by a mask option.

A timeout of the COP watchdog generates a COP reset. The COP watchdog is part of a software error detection system and must be cleared periodically to start a new timeout period. To clear the COP watchdog and prevent a COP reset, write a logic zero to the COPC bit of the COP register at location $1FF0.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
COPR	R	0	0	0	0	0	0	0	0
$1FF0	W								COPC
reset:		U	U	U	U	U	U	U	0

U = UNAFFECTED BY RESET

Figure 5-2. COP Watchdog Register (COPR)

COPC — COP Clear

COPC is a write-only bit. Periodically writing a logic zero to COPC prevents the COP watchdog from resetting the MCU. Reset clears the COPC bit.

1 = No effect on system.
0 = Reset COP watchdog timer.

The COP Watchdog reset will assert the pulldown device to pull the $\overline{\text{RESET}}$ pin low for one cycle of the internal bus clock.

See section on Core Timer for detail on COP watchdog timeout periods.

5.3.4 Low Voltage Reset (LVR)

The LVR activates the RST reset signal to reset the device when the voltage on the V_{DD} pin falls below the LVR trip voltage. The LVR will assert the pulldown device to pull the $\overline{\text{RESET}}$ pin low one cycle of the internal bus clock. The Low Voltage Reset circuit is enabled by a mask option.

5.3.5 Illegal Address Reset

An opcode fetch from an address that is not in the ROM (locations $1000 to $1FFF) or the RAM (locations $0080 to $012F) generates an illegal address reset. The illegal address reset will assert the pulldown device to pull the $\overline{\text{RESET}}$ pin low for 3 to 4 cycles of the internal bus clock.

SECTION 6
LOW POWER MODES

There are three modes of operation that reduce power consumption:

- Stop mode
- Wait mode
- Data retention mode

Figure 6-1 shows the sequence of events in Stop and Wait modes.

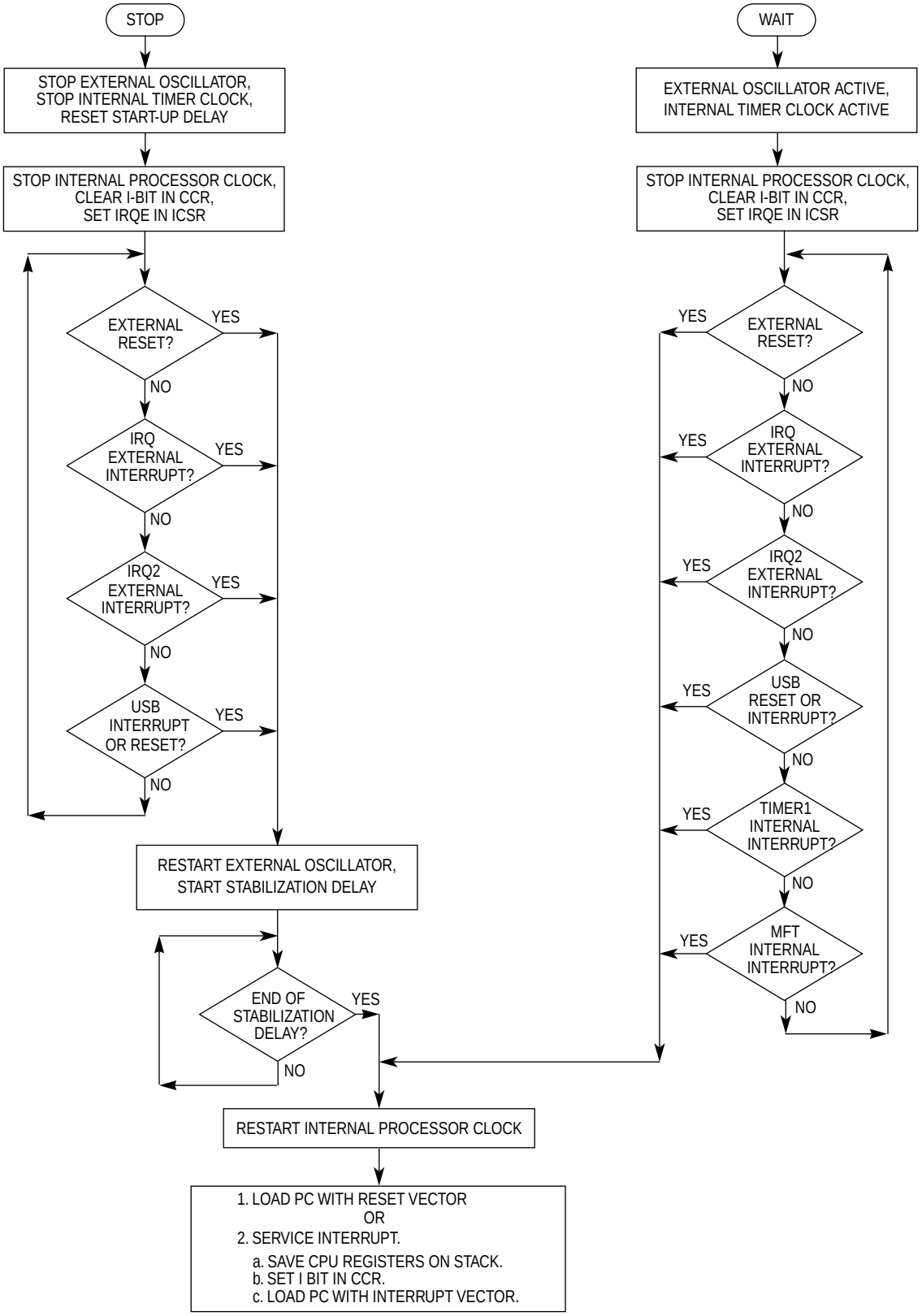


Figure 6-1. STOP and WAIT Flowchart

6.1 STOP MODE

STOP mode is entered by executing the STOP instruction. This is the lowest power consumption mode of the MCU. In the STOP Mode the internal oscillator is turned off, halting all internal processing.

Execution of the STOP instruction automatically clears the I-bit in the Condition Code Register and sets the IRQE enable bit in the IRQ Control/Status Register so that the IRQ external interrupt is enabled. All other registers, including the other bits in the TCSR, and memory remain unaltered. All input/output lines remain unchanged.

The MCU can be brought out of the STOP Mode by an IRQ external interrupt, IRQ2 external interrupt or a USB coming out from Suspend Mode Interrupt (Bus activity detection) or an externally generated RESET, USB Reset or an LVR reset. When exiting the STOP Mode the internal oscillator will resume after a 128 or 4064 internal processor clock cycle oscillator stabilization delay.

6.2 WAIT MODE

WAIT mode is entered by executing the WAIT instruction. This places the MCU in a low-power mode, which consumes more power than the STOP Mode. In the WAIT Mode the internal processor clock is halted, suspending all processor and internal bus activity. Execution of the WAIT instruction automatically clears the I-bit in the Condition Code Register and sets the IRQE enable bit in the IRQ Control/ Status Register so that the IRQ external interrupt is enabled. All other registers, memory, and input/output lines remain in their previous states.

The WAIT Mode may be exited when an external IRQ, IRQ2, USB, Timer1 or MFT interrupt, an LVR reset, USB reset or an external RESET occurs.

6.3 DATA-RETENTION MODE

The Data-Retention mode is only available if the Low Voltage Reset function (mask option) is not enabled.

In the data retention mode, the MCU retains RAM contents and CPU register contents at V_{DD} voltages as low as 2.0Vdc. The data retention feature allows the MCU to remain in a low power consumption state during which it retains data, but the CPU cannot execute instructions. The $\overline{\text{RESET}}$ pin must be held low during data-retention mode.

SECTION 7
INPUT/OUTPUT PORTS

In normal operating mode there are 19 usable bidirectional I/O lines arranged as one 8-bit I/O port (Port-A), one 5-bit I/O port (Port-B), and one 6-bit I/O port (Port C). The individual bits in these ports are programmable as either inputs or outputs under software control by the data direction registers (DDRs).

Each pin on Port-A and Port-B has individual internal pull-up resistor (50kΩ typical) which can be enabled by software. In addition, port pins PA6, PA7, and all port-B pins have built in Slow Falling Edge transition feature. This software selectable feature helps to eliminate EMI noise.

Other functions such as high current drive, interrupt, are available on some port pins via mask option.

7.1 SLOW FALLING-EDGE OUTPUT DRIVER

When enabled, the slow falling-edge output drive feature has a slow falling edge (drops from 5.0V to 2.2V in 167ns typically, with 50pF load) followed by a fast transition to Vss. The fast transition duration is depending on the strength of the output driver defined for each port.

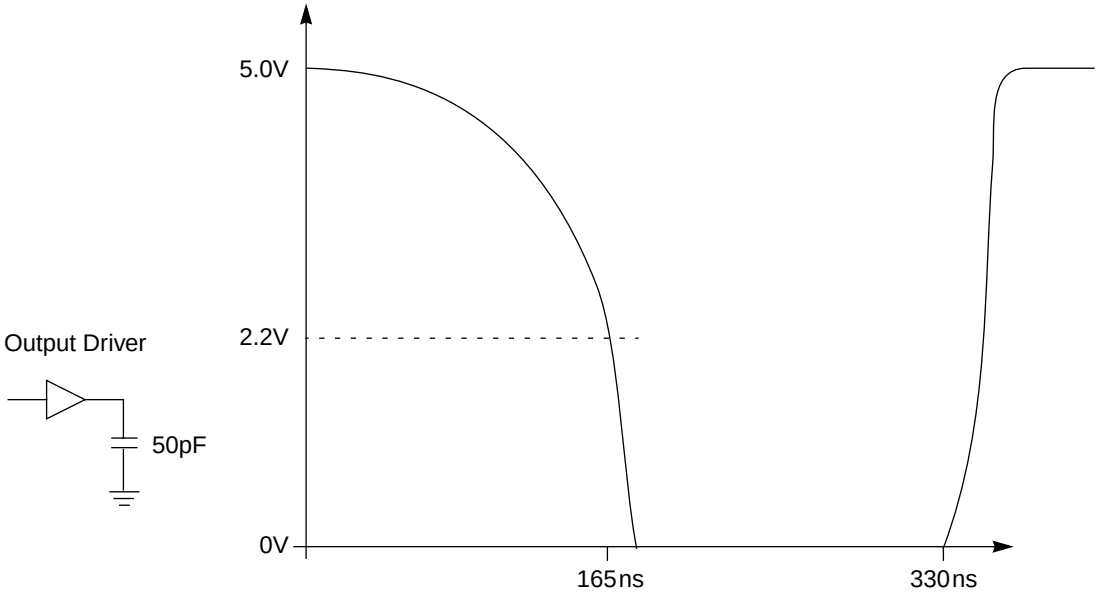


Figure 7-1. Slow Falling-edge Output Driver

7.2 PORT-A

Port-A is an 8-bit bi-directional port. The port-A data register is at $0000 and the data direction register (DDRA) is at $0004. Reset does not affect the data registers, but clears the data direction registers, thereby returning the port pins to inputs. Writing a '1' to a DDR bit sets the corresponding port bit to output mode.

Each pin in Port-A has an internal pull-up resistor (50kΩ typical) which can be individually enabled by writing a '1' to the corresponding bit in the Port-A pull-up control register at location $0010. PA0 to PA4 have built-in schmitt triggered input to improve noise immunity.

PA5 to PA7 port pins each has built in high current drive (10mA sink typical) for direct LED drive. In addition, PA6 and PA7 each has optional 25mA drive which can be enabled by Mask Option. To minimize EME (Electro-Magnetic Emission) noise, PA6 and PA7 has slow output transition which can be enabled by writing a '1' to bit-7 of the Port-B data direction register at location $0005.

PA0 to PA3 and PA4 can be used to generate IRQ and IRQ2 interrupts respectively. PA0 to PA3 and PA4 cannot generate interrupts via IRQ and IRQ2 if these port pins are configured as output pins.

If the pull-up device is enabled and the port pin is configured as output, the output port becomes an open-drain output with 50kΩ pull-up.

Port-A PURX	Port-A DDRX	Pin Configuration
0	0	Input
0	1	Output Push/Pull
1	0	Input with 50k pull-up
1	1	Open-drain Output with 50k pull-up

NOTE

Enabling or disabling the SLOW edge function on PA6 and PA7 does not change the pin configuration. Reading from an output pin will return the content of the data register.

7.2.1 Port-A Data Register

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
$0000	W								
reset:		0	0	0	0	0	0	0	0

7.2.2 Port-A Data Direction Register

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
$0004	W								
reset:		0	0	0	0	0	0	0	0

7.2.3 Port-A Pull-up Control Register

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PURA	R								
$0010	W	PURA7	PURA6	PURA5	PURA4	PURA3	PURA2	PURA1	PURA0
reset:		0	0	0	0	0	0	0	0

7.3 PORT-B

Port-B is a 5-bit bi-directional port. The port-B data register is at $0001 and the data direction register (DDRB) is at $0005. Reset does not affect the data registers, but clears the data direction registers, thereby returning the port pins to inputs. Writing a ‘one’ to a DDR bit sets the corresponding port bit to output mode.

Each pin in Port-B has an internal pull-up resistor (50kΩ typical) which can be individually enabled by writing a ‘1’ to the corresponding bit in the Port-B pull-up control register at location $0011.

All Port-B pins have built in Slow Output edge transition driver which can be enabled by writing a ‘1’ to bit-6 of Port-B data direction register at location $0005.

When PB0 is configured as an input, it also serves as the input capture pin for the 16-bit Timer. When configured as output, the input to the input capture will be permanently tied “low” and no input capture can be generated. PB0 has built-in schmitt triggered input to improve noise immunity.

PB3 and PB4 also serve as extra ADC inputs, AD4 and AD5. When a port pin is selected as ADC input and the ADON bit is set to ‘1’, the pin will be configured as input pin and its pull-up will be disabled automatically regardless of the status of the DDR-bit. The value of the DDR-bit will not be affected.

If the pull-up device is enabled and the port pin is configured as output, the output port becomes an open-drain output with 50kΩ pull-up.

Port-B PURX	Port-B DDRX	Pin Configuration
0	0	Input
0	1	Output Push/Pull
1	0	Input with 50k pull-up
1	1	Open-drain Output with 50k pull-up

NOTE

Enabling or disabling the SLOW edge function does not change the pin configuration. Reading from an output pin will return the content of the data register.

7.3.1 Port-B Data Register

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PTB	R	0	0	0	PB4	PB3	PB2	PB1	PB0
$0001	W								
reset:		0	0	0	0	0	0	0	0

7.3.2 Port-B Data Direction Register

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DDRB	R	SLOWEA	SLOWEB	0	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
$0005	W								
reset:		0	0	0	0	0	0	0	0

SLOWEA

1 = Enable slow falling-edge output transition feature on PA6 and PA7.
0 = Disable slow falling-edge output transition feature on PA6 and PA7.

SLOWEB

1 = Enable slow falling-edge output transition feature on PB0 to PB4.
0 = Disable slow falling-edge output transition feature on PB0 to PB4.

7.3.3 Port-B Pull-up Control Register

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PURB	R								
$0011	W				PURB4	PURB3	PURB2	PURB1	PURB0
reset:		0	0	0	0	0	0	0	0

7.4 PORT-C

Port-C is a 6-bit bi-directional port. The port-C data register is at $0002 and the data direction register (DDRC) is at $0006. Reset does not affect the data registers, but clears the data direction registers, thereby returning the port pins to inputs. Writing a ‘one’ to a DDR bit sets the corresponding port bit to output mode.

When the ADON-bit is set, PC4 and PC5 are used as dedicated ADC reference input, reference high (VRH) and reference low (VRL) respectively. And PC0 to PC3 can be used as ADC inputs AD0 to AD3 when the appropriate channel is

selected. When a port pin is selected as ADC input and the ADON bit is set to ‘1’, the pin will be configured as input pin automatically regardless of the status of the DDR-bit. The value of the DDR-bit will not be affected.

Port-C DDRX	Pin Configuration
0	Input
1	Output Push/Pull

NOTE

Reading from an output pin will return the content of the data register.

7.4.1 Port-C Data Register

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PTC	R	0	0	PC5	PC4	PC3	PC2	PC1	PC0
$0002	W								
reset:		0	0	0	0	0	0	0	0

7.4.2 Port-C Data Direction Register

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DDRC	R	0	0	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
$0006	W								
reset:		0	0	0	0	0	0	0	0

SECTION 8
MULTI-FUNCTION TIMER

The Multi-Function Timer module is a 15-stage ripple counter with Timer Over Flow (TOF), Real Time Interrupt (RTI), and COP Watchdog function.

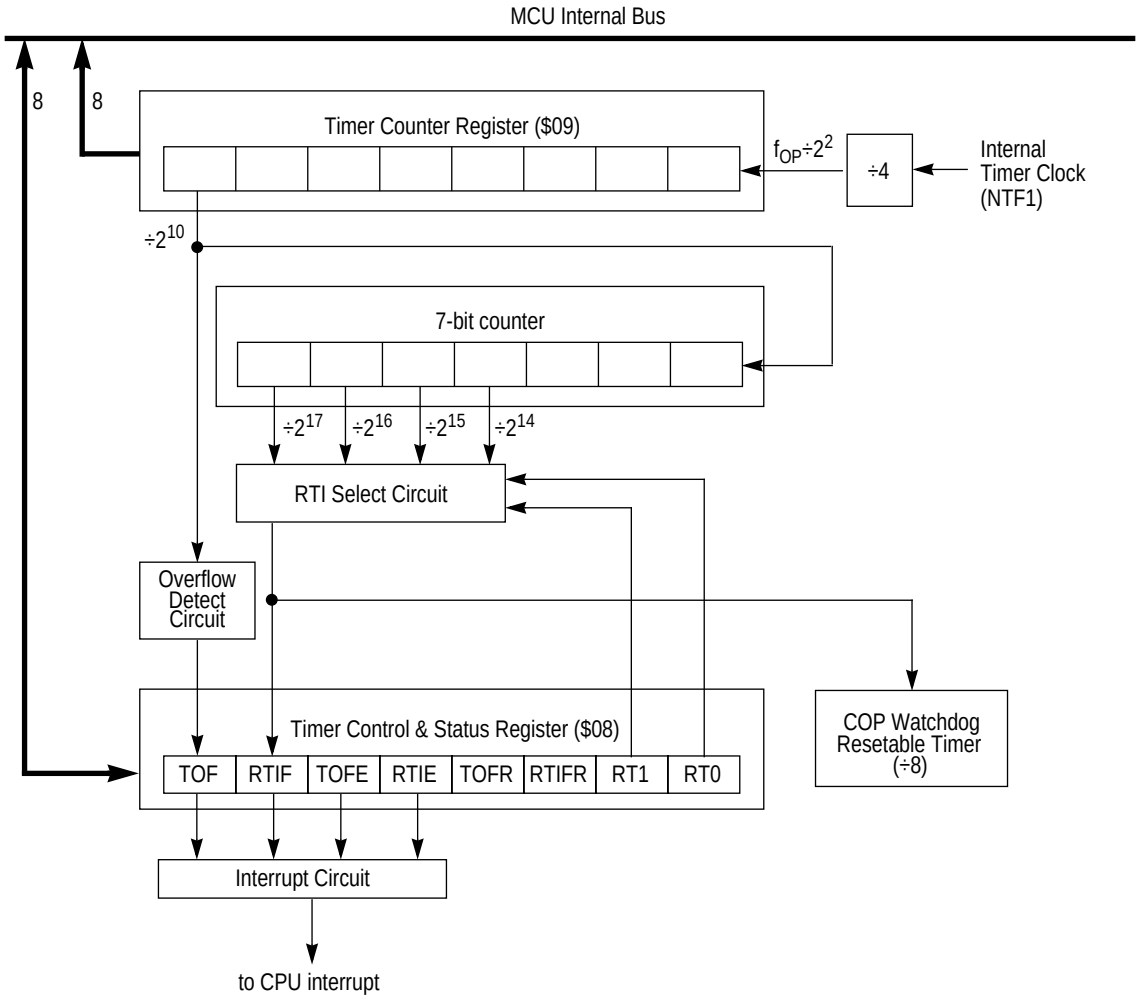


Figure 8-1. Multi-Function Timer Block Diagram

8.1 OVERVIEW

As shown in **Figure 8-1**, the Timer is driven by the timer clock, NTF1, divided by four. NTF1 has the same phase and frequency as the processor bus clock, PH2, but continues to run in WAIT mode. The NTF1 drives an 8-bit ripple counter. The value of this 8-bit ripple counter can be read by the CPU at any time by accessing the Timer Counter Register (TCNT) at address $09. A timer overflow function is implemented on the last stage of this 8-bit counter, giving a possible interrupt rate of $f_{OP} \div 1024$.

The last stage of the 8-bit counter also drives a further 7-bit counter. The final four stages is used by the RTI circuit, giving possible RTI rates of $f_{OP} \div 2^{14}$, 2^{15}, 2^{16} or 2^{17}, selected by RT1 and RT0 (see **Table 8-1**). The RTI rate selector bits, and the RTI and TOF enable bits and flags are located in the Timer Control and Status Register at location $08.

The power-on cycle clears the entire counter chain and begins clocking the counter. After 128 or 4064 cycles, the power-on reset circuit is released which again clears the counter chain and allows the device to come out of reset. At this point, if $\overline{\text{RESET}}$ is not asserted, the timer will start counting up from zero and normal device operation will begin. If $\overline{\text{RESET}}$ is asserted at any time during operation the counter chain will be cleared.

8.2 COMPUTER OPERATING PROPERLY (COP) WATCHDOG

The COP Watchdog is enabled by a mask option.

The COP Watchdog Timer function is implemented by using the output of the RTI circuit and further dividing it by eight. The minimum COP reset rates are listed in **Table 8-1**. If the COP circuit times out, an internal reset is generated and the normal reset vector is fetched.

Preventing a COP time-out is done by writing a “0” to bit-0 of address $1FF0. When the COP is cleared, only the final divide by eight stage (output of the RTI) is cleared.

Table 8-1. RTI and COP Rates at f_{OP}=3.0 MHz

		Bus Frequency, $f_{BUS}=f_{OP}$=3.0 MHz		
RT1	RT0	Divide Ratio	RTI Rate	COP Reset Period (RTI x 8)
0	0	2^{14}	5.46 ms	43.68 ms
0	1	2^{15}	10.92 ms	87.36 ms
1	0	2^{16}	21.85 ms	174.8 ms
1	1	2^{17}	43.69 ms	349.52 ms

8.3 MFT REGISTERS

8.3.1 Timer Counter Register (TCNT) $09

The Timer Counter Register is a read-only register which contains the current value of the 8-bit ripple counter at the beginning of the timer chain. This counter is clocked at $f_{OP} \div 4$ and can be used for various functions including a software input capture. Extended time periods can be attained using the TOF function to increment a temporary RAM storage location thereby simulating a 16-bit (or more) counter. The value of each bit of the TCNT is shown in **Figure 8-2**. This register is cleared by reset.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TCNT	R	TMR7	TMR6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0
$0009	W								
reset:		0	0	0	0	0	0	0	0

Figure 8-2. Timer Counter Register

8.3.2 Timer Control/Status Register (TCSR) $08

The TCSR contains the timer interrupt flag bits, the timer interrupt enable bits, and the real time interrupt rate select bits. Bit 2 and bit 3 are write-only bits which will read as logical zeros. **Figure 8-3** shows the value of each bit in the TCSR following reset.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TCSR	R	TOF	RTIF	TOFE	RTIE	0	0	RT1	RT0
$0008	W					TOFR	RTIFR		
reset:		0	0	0	0	0	0	1	1

Figure 8-3. Timer Control/Status Register (TCSR)

TOF - Timer Overflow Flag

The TOF is a read-only flag bit.

1 = Set when the 8-bit ripple counter rolls over from $FF to $00. A TIMER Interrupt request will be generated if TOFE is also set.

0 = Reset by writing a logical one to the TOF acknowledge bit, TOFR. Writing to the TOF flag bit has no effect on its value. This bit is cleared by reset.

RTIF - Real Time Interrupt Flag

The RTIF is a read-only flag bit.

1 = Set when the output of the chosen (1 of 4 selections) Real Time Interrupt stage goes active. A TIMER Interrupt request will be generated if RTIE is also set.

0 = Reset by writing a logical one to the RTIF acknowledge bit, RTIFR. Writing to the RTIF flag bit has no effect on its value. This bit is cleared by reset.

TOFE - Timer Overflow Enable

The TOFE is an enable bit that allows generation of a TIMER Interrupt upon overflow of the Timer Counter Register.

1 = When set, the TIMER Interrupt is generated when the TOF flag bit is set.

0 = When cleared, no TIMER interrupt caused by TOF bit set will be generated. This bit is cleared by reset.

RTIE - Real Time Interrupt Enable

The RTIE is an enable bit that allows generation of a TIMER Interrupt by the RTIF bit.

1 = When set, the TIMER Interrupt is generated when the RTIF flag bit is set.

0 = When cleared, no TIMER interrupt caused by RTIF bit set will be generated. This bit is cleared by reset.

TOFR - Timer Overflow Acknowledge

The TOFR is an acknowledge bit that resets the TOF flag bit. This bit is unaffected by reset. Reading the TOFR will always return a logical zero.

1 = Clears the TOF flag bit.

0 = Does not clear the TOF flag bit.

RTIFR - Real Time Interrupt Acknowledge

The RTIFR is an acknowledge bit that resets the RTIF flag bit. This bit is unaffected by reset. Reading the RTIFR will always return a logical zero.

1 = Clears the RTIF flag bit.

0 = Does not clear the RTIF flag bit.

8.4 OPERATION DURING STOP MODE

When STOP is exited by an external interrupt or an LVR reset or an external $\overline{\text{RESET}}$, the internal oscillator will resume, followed by a 128 or 4064 internal processor oscillator stabilization delay.

8.5 COP CONSIDERATION DURING STOP MODE

In STOP mode, the clock to the Watchdog Timer is stopped and is therefore impossible to generate COP reset when in STOP mode. The COP function will resume 128 or 4064 cycles after exiting from STOP.

SECTION 9
16-BIT TIMER

This 16-bit Programmable Timer (Timer1) has an Input Capture function and an Output Compare function. **Figure 9-1** shows a block diagram of the 16-bit programmable timer.

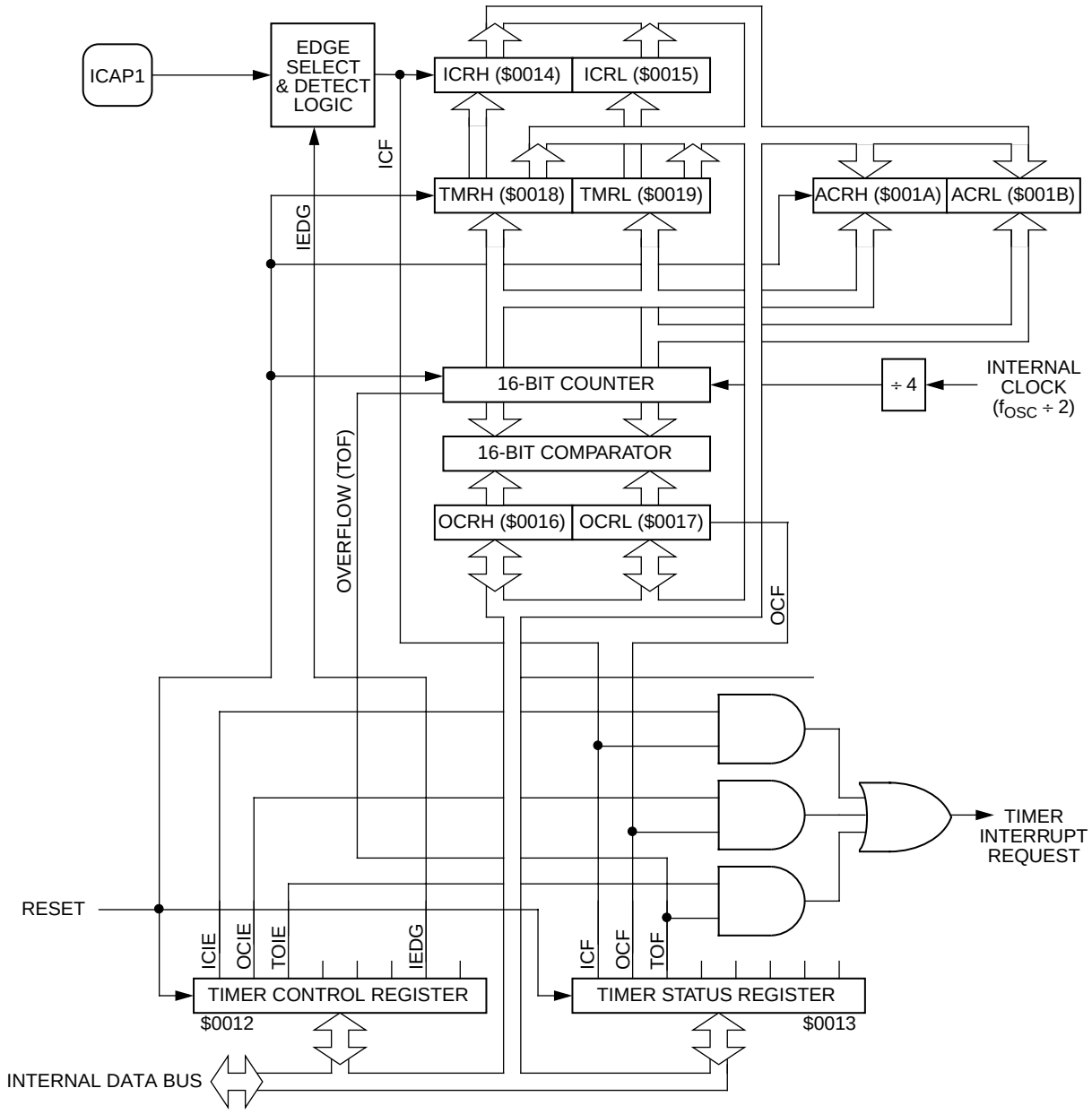


Figure 9-1. Programmable Timer Block Diagram

The basis of the 16-bit Timer is a 16-bit free-running counter which increases in count with each internal bus clock cycle. The counter is the timing reference for the input capture and output compare functions. The input capture and output compare functions provide a means to latch the times at which external events occur, to measure input waveforms, and to generate output waveforms and timing delays. Software can read the value in the 16-bit free-running counter at any time without affect the counter sequence.

Because of the 16-bit timer architecture, the I/O registers for the input capture and output compare functions are pairs of 8-bit registers. Each register pair contains the high and low byte of that function. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

Because the counter is 16 bits long and preceded by a fixed divide-by-four prescaler, the counter rolls over every 262,144 internal clock cycles. Timer resolution with a 4MHz crystal oscillator is 2 microsecond/count.

The interrupt capability, the input capture edge, and the output compare state are controlled by the timer control register (TCR) located at $0012 and the status of the interrupt flags can be read from the timer status register (TSR) located at $0013.

9.1 TIMER REGISTERS (TMRH, TMRL)

The functional block diagram of the 16-bit free-running timer counter and timer registers is shown in **Figure 9-2**. The timer registers include a transparent buffer latch on the LSB of the 16-bit timer counter.

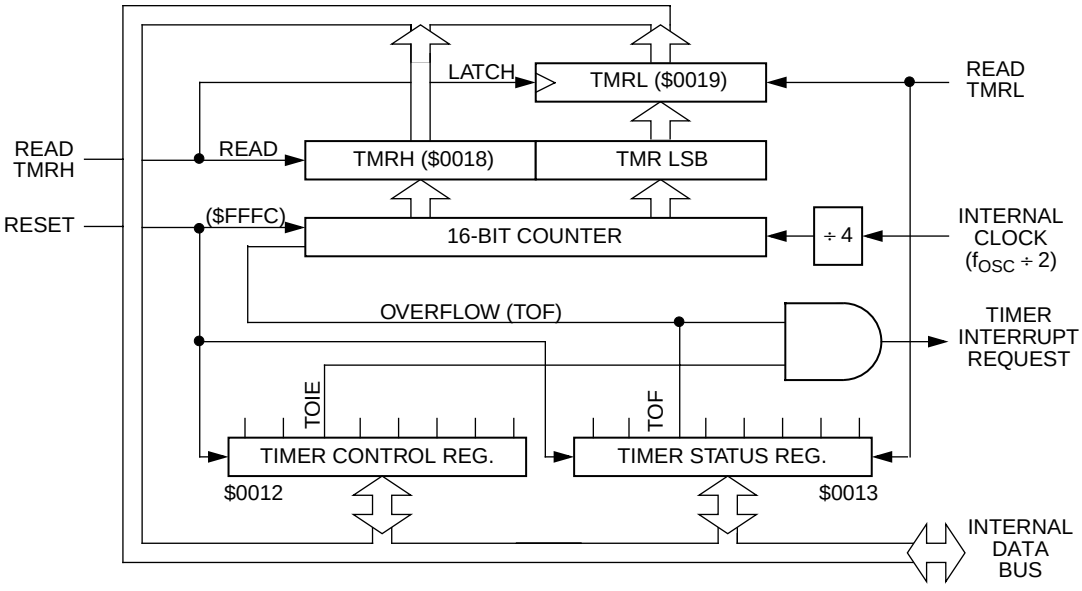


Figure 9-2. Programmable Timer Counter Block Diagram

The timer registers (TMRH, TMRL) shown in **Figure 9-3** are read-only locations which contain the current high and low bytes of the 16-bit free-running counter. Writing to the timer registers has no effect. Reset of the device presets the timer counter to $FFFC.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TMRH	R	TMRH7	TMRH6	TMRH5	TMRH4	TMRH3	TMRH2	TMRH1	TMRH0
$0018	W								
reset:		1	1	1	1	1	1	1	1
TMRL	R	TMRL7	TMRL6	TMRL5	TMRL4	TMRL3	TMRL2	TMRL1	TMRL0
$0019	W								
reset:		1	1	1	1	1	1	0	0

Figure 9-3. Programmable Timer Counter Registers (TMRH, TMRL)

The TMRL latch is a transparent read of the LSB until the a read of the TMRH takes place. A read of the TMRH latches the LSB into the TMRL location until the TMRL is again read. The latched value remains fixed even if multiple reads of the TMRH take place before the next read of the TMRL. Therefore, when reading the MSB of the timer at TMRH the LSB of the timer at TMRL must also be read to complete the read sequence.

During power-on-reset (POR), the counter is initialized to $FFFC and begins counting after the oscillator start-up delay. Because the counter is sixteen bits and preceded by a fixed divide-by-four prescaler, the value in the counter repeats every 262, 144 internal bus clock cycles (524, 288 oscillator cycles).

When the free-running counter rolls over from $FFFF to $0000, the timer overflow flag bit (TOF) is set in the TSR. When the TOF is set, it can generate an interrupt if the timer overflow interrupt enable bit (TOIE) is also set in the TCR. The TOF flag bit can only be reset by reading the TMRL after reading the TSR.

Other than clearing any possible TOF flags, reading the TMRH and TMRL in any order or any number of times does not have any effect on the 16-bit free-running counter.

NOTE

To prevent interrupts from occurring between readings of the TMRH and TMRL, set the I bit in the condition code register (CCR) before reading TMRH and clear the I bit after reading TMRL.

9.2 ALTERNATE COUNTER REGISTERS (ACRH, ACRL)

The functional block diagram of the 16-bit free-running timer counter and alternate counter registers is shown in **Figure 9-4**. The alternate counter registers behave the same as the timer registers, except that any reads of the alternate counter will not have any effect on the TOF flag bit and Timer interrupts. The alternate counter registers include a transparent buffer latch on the LSB of the 16-bit timer counter.

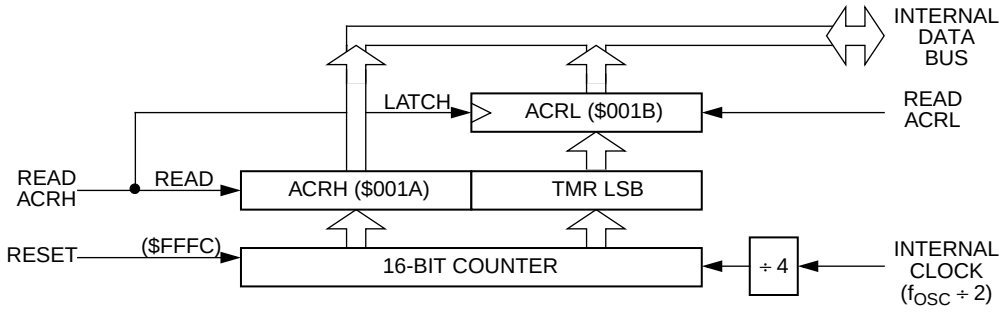


Figure 9-4. Alternate Counter Block Diagram

The alternate counter registers (ACRH, ACRL) shown in **Figure 9-5** are read-only locations which contain the current high and low bytes of the 16-bit free-running counter. Writing to the alternate counter registers has no effect. Reset of the device presets the timer counter to $FFFC.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ACRH	R	ACRH7	ACRH6	ACRH5	ACRH4	ACRH3	ACRH2	ACRH1	ACRH0
$001A	W								
reset:		1	1	1	1	1	1	1	1

ACRL	R	ACRL7	ACRL6	ACRL5	ACRL4	ACRL3	ACRL2	ACRL1	ACRL0
$001B	W								
reset:		1	1	1	1	1	1	0	0

Figure 9-5. Alternate Counter Registers (ACRH, ACRL)

The ACRL latch is a transparent read of the LSB until the a read of the ACRH takes place. A read of the ACRH latches the LSB into the ACRL location until the ACRL is again read. The latched value remains fixed even if multiple reads of the ACRH take place before the next read of the ACRL. Therefore, when reading the MSB of the timer at ACRH the LSB of the timer at ACRL must also be read to complete the read sequence.

During power-on-reset (POR), the counter is initialized to $FFFC and begins counting after the oscillator start-up delay. Because the counter is sixteen bits and preceded by a fixed divide-by-four prescaler, the value in the counter repeats every 262,144 internal bus clock cycles (524,288 oscillator cycles).

Reading the ACRH and ACRL in any order or any number of times does not have any effect on the 16-bit free-running counter or the TOF flag bit.

NOTE

To prevent interrupts from occurring between readings of the ACRH and ACRL, set the I bit in the condition code register (CCR) before reading ACRH and clear the I bit after reading ACRL.

9.3 INPUT CAPTURE REGISTERS

The input capture function is a technique whereby an external signal (connected to PB0/ICAP1 pin) is used to trigger the 16-bit timer counter. In this way it is possible to relate the timing of an external signal to the internal counter value, and hence to elapsed time.

When the input capture circuitry detects an active edge on the ICAP1 pin, it latches the contents of the free-running timer counter registers into the input capture registers as shown in **Figure 9-6**.

Latching values into the input capture registers at successive edges of the same polarity measures the period of the selected input signal. Latching the counter values at successive edges of opposite polarity measures the pulse width of the signal.

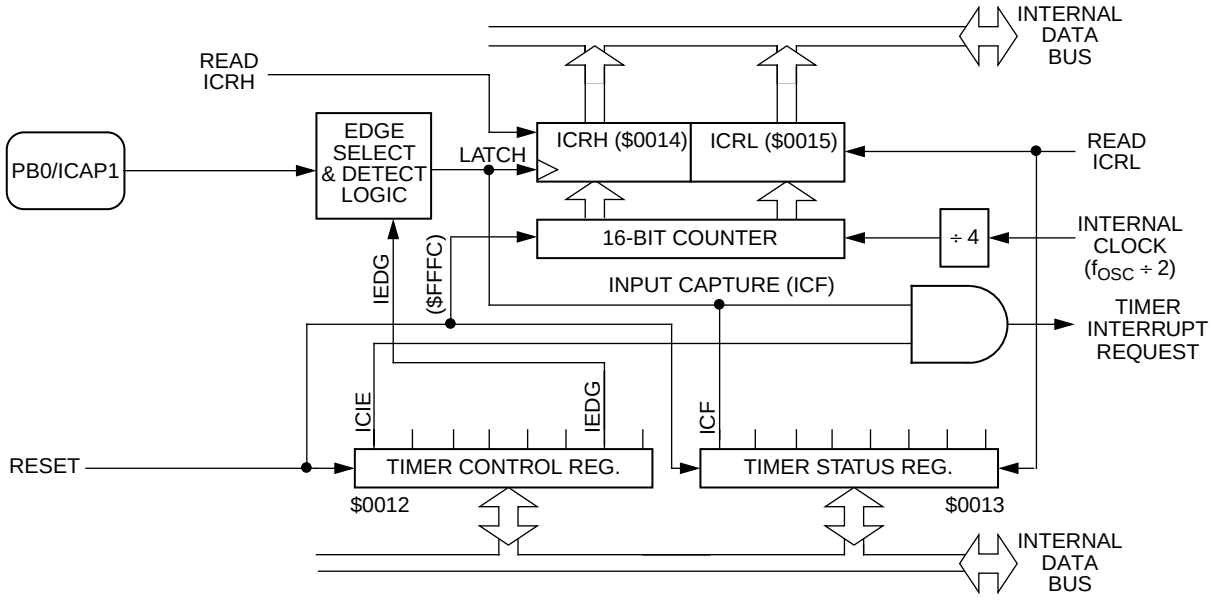


Figure 9-6. Timer Input Capture Block Diagram

The input capture registers are made up of two 8-bit read-only registers (ICRH, ICRL) as shown in **Figure 9-7**. The input capture edge detector contains a Schmitt trigger to improve noise immunity. The edge that triggers the counter transfer is defined by the input edge bit (IEDG) in the TCR. Reset does not affect the contents of the input capture registers.

The result obtained by an input capture will be one count higher than the value of the free-running timer counter preceding the external transition. This delay is required for internal synchronization. Resolution is affected by the prescaler, allowing the free-running timer counter to increment once every four internal clock cycles (eight oscillator clock cycles).

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ICRH	R	ICRH7	ICRH6	ICRH5	ICRH4	ICRH3	ICRH2	ICRH1	ICRH0
$0014	W								
reset:		U	U	U	U	U	U	U	U
ICRL	R	ICRL7	ICRL6	ICRL5	ICRL4	ICRL3	ICRL2	ICRL1	ICRL0
$0015	W								
reset:		U	U	U	U	U	U	U	U

U = UNAFFECTED BY RESET

Figure 9-7. Input Capture Registers (ICRH, ICRL)

Reading the ICRH inhibits further captures until the ICRL is also read. Reading the ICRL after reading the timer status register (TSR) clears the ICF flag bit. does not inhibit transfer of the free-running counter. There is no conflict between reading the ICRL and transfers from the free-running timer counters. The input capture registers always contain the free-running timer counter value which corresponds to the most recent input capture.

NOTE

To prevent interrupts from occurring between readings of the ICRH and ICRL, set the I bit in the condition code register (CCR) before reading ICRH and clear the I bit after reading ICRL.

9.4 OUTPUT COMPARE REGISTERS

The Output Compare function is a means of generating an interrupt when the 16-bit timer counter reaches a selected value as shown in **Figure 9-8**. Software writes the selected value into the output compare registers. On every fourth internal clock cycle (every eight oscillator clock cycle) the output compare circuitry compares the value of the free-running timer counter to the value written in the output compare registers. When a match occurs, the output compare interrupt flag, OCF is set. A timer interrupt request to the CPU is generated if the output compare interrupt enable is set, i.e. OCIE=1.

Software can use the output compare register to measure time periods, to generate timing delays, or to generate a pulse of specific duration or a pulse train of specific frequency and duty cycle.

Writing to the OCRH before writing to the OCRL inhibits timer compares until the OCRL is written. Reading or writing to the OCRL after reading the TSR will clear the output compare flag bit (OCF).

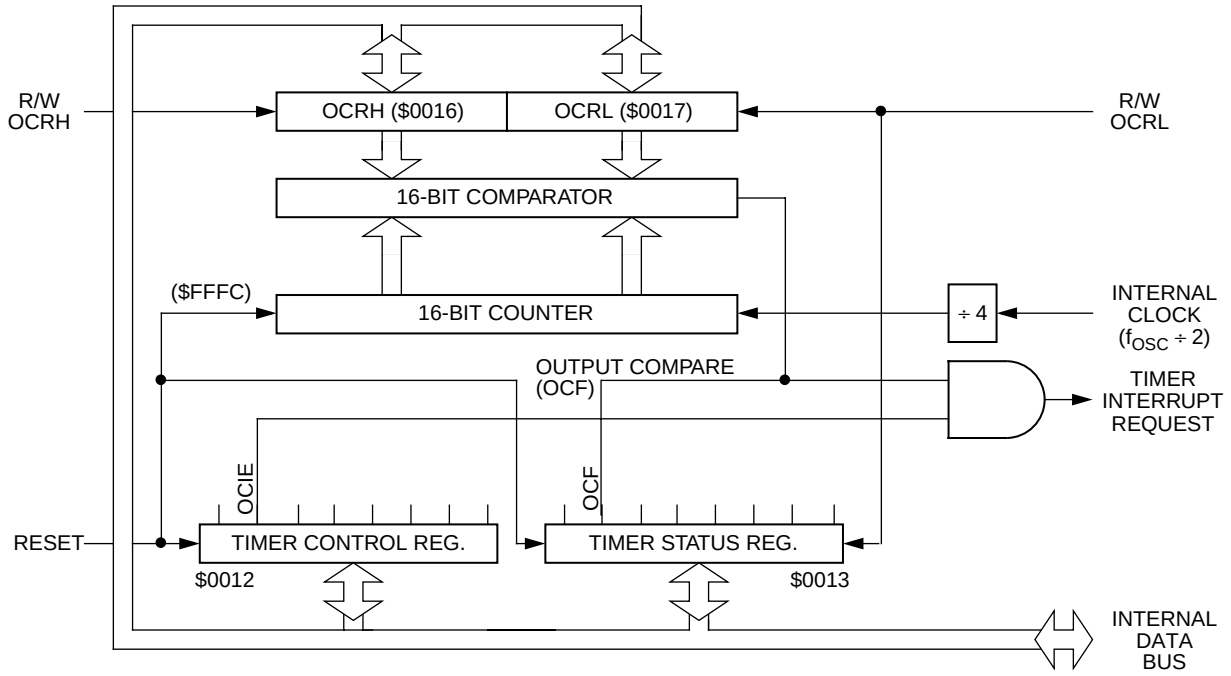


Figure 9-8. Timer Output Compare Block Diagram

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OCRH	R	OCRH7	OCRH6	OCRH5	OCRH4	OCRH3	OCRH2	OCRH1	OCRH0
$0016	W								
reset:		U	U	U	U	U	U	U	U
OCRL	R	OCRL7	OCRL6	OCRL5	OCRL4	OCRL3	OCRL2	OCRL1	OCRL0
$0017	W								
reset:		U	U	U	U	U	U	U	U

U = UNAFFECTED BY RESET

Figure 9-9. Output Compare Registers (OCRH, OCRL)

To prevent OCF from being set between the time it is read and the time the output compare registers are updated, use the following procedure:

1. Disable interrupts by setting the I bit in the condition code register.
2. Write to the OCRH. Compares are now inhibited until OCRL is written.
3. Read the TSR to arm the OCF for clearing.
4. Enable the output compare registers by writing to the OCRL. This also clears the OCF flag bit in the TSR.
5. Enable interrupts by clearing the I bit in the condition code register.

A software example of this procedure is shown below.

```
9B              SEI             DISABLE INTERRUPTS
...             ...             .....
...             ...             .....
B7      16      STA     OCRH    INHIBIT OUTPUT COMPARE
B6      13      LDA     TSR     ARM OCF FLAG FOR CLEARING
BF      17      STX     OCRL    READY FOR NEXT COMPARE, OCF CLEARED
...             ...             .....
...             ...             .....
9A              CLI             ENABLE INTERRUPTS
```

9.5 TIMER CONTROL REGISTER (TCR)

The timer control register is shown in **Figure 9-10** performs the following functions:

- Enables input capture interrupts
- Enables output compare interrupts
- Enables timer overflow interrupts
- Control the active edge polarity of the ICAP1 signal on pin PB0/ICAP1

Reset clears all the bits in the TCR with the exception of the IEDG bit which is unaffected.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TCR	R	ICIE	OCIE	TOIE	0	0	0	IEDG	0
$0012	W								
reset:		0	0	0	0	0	0	Unaffected	0

Figure 9-10. Timer Control Register (TCR)

ICIE - INPUT CAPTURE INTERRUPT ENABLE

This read/write bit enables interrupts caused by an active signal on the PB0/ICAP1 pin. Reset clears the ICIE bit.

1 = Input capture interrupts enabled.
0 = Input capture interrupts disabled.

OCIE - OUTPUT COMPARE INTERRUPT ENABLE

This read/write bit enables interrupts caused by a successful compare between the timer counter and the output compare registers. Reset clears the OCIE bit.

1 = Output compare interrupts enabled.
0 = Output compare interrupts disabled.

TOIE - TIMER OVERFLOW INTERRUPT ENABLE

This read/write bit enables interrupts caused by a timer overflow. Reset clears the TOIE bit.

1 = Timer overflow interrupts enabled.
0 = Timer overflow interrupts disabled.

IEDG - INPUT CAPTURE EDGE SELECT

The state of this read/write bit determines whether a positive or negative transition on the ICAP1 pin triggers a transfer of the contents of the timer register to the input capture register. Reset has no effect on the IEDG bit.

1 = Positive edge (low to high transition) triggers input capture.
0 = Negative edge (high to low transition) triggers input capture.

9.6 TIMER STATUS REGISTER (TSR)

The timer status register (TSR) shown in **Figure 9-11** contains flags for the following events:

- An active signal on the PB0/ICAP1 pin, transferring the contents of the timer registers to the input capture registers.
- A match between the 16-bit counter and the output compare registers
- An overflow of the timer registers from $FFFF to $0000.

Writing to any of the bits in the TSR has no effect. Reset does not change the state of any of the flag bits in the TSR.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TSR	R	ICF	OCF	TOF	0	0	0	0	0
$0013	W								
reset:		U	U	U	0	0	0	0	0

U = UNAFFECTED BY RESET

Figure 9-11. Timer Status Registers (TSR)

ICF - INPUT CAPTURE FLAG

The ICF bit is automatically set when an edge of the selected polarity occurs on the PB0/ICAP1 pin. Clear the ICF bit by reading the timer status register with the ICF set, and then reading the low byte (ICRL, $0015) of the input capture registers. Reset has no effect on ICF.

OCF - OUTPUT COMPARE FLAG

The OCF bit is automatically set when the value of the timer registers matches the contents of the output compare registers. Clear the OCF bit by reading the timer status register with the OCF set, and then accessing the low byte (OCRL, $0017) of the output compare registers. Reset has no effect on OCF.

TOF - TIMER OVERFLOW FLAG

The TOF bit is automatically set when the 16-bit timer counter rolls over from $FFFF to $0000. Clear the TOF bit by reading the timer status register with the TOF set, and then accessing the low byte (TMRL, $0019) of the timer registers. Reset has no effect on TOF.

9.7 TIMER OPERATION DURING WAIT MODE

During WAIT mode the 16-bit timer continues to operate normally and may generate an interrupt to trigger the MCU out of the WAIT mode.

9.8 TIMER OPERATION DURING STOP MODE

When the MCU enters the STOP mode the free-running counter stops counting (the internal processor clock is stopped). It remains at that particular count value until the STOP mode is exited by applying a low signal to the $\overline{IRQ}$ pin, at which time the counter resumes from its stopped value as if nothing had happened. If STOP mode is exited via an external reset (logic low applied to the $\overline{RESET}$ pin) the counter is forced to $FFFC.

If a valid input capture edge occurs at the PB0/TCAP pin during the STOP mode the input capture detect circuitry will be armed. This action does not set any flags or “wake up” the MCU, but when the MCU does “wake up” there will be an active input capture flag (and data) from the first valid edge. If the STOP mode is exited by an external reset, no input capture flag or data will be present even if a valid input capture edge was detected during the STOP mode.

SECTION 10
UNIVERSAL SERIAL BUS MODULE

This USB Module is designed for USB application in LS products. With minimized software effort, it can fully comply with USB LS device specification. See USB specification version 1.0 for the detail description of USB.

10.1 FEATURES

- Integrated 3.3 Volt Regulator with 3.3V Output Pin
- Integrated USB transceiver supporting Low Speed functions
- USB Data Control Logic
  - Packet decoding/generation
  - CRC generation and checking
  - NRZI encoding/decoding
  - Bit-stuffing
- USB reset support
- Control Endpoint 0 and Interrupt Endpoints 1 and 2
- Two 8-byte transmit buffers
- One 8-byte receive buffer
- Suspend and resume operations
- Remote Wake-up support
- USB generated interrupts
- transaction interrupt driven
- Resume interrupt
- End of Packet interrupt
- Stall, Nak, and Ack handshake generation

10.2 OVERVIEW

This section provides an overview of the Universal Serial Bus (USB) module in the MC68HC05JB4. This USB module is designed to serve as a low-speed (LS) USB device per the Universal Serial Bus Specification Rev 1.0. Three types of USB data transfers are supported: control, interrupt, and bulk (transmit only). Endpoint 0 functions as a receive/transmit control endpoint. Endpoints 1 and 2 can function as interrupt or bulk, but only in the transmit direction.

A block diagram of the USB module is shown **Figure 10-1**. The USB module manages communications between the host and the USB function. The module is partitioned into four functional blocks. These blocks consist of a 3.3 volt regulator, a dual function transceiver, the USB control logic, and the endpoint registers. The blocks are further detailed in **Section 10.4**.

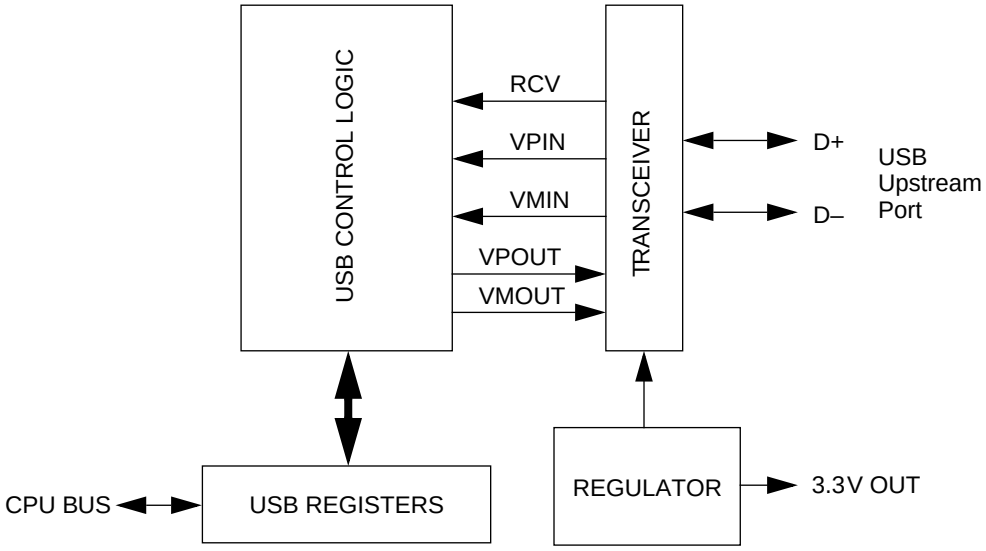


Figure 10-1. USB Block Diagram

10.2.1 USB Protocol

Figure 10-2 shows the various transaction types supported by the MC68HC05JB4 USB module. The transactions are portrayed as error free. The effect of errors in the data flow are discussed later.

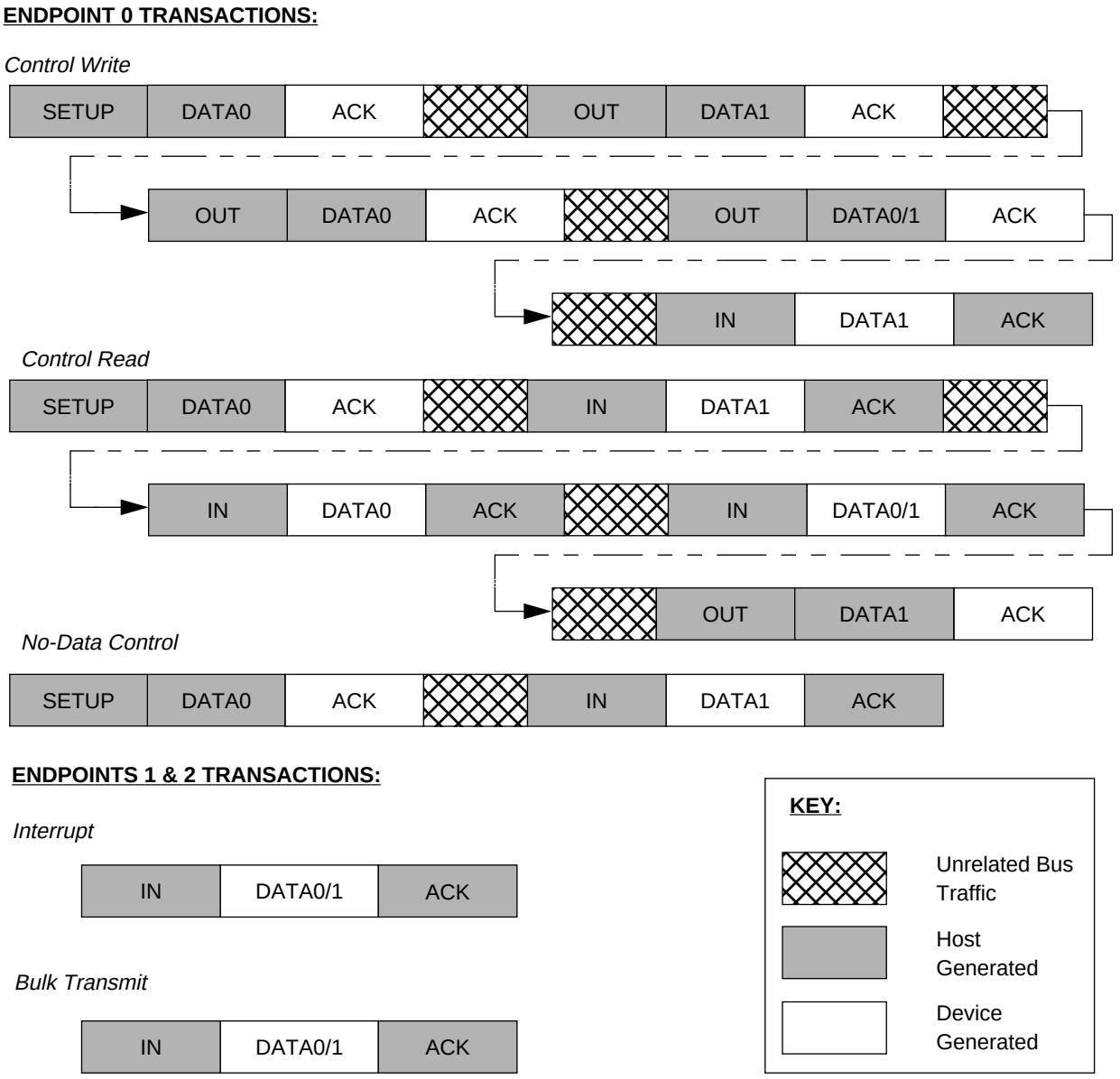


Figure 10-2. Supported Transaction Types per Endpoint

Each USB transaction is comprised of a series of packets. The MC68HC05JB4 USB module supports the packet types shown in **Figure 10-3**. Token packets are generated by the USB host and decoded by the USB device. Data and Handshake packets are both decoded and generated by the USB device depending on the type of transaction.

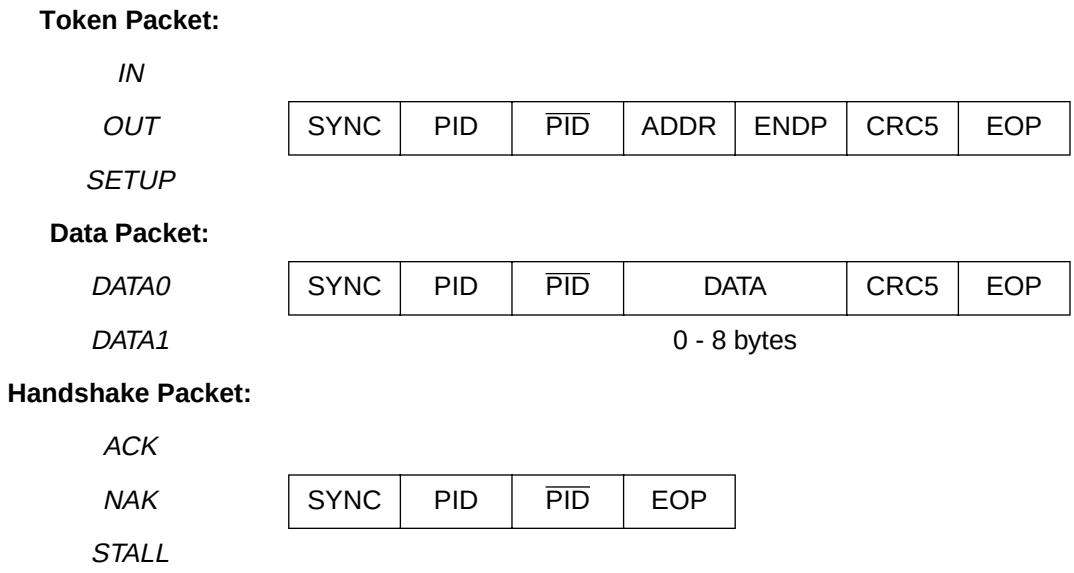


Figure 10-3. Supported USB Packet Types

The following sections will give some detail on each segment used to form a complete USB transaction.

10.2.1.1 Sync Pattern

The NRZI (See **Section 10.4.4.1**) bit pattern shown in **Figure 10-4** is used as a synchronization pattern and is prefixed to each packet. This pattern is equivalent to a data pattern of seven 0's followed by a 1 (0x80).

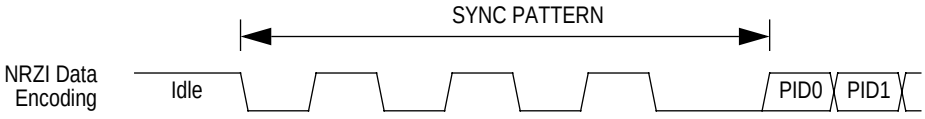


Figure 10-4. Sync Pattern

The start of a packet (SOP) is signaled by the originating port by driving the D+ and D– lines from the idle state (also referred to as the "J" state) to the opposite logic level (also referred to as the "K" state). This switch in levels represents the first bit of the Sync field. **Figure 10-5** shows the data signaling and voltage levels for the start of packet and the sync pattern.

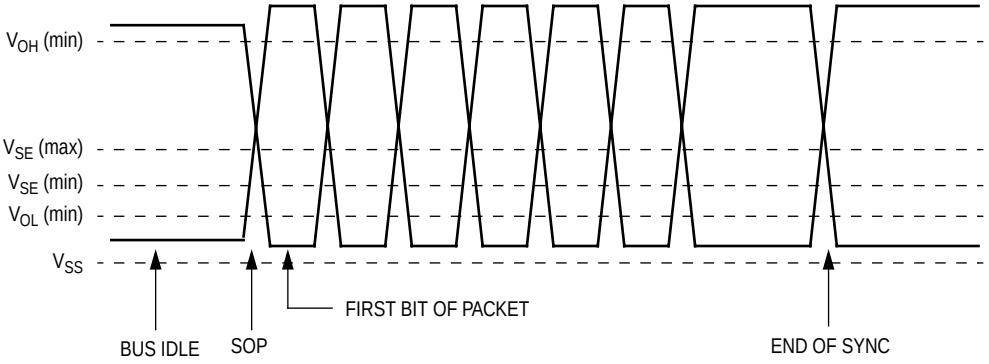


Figure 10-5. SOP, Sync Signaling and Voltage Levels

10.2.1.2 Packet Identifier Field

The Packet Identifier field is an eight bit number comprised of the four bit packet identification (PID) and its complement. The field follows the sync pattern and determines the direction and type of transaction on the bus. **Table 10-1** shows the PID values for the supported packet types.

Table 10-1. Supported Packet Identifiers

PID Value	PID Type
%1001	IN Token
%0001	OUT Token
%1101	SETUP Token
%0011	DATA0 Packet
%1011	DATA1 Packet
%0010	ACK Handshake
%1010	NAK Handshake
%1110	STALL Handshake

10.2.1.3 Address Field (ADDR)

The Address field is a seven bit number that is used to select a particular USB device. This field is compared to the lower seven bits of the UADDR register to determine if a given transaction is targeting the MC68HC05JB4 USB device.

10.2.1.4 Endpoint Field (ENDP)

The Endpoint field is a four bit number that is used to select a particular endpoint within a USB device. For the MC68HC05JB4, this will be a binary number between zero and two inclusive. Any other value will cause the transaction to be ignored.

10.2.1.5 Cyclic Redundancy Check (CRC)

Cyclic Redundancy Checks are used to verify the address and data stream of a USB transaction. This field is five bits wide for token packets and sixteen bits wide for data packets. CRCs are generated in the transmitter and sent on the USB data lines after both the endpoint field and the data field. **Figure 10-6** shows how the five bit CRC value is calculated from the data stream and verified for the address and endpoint fields of a token packet. **Figure 10-7** shows how the sixteen bit CRC value is calculated and either transmitted or verified for the data packet of a given transaction.

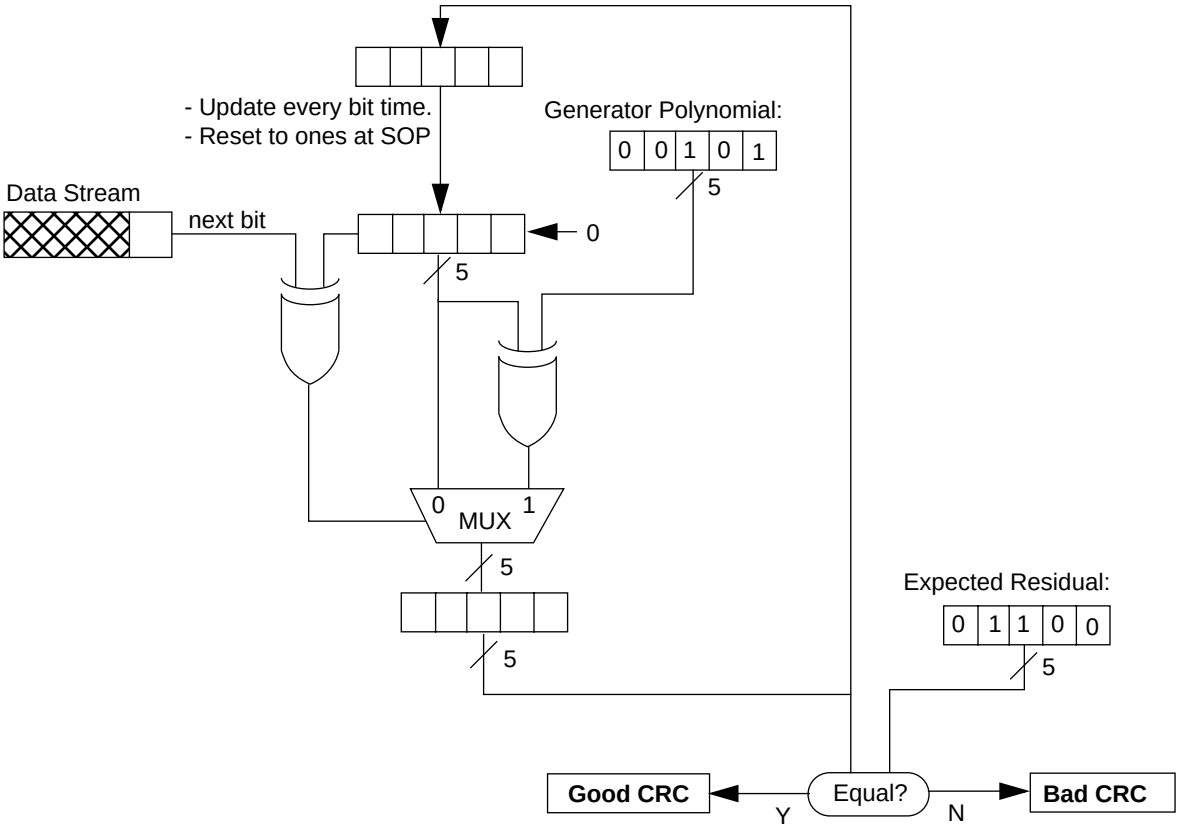


Figure 10-6. CRC Block Diagram for Address and Endpoint Fields

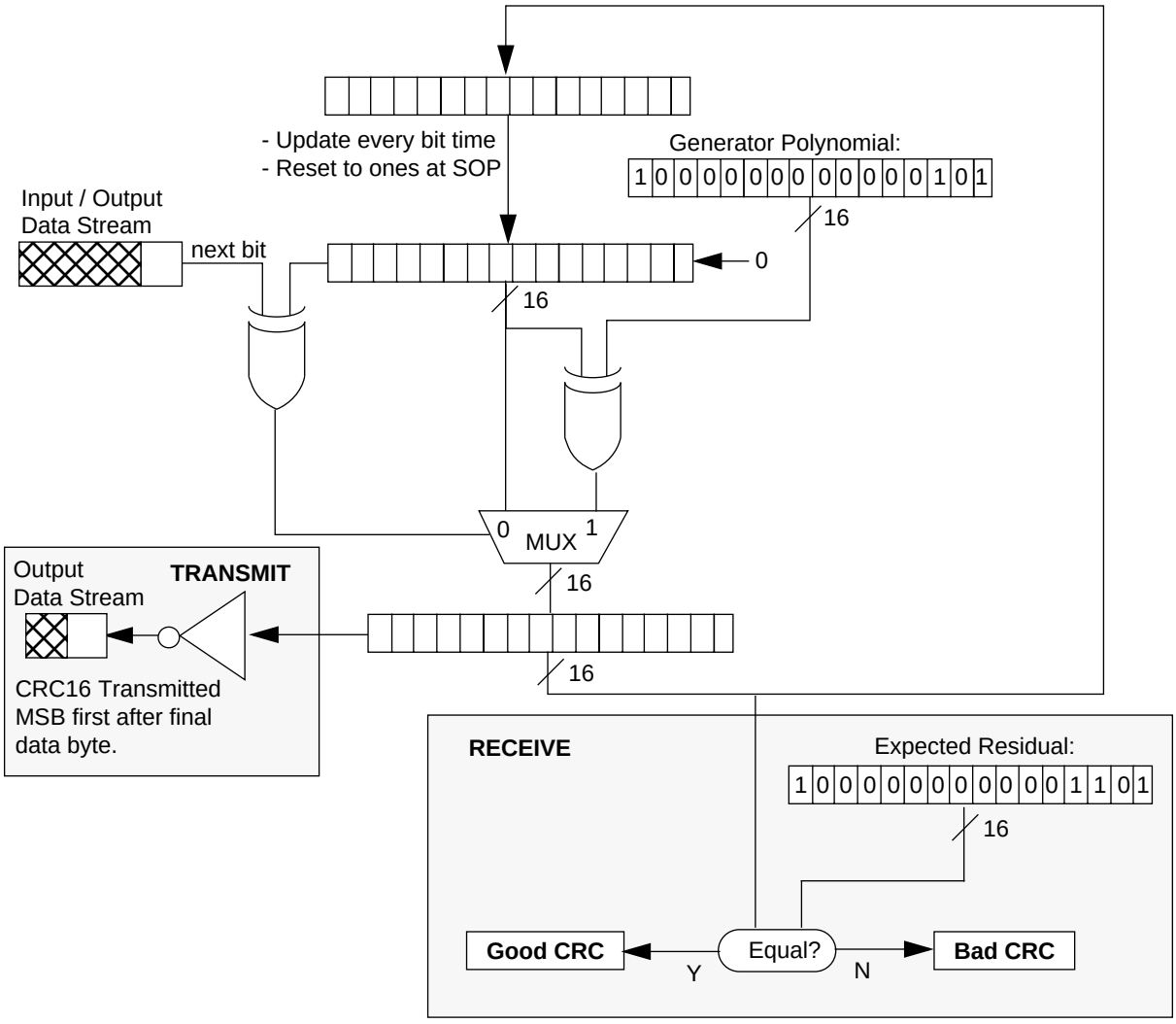


Figure 10-7. CRC Block Diagram for Data Packets

10.2.1.6 End Of Packet (EOP)

The single-ended 0 (SE0) state is used to signal an end of packet (EOP). The single-ended 0 state is indicated by both D+ and D– being below 0.8 V. EOP will be signaled by driving D+ and D– to the single-ended 0 state for two bit times followed by driving the lines to the idle state for one bit time. The transition from the single-ended 0 to the idle state defines the end of the packet. The idle state is asserted for one bit time and then both the D+ and D– output drivers are placed in their high-impedance state. The bus termination resistors hold the bus in the idle state. **Figure 10-8** shows the data signaling and voltage levels for an end of packet transaction.

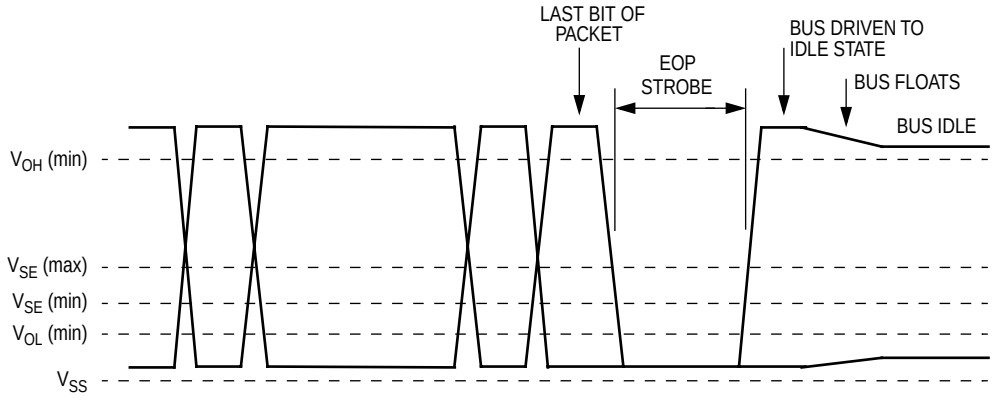


Figure 10-8. EOP Transaction Voltage Levels

The width of the SE0 in the EOP is about two bit times. The EOP width is measured with the same capacitive load used for maximum rise and fall times and is measured at the same level as the differential signal crossover points of the data lines.

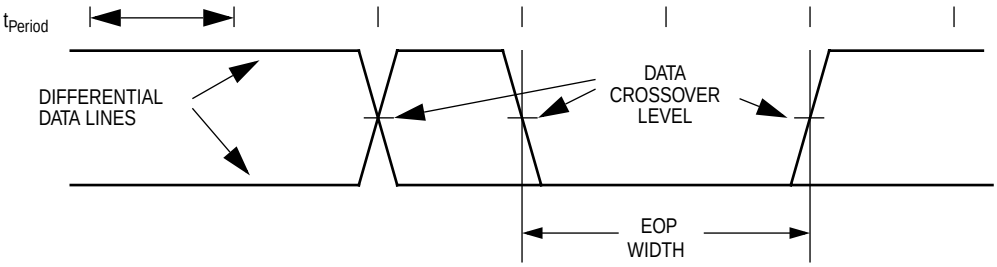


Figure 10-9. EOP Width Timing

10.2.2 Reset Signaling

A reset is signaled on the bus by the presence of an extended SE0 at the USB data pins of a device. The reset signaling is specified to be present for a minimum of 10 ms. An active device (powered and not in the suspend state) seeing a single-ended zero on its USB data inputs for more than 2.5µs may treat that signal as a reset, but must have interpreted the signaling as a reset within 5.5 µs. For a Low speed device, an SE0 condition between 4 and 8 low speed bit times represents a valid USB reset.

A USB sourced reset will hold the MC68HC05JB4 in reset for the duration of the reset on the USB bus. The RSTF bit in the USB interrupt register 0 (UIR0) will be set after the internal reset is removed (See **Section 10.5.2** for more detail).

After a reset is removed, the device will be in the attached, but not yet addressed or configured state (refer to Section 9.1 of the USB specification). The device must be able to accept a device address via a SET_ADDRESS command (refer to section 9.4 of the USB specification) no later than 10 ms after the reset is removed.

Reset can wake a device from the suspended mode. A device may take up to 10 ms to wake up from the suspended state.

10.2.3 Suspend

The MC68HC05JB4 supports suspend mode for low power. Suspend mode should be entered when the USB data lines are in the idle state for more than 3.0 ms. Entry into Suspend mode is controlled by the SUSPND bit in the USB Interrupt Register. Any low speed bus activity should keep the device out of the suspend state. Low speed devices are kept awake by periodic low speed EOP signals from the host. This is referred to as Low speed keep alive (refer to Section 11.2.5.1 of the USB specification).

Firmware should monitor the EOPF flag and enter suspend mode by setting the SUSPND bit if an EOP is not detected for 3 ms.

Per the USB specification, the MC68HC05JB4 is required to draw less than 500 µA from the V_{DD} supply when in the suspend state. This includes the current supplied by the voltage regulator to the 15 KΩ to ground termination resistors placed at the host end of the USB bus. This low current requirement means that firmware is responsible for entering STOP mode once the USB module has been placed in the suspend state.

10.2.4 Resume After Suspend

The MC68HC05JB4 can be activated from the suspend state by normal bus activity, a USB reset signal, or by a forced resume driven from the MC68HC05JB4.

10.2.4.1 Host Initiated Resume

The host signals resume by initiating resume signalling (“K” state) for at least 20 ms followed by a standard low speed EOP signal. This 20 ms ensures that all devices in the USB network are awakened.

After resuming the bus, the host must begin sending bus traffic within 3 ms to prevent the device from re-entering suspend mode.

10.2.4.2 USB Reset Signalling

Reset can wake a device from the suspended mode. A device may take up to 10 ms to wake up from the suspended state.

10.2.4.3 Remote Wake-up

The MC68HC05JB4 also supports the remote wake-up feature. The firmware has the ability to exit suspend mode by signaling a resume state to the upstream Host or Hub. A non-idle state (“K” state) on the USB data lines is accomplished by asserting the FRESUM bit in the UCR1 register.

When using the remote wake-up capability, the firmware must wait for at least 5 ms after the bus is in the idle state before sending the remote wake-up resume signaling. This allows the upstream devices to get into their suspend state and prepare for propagating resume signaling. The FRESUM bit should be asserted to cause the resume state on the USB data lines for at least 10ms, but not more than 15ms. Note that the resume signalling is controlled by the FRESUM bit and meeting the timing specifications is dependent on the firmware. When FRESUM is cleared by firmware, the data lines will return to their high impedance state. Refer to **Section 10.5.5** for more information about how the Force Resume (FRESUM) bit can be used to initiate the remote wake-up feature.

10.2.5 Low Speed Device

Externally, low speed devices are configured by the position of a pull-up resistor on the USB D– pin of the MC68HC05JB4. Low speed devices are terminated as shown in **Figure 10-10** with the pull-up on the D– line.

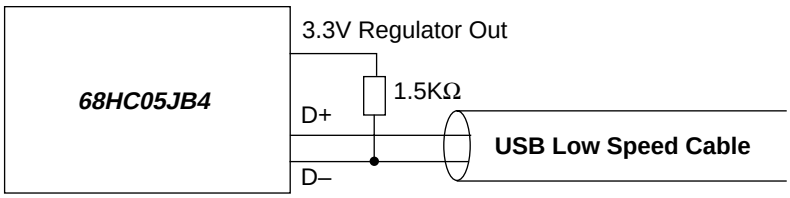


Figure 10-10. External Low Speed Device Configuration

For low speed transmissions, the transmitter's EOP width must be between 1.25µs and 1.50µs. These ranges include timing variations due to differential buffer delay and rise/fall time mismatches and to noise and other random effects. A low speed receiver must accept a 670ns wide SE0 followed by a J transition as a valid EOP. An SE0 narrower than 330ns or an SE0 not followed by a J transition must be rejected as an EOP. An EOP between 330ns and 670ns may be rejected or accepted as above. Any SE0 that is 2.5µs or wider is automatically a reset.

10.3 CLOCK REQUIREMENTS

The low speed data rate is nominally 1.5 Mbs. The OSCXCLK signal driven by the oscillator circuits is the clock source for the USB module and requires that a 6 MHz oscillator circuit be connected to the OSC1 and OSC2 pins. The permitted frequency tolerance for low speed functions is approximately ±1.5% (15000 ppm). This tolerance includes inaccuracies from all sources: initial frequency accuracy, crystal capacitive loading, supply voltage on the oscillator, temperature, and aging. The jitter in the low speed data rate must be less than 10 ns. This tolerance allows the use of resonators in low cost, low speed devices.

10.4 HARDWARE DESCRIPTION

The USB module as previously shown in **Figure 10-1** contains four functional blocks: a 3.3 volt Regulator, a LS USB transceiver, the USB control logic, and the

USB registers. The following will detail the function of the regulator, transceiver and control logic. See **Section 10.5** for the register discussion.

10.4.1 Voltage Regulator

The USB data lines are required by the USB Specification to have a maximum output voltage between 2.8V and 3.6V. The data lines are also required to have an external 1.5KΩ pullup resistor connected between a data line and a voltage source between 3.0V and 3.6V. Since the power provided by the USB cable is specified to be between 4.4V and 5.0V, an on-chip regulator is used to drop the voltage to the appropriate level for sourcing the USB transceiver and external pullup resistor. An output pin driven by the regulator voltage is provided to source the 1.5KΩ external resistor. **Figure 10-11** shows the worst case electrical connection for the voltage regulator.

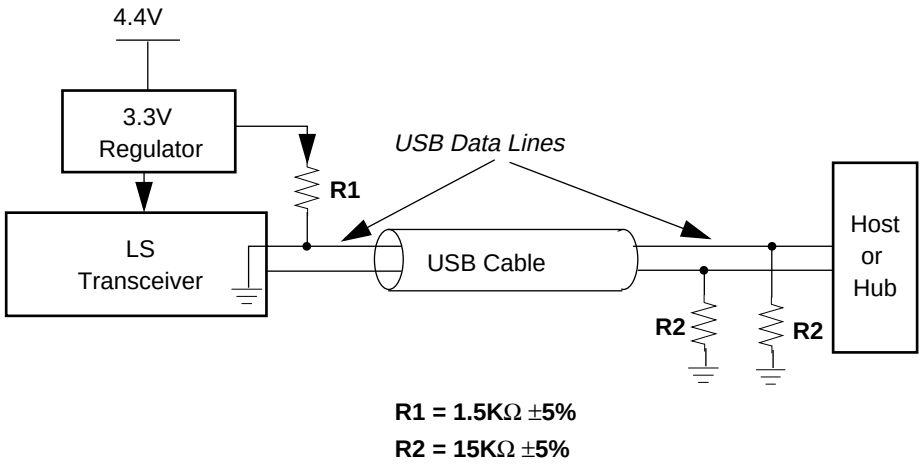


Figure 10-11. Regulator Electrical Connections

10.4.2 USB Transceiver

The USB transceiver provides the physical interface to the USB D+ and D– data lines. The transceiver is composed of two parts: an output drive circuit and a differential receiver.

10.4.2.1 Output Driver Characteristics

The USB transceiver uses a differential output driver to drive the USB data signal onto the USB cable. The static output swing of the driver in its low state is below the V_{OL} of 0.3 V with a 1.5 kΩ load to 3.6 V and in its high state is above the V_{OH} of 2.8 V with a 15 kΩ load to ground. The output swings between the differential high and low state are well balanced to minimize signal skew. Slew rate control on the driver is used to minimize the radiated noise and cross talk. The driver's outputs support three-state operation to achieve bi-directional half duplex operation. The driver can tolerate a voltage on the signal pins of –0.5 V to 3.8 V

with respect to local ground reference without damage.

10.4.2.2 Low Speed (1.5 Mbs) Driver Characteristics

The rise and fall time of the signals on this cable are greater than 75 ns to keep RFI emissions under FCC class B limits, and less than 300 ns to limit timing delays and signaling skews and distortions. The driver reaches the specified static signal levels with smooth rise and fall times, and minimal reflections and ringing when driving the cable. This driver is used only on network segments between low speed devices and the ports to which they are connected.

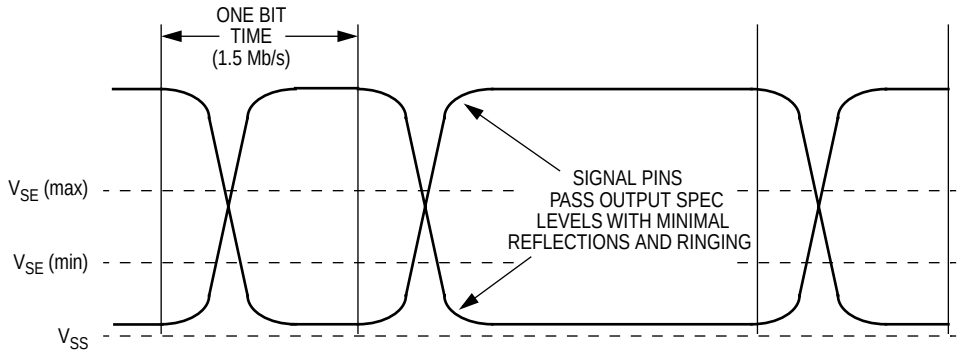


Figure 10-12. Low Speed Driver Signal Waveforms

10.4.3 Receiver Characteristics

USB data transmission is done with differential signals. A differential input receiver is used to accept the USB data signal. A differential 1 on the bus is represented by D+ being at least 200 mV more positive than D– as seen at the receiver, and a differential 0 is represented by D– being at least 200 mV more positive than D+ as seen at the receiver. The signal cross over point must be between 1.3V and 2.0V.

The receiver features an input sensitivity of 200 mV when both differential data inputs are in the range of 0.8 V to 2.5 V with respect to the local ground reference. This is called the common mode input voltage range. Proper data reception is also achieved when the differential data lines are outside the common mode range, as shown in **Figure 10-13**. The receiver can tolerate static input voltages between –0.5V to 3.8 V with respect to its local ground reference without damage. In addition to the differential receiver, there is a single-ended receiver (schmitt trigger) for each of the two data lines.

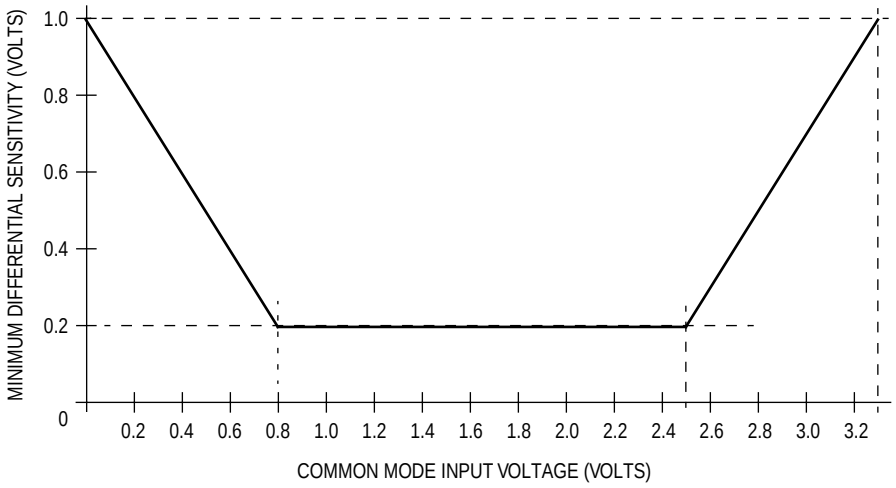


Figure 10-13. Differential Input Sensitivity Over Entire Common Mode Range

10.4.3.1 Receiver Data Jitter

The data receivers for all types of devices must be able to properly decode the differential data in the presence of jitter. The more of the bit cell that any data edge can occupy and still be decoded, the more reliable the data transfer will be. Data receivers are required to decode differential data transitions that occur in a window plus and minus a nominal quarter bit cell from the nominal (centered) data edge position.

Jitter will be caused by the delay mismatches and by mismatches in the source and destination data rates (frequencies). The receive data jitter budget for low speed is given in the electrical section of the this specification. The specification includes the consecutive (next) and paired transition values for each source of jitter.

10.4.3.2 Data Source Jitter

The source of data can have some variation (jitter) in the timing of edges of the data transmitted. The time between any set of data transitions is $N \times T_{PERIOD} \pm$ jitter time, where ‘N’ is the number of bits between the transitions and T_{PERIOD} is defined as the actual period of the data rate. The data jitter is measured with the same capacitive load used for maximum rise and fall times and is measured at the crossover points of the data lines as shown in **Figure 10-14**.

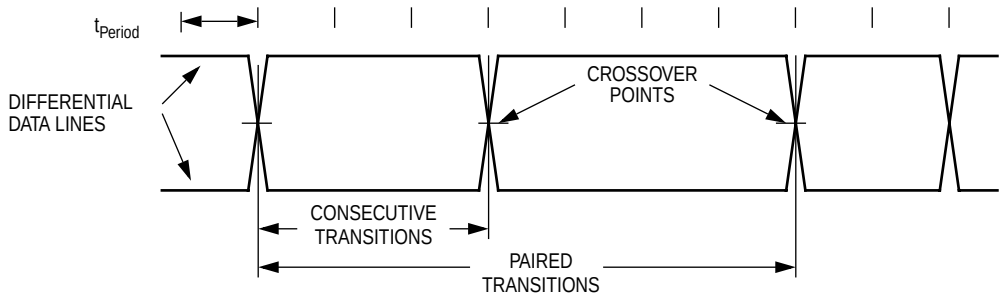


Figure 10-14. Data Jitter

For low speed transmissions, the jitter time for any consecutive differential data transitions must be within ±25 ns and within ±10 ns for any set of paired differential data transitions. These jitter numbers include timing variations due to differential buffer delay, rise/fall time mismatches, internal clock source jitter, and to noise and other random effects.

10.4.3.3 Data Signal Rise and Fall Time

The output rise time and fall time are measured between 10% and 90% of the signal. Edge transition time for the rising and falling edges of low speed signals is 75 ns (minimum) into a capacitive load (C_L) of 50 pF and 300 ns (maximum) into a capacitive load of 350 pF. The rising and falling edges should be smooth transitional (monotonic) when driving the cable to avoid excessive EMI.

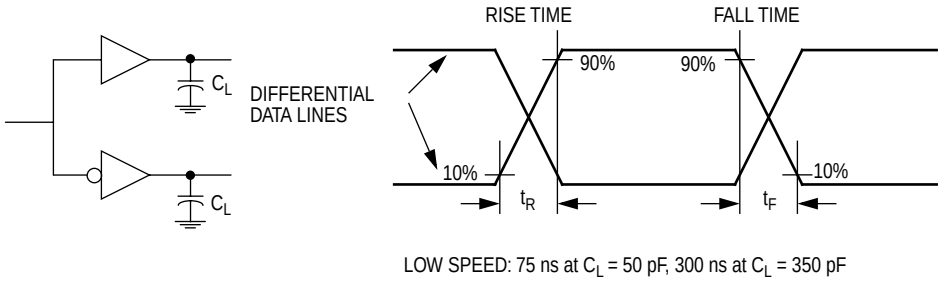


Figure 10-15. Data Signal Rise and Fall Time

10.4.4 USB Control Logic

The USB control logic manages data movement between the CPU and the transceiver. The control logic handles both transmit and receive operations on the USB. It contains the logic used to manipulate the transceiver and the endpoint registers. The logic contains byte count buffers for transmit operations that load the active transmit endpoints byte count and use this to determine the number of bytes to transfer. This same buffer is used for receive transactions to count the number of bytes received and, upon the end of the transaction, transfer that number to the receive endpoints byte count register.

When transmitting, the control logic handles parallel to serial conversion, CRC generation, NRZI encoding, and bit stuffing.

When Receiving, the control logic handles Sync detection, packet identification, end of packet detection, bit (un)stuffing, NRZI decoding, CRC validation, and serial to parallel conversion. Errors detected by the control logic include bad CRC, time-out while waiting for EOP, and bit stuffing violations.

10.4.4.1 Data Encoding/Decoding

The USB employs NRZI data encoding when transmitting packets. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. **Figure 10-16** shows a data stream and the NRZI equivalent and **Figure 10-17** is a flow diagram for NRZI. The high level represents the J state on the data lines in this and subsequent figures showing NRZI encoding. A string of zeros causes the NRZI data to toggle each bit time. A string of ones causes long periods with no transitions in the data.

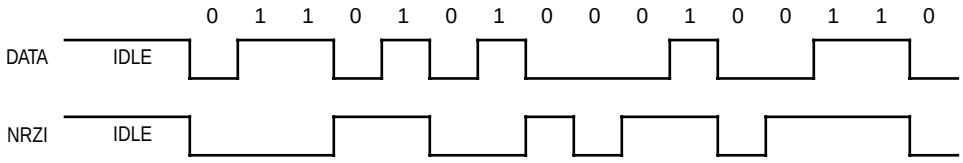


Figure 10-16. NRZI Data Encoding

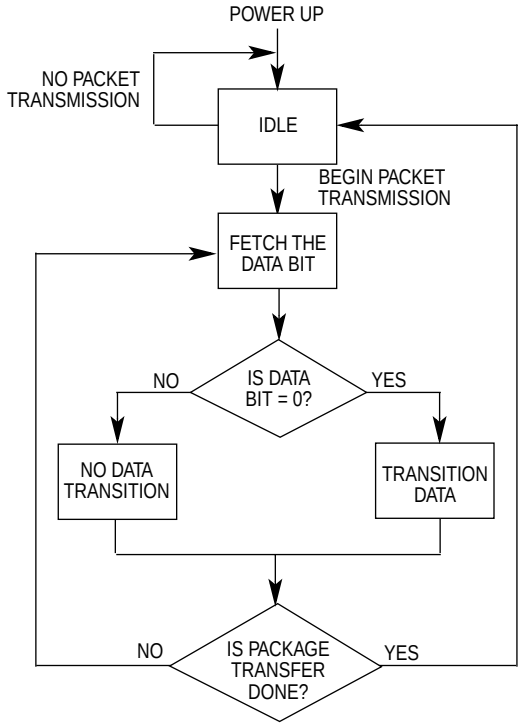


Figure 10-17. Flow Diagram for NRZI

10.4.4.2 Bit Stuffing

In order to ensure adequate signal transitions, bit stuffing is employed by the transmitting device when sending a packet on the USB (see **Figure 10-18** and **Figure 10-19**). A 0 is inserted after every six consecutive 1's in the data stream before the data is NRZI encoded to force a transition in the NRZI data stream. This gives the receiver logic a data transition at least once every seven bit times to guarantee the data and clock lock. The receiver must decode the NRZI data, recognize the stuffed bits, and discard them. Bit stuffing is enabled beginning with the Sync Pattern and throughout the entire transmission. The data "one" that ends the Sync Pattern is counted as the first one in a sequence. Bit stuffing is always enforced, without exception. If required by the bit stuffing rules, a zero bit will be inserted even if it is the last bit before the end-of-packet (EOP) signal.

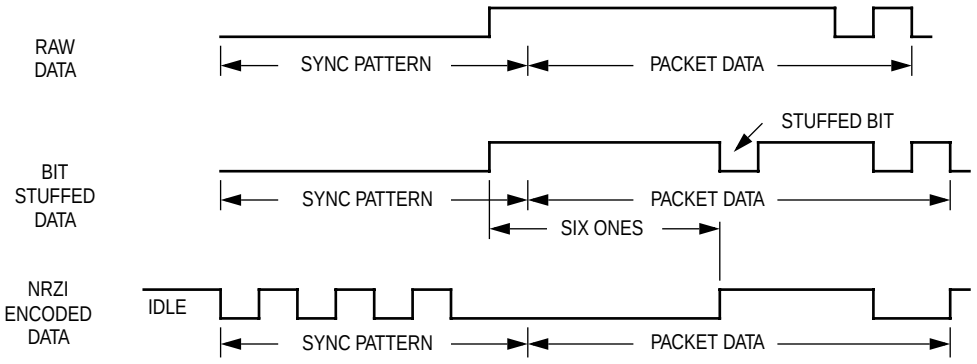


Figure 10-18. Bit Stuffing

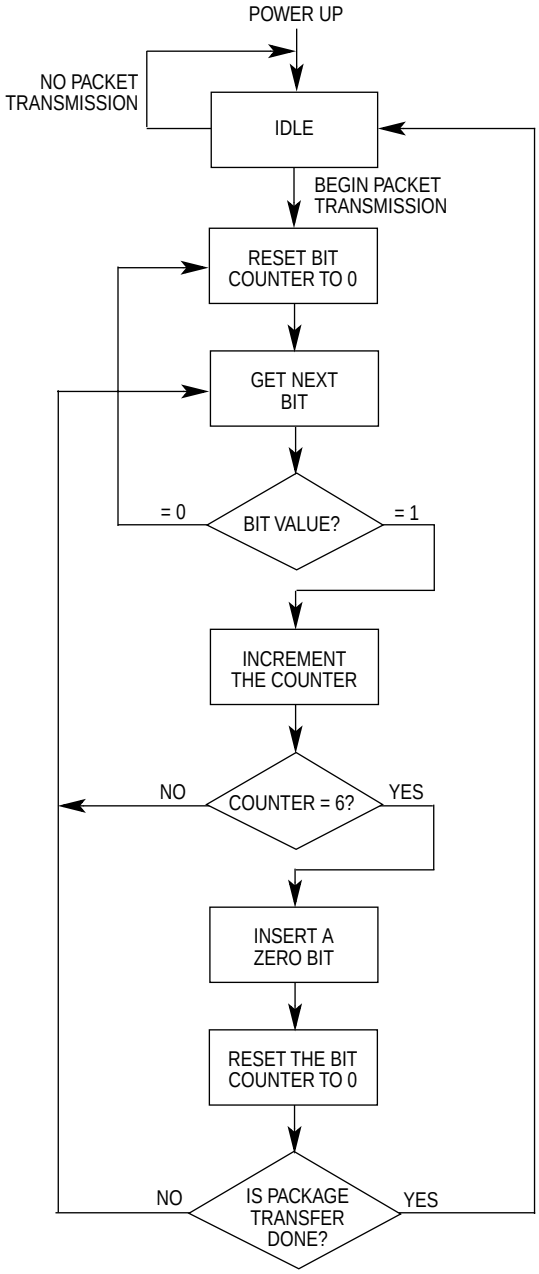


Figure 10-19. Flow Diagram for Bit Stuffing

10.5 I/O REGISTER DESCRIPTION

The USB Endpoint registers are comprised of a set of control/status registers and twenty-four data registers that provide storage for the buffering of data between the USB and the CPU. These registers are shown in **Table 10-2**.

Table 10-2. Register Summary

Register Name	Bit 7	6	5	4	3	2	1	Bit 0	Addr
USB Control Register 2 (UCR2)		0	TX1ST	0	ENABLE2	ENABLE1	STALL2	STALL1	$0037
		TX1STR							
USB Address Register (UADDR)	USBEN	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0	$0038
USB Interrupt Register 0 (UIR0)	TXD0F	RXD0F	RSTF	SUSPND	TXD0IE	RXD0IE	0	0	$0039
							TXD0FR	RXD0FR	
USB Interrupt Register 1 (UIR1)	TXD1F	EOPF	RESUMF	0	TXD1IE	EOPIE	0	0	$003A
				RESUMFR			TXD1FR	EOPFR	
USB Control Register 0 (UCR0)	T0SEQ	STALL0	TX0E	RX0E	TP0SIZ3	TP0SIZ2	TP0SIZ1	TP0SIZ0	$003B
USB Control Register 1 (UCR1)	T1SEQ	ENDADD	TX1E	FRESUM	TP1SIZ3	TP1SIZ2	TP1SIZ1	TP1SIZ0	$003C
USB Status Register (USR)	RSEQ	SETUP	0	0	RPSIZ3	RPSIZ2	RPSIZ1	RPSIZ0	$003D
USB Endpoint 0 Data Register 0 (UE0D0)	UE0RD7	UE0RD6	UE0RD5	UE0RD4	UE0RD3	UE0RD2	UE0RD1	UE0RD0	$0020
	UE0TD7	UE0TD6	UE0TD5	UE0TD4	UE0TD3	UE0TD2	UE0TD1	UE0TD0	
↓									↓
USB Endpoint 0 Data Register 7 (UE0D7)	UE0RD7	UE0RD6	UE0RD5	UE0RD4	UE0RD3	UE0RD2	UE0RD1	UE0RD0	$0027
	UE0TD7	UE0TD6	UE0TD5	UE0TD4	UE0TD3	UE0TD2	UE0TD1	UE0TD0	
USB Endpoint 1/2 Data Register 0 (UE1D0)									$0028
	UE1TD7	UE1TD6	UE1TD5	UE1TD4	UE1TD3	UE1TD2	UE1TD1	UE1TD0	
↓									↓
USB Endpoint 1/2 Data Register 7 (UE1D7)									$002F
	UE1TD7	UE1TD6	UE1TD5	UE1TD4	UE1TD3	UE1TD2	UE1TD1	UE1TD0	

= Unimplemented

10.5.1 USB Address Register (UADDR)

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UADDR	R	USBEN	UADD6	UADD5	UADD4	UADD3	UADD2	UADD1	UADD0
$0038	W								
reset:		0	0	0	0	0	0	0	0

Figure 10-20. USB Address Register (UADDR)

USBEN — USB Module Enable

This read/write bit enables and disables the USB module and the USB pins. When USBEN is clear, the USB module will not respond to any tokens. Reset clears this bit.

1 = USB function enabled
0 = USB function disabled

UADD6-UADD0 — USB Function Address

These bits specify the USB address of the device. Reset clears these bits.

10.5.2 USB Interrupt Register 0 (UIR0)

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UIR0	R	TXD0F	RXD0F	RSTF	SUSPND	TXD0IE	RXD0IE	0	0
$0039	W							TXD0FR	RXD0FR
reset:		0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-21. USB Interrupt Register 0 (UIR0)

TXD0F — Endpoint 0 Data Transmit Flag

This read only bit is set after the data stored in Endpoint 0 transmit buffers has been sent and an ACK handshake packet from the host is received. Once the next set of data is ready in the transmit buffers, software must clear this flag by writing a logic 1 to the TXD0FR bit. To enable the next data packet transmission, TX0E must also be set. If TXD0F bit is not cleared, a NAK handshake will be returned in the next IN transaction.

Reset clears this bit. Writing a logic 0 to TXD0F has no effect.

1 = Transmit on Endpoint 0 has occurred
0 = Transmit on Endpoint 0 has not occurred

RXD0F — Endpoint 0 Data Receive Flag

This read only bit is set after the USB module has received a data packet and responded with an ACK handshake packet. Software must clear this flag by writing a logic 1 to the RXD0FR bit after all of the received data has been read. Software must also set RX0E bit to one to enable the next data packet reception. If RXD0F bit is not cleared, a NAK handshake will be returned in the next OUT transaction.

Reset clears this bit. Writing a logic 0 to RXD0F has no effect.

1 = Receive on Endpoint 0 has occurred
0 = Receive on Endpoint 0 has not occurred

RSTF — USB Reset Flag

This read only bit is set when a valid reset signal state is detected on the D+ and D– lines. This reset detection will also generate an internal reset signal to reset the CPU and other peripherals including the USB module. This bit is cleared by a POR reset.

SUSPND — USB Suspend Flag

To save power, this read/write bit should be set by the software if a 3ms constant idle state is detected on USB bus. Setting this bit stops the clock to the USB and causes the USB module to enter Suspend mode. Unnecessary analog circuitry will be powered down. Software must clear this bit after the Resume flag (RESUMF) is set while this Resume interrupt flag is serviced.

TXD0IE — Endpoint 0 Transmit Interrupt Enable

This read/write bit enables the Transmit Endpoint 0 to generate a USB interrupt when the TXD0F bit becomes set.

1 = USB interrupts enabled for Transmit Endpoint 0
0 = USB interrupts disabled for Transmit Endpoint 0

RXD0IE — Endpoint 0 Receive Interrupt Enable

This read/write bit enables the Transmit Endpoint 0 to generate a USB interrupt when the RXD0F bit becomes set.

1 = USB interrupts enabled for Receive Endpoint 0
0 = USB interrupts disabled for Receive Endpoint 0

TXD0FR — Endpoint 0 Transmit Flag Reset

Writing a logic 1 to this write only bit will clear the TXD0F bit if it is set.Writing a logic 0 to TXD0FR has no effect. Reset clears this bit.

RXD0FR — Endpoint 0 Receive Flag Reset

Writing a logic 1 to this write only bit will clear the RXD0F bit if it is set.Writing a logic 0 to RXD0FR has no effect. Reset clears this bit.

10.5.3 USB Interrupt Register 1 (UIR1)

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UIR1	R	TXD1F	EOPF	RESUMF	0	TXD1IE	EOPIE	0	0
$003A	W				RESUMFR			TXD1FR	EOPFR
reset:		0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-22. USB Interrupt Register 1(UIR1)

TXD1F — Endpoint 1/Endpoint 2 Data Transmit Flag

This read only bit is shared by Endpoint 1 and Endpoint 2. It is set after the data stored in the shared Endpoint 1/Endpoint 2 transmit buffer has been sent and an ACK handshake packet from the host is received. Once the next set of data is ready in the transmit buffers, software must clear this flag by writing a logic 1 to the TXD1FR bit. To enable the next data packet transmission, TX1E must also be set. If TXD1F bit is not cleared, a NAK handshake will be returned in the next IN transaction.

Reset clears this bit. Writing a logic 0 to TXD1F has no effect.

1 = Transmit on Endpoint 1 or Endpoint 2 has occurred
0 = Transmit on Endpoint 1 or Endpoint 2 has not occurred

EOPF — End of Packet Detect Flag

This read only bit is set when a valid End-of-Packet sequence is detected on the D+ and D– lines. Software must clear this flag by writing a logic 1 to the EOPFR bit.

Reset clears this bit. Writing a logic 0 to EOPF has no effect.

1 = End-of-Packet sequence has been detected
0 = End-of-Packet sequence has not been detected

RESUMF — Resume Flag

This read only bit is set when USB bus activity is detected while the SUSPND bit is set. Software must clear this flag by writing a logic 1 to the RESUMFR bit. Reset clears this bit. Writing a logic 0 to RESUMF has no effect.

1 = USB bus activity has been detected
0 = No USB bus activity has been detected

RESUMFR — Resume Flag Reset

Writing a logic 1 to this write only bit will clear the RESUMF bit if it is set. Writing a logic 0 to RESUMFR has no effect. Reset clears this bit.

TXD1IE — Endpoint 1/Endpoint 2 Transmit Interrupt Enable

This read/write bit enables the USB to generate an interrupt when the shared Transmit Endpoint 1/Endpoint 2 interrupt flag (TXD1F) bit becomes set. Reset clears this bit.

1 = USB interrupts enabled for Transmit Endpoints 1 and 2
0 = USB interrupts disabled for Transmit Endpoints 1 and 2

EOPIE — End of Packet Detect Interrupt Enable

This read/write bit enables the USB to generate an interrupt when the EOPF bit becomes set. Reset clears this bit.

1 = USB interrupts enabled for Transmit Endpoints 1 and 2
0 = USB interrupts disabled for Transmit Endpoint 1 and 2

TXD1FR — Endpoint 1/Endpoint 2 Transmit Flag Reset

Writing a logic 1 to this write only bit will clear the TXD1F bit if it is set. Writing a logic 0 to TXD1FR has no effect. Reset clears this bit.

EOPFR — End of Packet Flag Reset

Writing a logic 1 to this write only bit will clear the EOPF bit if it is set. Writing a logic 0 to the EOPFR has no effect. Reset clears this bit.

10.5.4 USB Control Register 0 (UCR0)

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UCR0	R	T0SEQ	STALL0	TX0E	RX0E	TP0SIZ3	TP0SIZ2	TP0SIZ1	TP0SIZ0
$003B	W								
reset:		0	0	0	0	0	0	0	0

Figure 10-23. USB Control Register 0 (UCR0)

T0SEQ — Endpoint 0 Transmit Sequence Bit

This read/write bit determines which type of data packet (DATA0 or DATA1) will be sent during the next IN transaction. Toggling of this bit must be controlled by software. Reset clears this bit.

1 = DATA1 Token active for next Endpoint 0 transmit
0 = DATA0 Token active for next Endpoint 0 transmit

STALL0 — Endpoint 0 Force Stall Bit

This read/write bit causes Endpoint 0 to return a STALL handshake when polled by either an IN or OUT token by the USB Host Controller. The USB hardware clears this bit when a SETUP token is received. Reset clears this bit.

1 = Send STALL handshake
0 = Default

TX0E — Endpoint 0 Transmit Enable

This read/write bit enables a transmit to occur when the USB Host controller sends an IN token to Endpoint 0. Software should set this bit when data is ready to be transmitted. It must be cleared by software when no more Endpoint 0 data needs to be transmitted.

If this bit is 0 or the TXD0F is set, the USB will respond with a NAK handshake to any Endpoint 0 IN tokens. Reset clears this bit.

1 = Data is ready to be sent.
0 = Data is not ready. Respond with NAK.

RX0E — Endpoint 0 Receive Enable

This read/write bit enables a receive to occur when the USB Host controller sends an OUT token to Endpoint 0. Software should set this bit when data is ready to be received. It must be cleared by software when data cannot be received.

If this bit is 0 or the RXD0F is set, the USB will respond with a NAK handshake to any Endpoint 0 OUT tokens. Reset clears this bit.

1 = Data is ready to be received.
0 = Not ready for data. Respond with NAK.

TP0SIZ3-TP0SIZ0 — Endpoint 0 Transmit Data Packet Size

These read/write bits store the number of transmit data bytes for the next IN token request for Endpoint 0. These bits are cleared by reset.

10.5.5 USB Control Register 1 (UCR1)

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UCR1	R	T1SEQ	ENDADD	TX1E	FRESUM	TP1SZ3	TP1SIZ2	TP1SIZ1	TP1SIZ0
$003C	W								
reset:		0	0	0	0	0	0	0	0

Figure 10-24. USB Control Register 1 (UCR1)

T1SEQ — Endpoint1/Endpoint 2 Transmit Sequence Bit

This read/write bit determines which type of data packet (DATA0 or DATA1) will be sent during the next IN transaction directed to Endpoint 1 or Endpoint 2. Toggling of this bit must be controlled by software. Reset clears this bit.

1 = DATA1 Token active for next Endpoint 1/Endpoint 2 transmit
0 = DATA0 Token active for next Endpoint 1/Endpoint 2 transmit

ENDADD — Endpoint Address Select

This read/write bit specifies whether the data inside the registers UE1D0-UE1D7 are used for Endpoint 1 or Endpoint 2. If all the conditions for a successful Endpoint 2 USB response to a hosts IN token are satisfied (TXD1F=0, TX1E=1, STALL2=0, and ENABLE2=1) except that the ENDADD bit is configured for Endpoint 1, the USB responds with a NAK handshake packet.

1 = The data buffers are used for Endpoint 2
0 = The data buffers are used for Endpoint 1

TX1E — Endpoint 1/Endpoint 2 Transmit Enable

This read/write bit enables a transmit to occur when the USB Host controller sends an IN token to Endpoint 1 or Endpoint 2. The appropriate endpoint enable bit, ENABLE1 or ENABLE2 bit in the UCR2 register, should also be set. Software should set the TX1E bit when data is ready to be transmitted. It must be cleared by software when no more data needs to be transmitted.

If this bit is 0 or the TXD1F is set, the USB will respond with a NAK handshake to any Endpoint 1 or Endpoint 2 directed IN tokens. Reset clears this bit.

1 = Data is ready to be sent.
0 = Data is not ready. Respond with NAK.

FRESUM — Force Resume

This read/write bit forces a resume state (“K” or non-idle state) onto the USB data lines to initiate a remote wake-up. Software should control the timing of the forced resume to be between 10ms and 15 ms. Setting this bit will not cause the RESUMF bit to set.

1 = Force data lines to “K” state
0 = Default

TP1SIZ3-TP1SIZ0 — Endpoint 1/Endpoint 2 Transmit Data Packet Size

These read/write bits store the number of transmit data bytes for the next IN token request for Endpoint 1 or Endpoint 2. These bits are cleared by reset.

10.5.6 USB Control Register 2 (UCR2)

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UCR2	R		0	TX1ST	0	ENABLE2	ENABLE1	STALL2	STALL1
$0037	W		TX1STR						
reset:		-	-	0	-	0	0	0	0

= Unimplemented

Figure 10-25. USB Control Register 2 (UCR2)

TX1STR — Clear Transmit First Flag

Writing a logic 1 to this write-only bit will clear the TX1ST bit if it is set. Writing a logic 0 to the TX1STR has no effect. Reset clears this bit.

TX1ST — Transmit First Flag

This read-only bit is set if the Endpoint 0 Data Transmit Flag (TXD0F) is set when the USB control logic is setting the Endpoint 0 Data Receive Flag (RXD0F). That is, this bit will be set if an Endpoint 0 Transmit Flag is still set at the end of an Endpoint 0 reception. This bit lets the firmware know that the Endpoint 0 transmission happened before the Endpoint 0 reception. Reset clears this bit.

1 = IN transaction occurred before SETUP/OUT.
0 = IN transaction occurred after SETUP/OUT.

ENABLE2 — Endpoint 2 Enable

This read/write bit enables Endpoint 2 and allows the USB to respond to IN packets addressed to Endpoint 2. Reset clears this bit.

1 = Endpoint 2 is enabled and can respond to an IN token.
0 = Endpoint 2 is disabled

ENABLE1 — Endpoint 1 Enable

This read/write bit enables Endpoint 1 and allows the USB to respond to IN packets addressed to Endpoint 1. Reset clears this bit.

1 = Endpoint 1 is enabled and can respond to an IN token.
0 = Endpoint 1 is disabled

STALL2 — Endpoint 2 Force Stall Bit

This read/write bit causes Endpoint 2 to return a STALL handshake when polled by either an IN or OUT token by the USB Host Controller. Reset clears this bit.

1 = Send STALL handshake.
0 = Default

STALL1 — Endpoint 1 Force Stall Bit

This read/write bit causes Endpoint 1 to return a STALL handshake when polled by either an IN or OUT token by the USB Host Controller. Reset clears this bit.

1 = Send STALL handshake
0 = Default

10.5.7 USB Status Register (USR)

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
USR	R	RSEQ	SETUP	0	0	RPSIZ3	RPSIZ2	RPSIZ1	RPSIZ0
$003D	W								
reset:		U	U	U	U	U	U	U	U

= Unimplemented

Figure 10-26. USB Status Register (USR)

RSEQ — Endpoint 0 Receive Sequence Bit

This read only bit indicates the type of data packet last received for Endpoint 0 (DATA0 or DATA1).

1 = DATA1 Token received in last Endpoint 0 receive
0 = DATA0 Token received in last Endpoint 0 receive

SETUP — SETUP Token Detect Bit

This read only bit indicates that a valid SETUP token has been received.

1 = Last token received for Endpoint 0 was a SETUP token
0 = Last token received for Endpoint 0 was not a SETUP token

RPSIZ3-RPSIZ0 — Endpoint 0 Receive Data Packet Size

These read only bits store the number of data bytes received for the last OUT or SETUP transaction for Endpoint 0. These bits are not affected by reset.

10.5.8 USB Endpoint 0 Data Registers (UE0D0-UE0D7)

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UE0D0	R	UE0RD7	UE0RD6	UE0RD5	UE0RD4	UE0RD3	UE0RD2	UE0RD1	UE0RD0
$0020	W	UE0TD7	UE0TD6	UE0TD5	UE0TD4	UE0TD3	UE0TD2	UE0TD1	UE0TD0
to		:	:	:	:	:	:	:	:
		:	:	:	:	:	:	:	:
UE0D7	R	UE0RD7	UE0RD6	UE0RD5	UE0RD4	UE0RD3	UE0RD2	UE0RD1	UE0RD0
$0027	W	UE0TD7	UE0TD6	UE0TD5	UE0TD4	UE0TD3	UE0TD2	UE0TD1	UE0TD0
reset:		X	X	X	X	X	X	X	X

Figure 10-27. USB Endpoint 0 Data Register (UE0D0-UE0D7)

UE0RD7 - UE0RD0 — Endpoint 0 Receive Data Buffer

These read only bits are serially loaded with OUT token or SETUP token data received over the USB's D+ and D– pins.

UE0TD7 - UE0TD0 — Endpoint 0 Transmit Data Buffer

These write only buffers are loaded by software with data to be sent on the USB bus on the next IN token directed at Endpoint 0.

10.5.9 USB Endpoint 1/Endpoint 2 Data Registers (UE1D0-UE1D7)

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
UE1D0	R								
$0028	W	UE1TD7	UE1TD6	UE1TD5	UE1TD4	UE1TD3	UE1TD2	UE1TD1	UE1TD0
to		:	:	:	:	:	:	:	:
		:	:	:	:	:	:	:	:
UE1D7	R								
$002F	W	UE1TD7	UE1TD6	UE1TD5	UE1TD4	UE1TD3	UE1TD2	UE1TD1	UE1TD0
reset:		X	X	X	X	X	X	X	X

Figure 10-28. USB Endpoint 1/Endpoint2 Data Registers (UE1D0-UE1D7)

UE1TD7 - UE1TD0 — Endpoint 1/ Endpoint 2 Transmit Data Buffer

These write only buffers are loaded by software with data to be sent on the USB bus on the next IN token directed at Endpoint 1 or Endpoint 2. These buffers are shared by Endpoints 1 and 2 and depend on proper configuration of the ENDADD bit.

10.6 USB INTERRUPTS

The USB module is capable of generating interrupts and causing the CPU to execute the USB interrupt service routine. There are three types of USB interrupts:

- End of Transaction interrupts signify a completed transaction (receive or transmit)
- Resume interrupts signify that the USB bus is reactivated after having been suspended
- End of Packet interrupts signify that a low speed end of packet signal was detected

All USB interrupts share the same interrupt vector. Firmware is responsible for determining which interrupt is active.

10.6.1 USB End of Transaction Interrupt

There are three possible end of transaction interrupts: Endpoint 0 Receive, Endpoint 0 Transmit, and a shared Endpoint 1 or Endpoint 2 Transmit. End of transaction interrupts occur as detailed in the following sections.

10.6.1.1 Receive Control Endpoint 0

For a Control OUT transaction directed at Endpoint 0, the USB module will generate an interrupt by setting the RXD0F flag in the UIR0 register. The conditions necessary for the interrupt to occur are shown in the flowchart of **Figure 10-29**.

SETUP transactions cannot be stalled by the USB function. A SETUP received by a control endpoint will clear the STALL0 bit if it is set. The conditions for receiving

a SETUP interrupt are shown in **Figure 10-30**.

10.6.1.2 Transmit Control Endpoint 0

For a Control IN transaction directed at Endpoint 0, the USB module will generate an interrupt by setting the TXD0F flag in the UIR0 register. The conditions necessary for the interrupt to occur are shown in the flowchart of **Figure 10-31**.

10.6.1.3 Transmit Endpoint 1 and Transmit Endpoint 2

Transmit Endpoints 1 & 2 share their interrupt flag. For an IN transaction directed at Endpoint 1 or 2, the USB module will generate an interrupt by setting the TXD1F flag in the UIR1 register. The conditions necessary for the interrupt to occur are shown in the flowchart of **Figure 10-32**.

10.6.2 Resume Interrupt

The USB module will generate a USB interrupt if low speed bus activity is detected after entering the suspend state. A transition of the USB data lines to the non-idle state (“K” state) while in the suspend mode will set the RESUMF flag in the UIR1 register. There is no interrupt enable bit for this interrupt source and an interrupt will be executed if the I bit in the CCR is cleared. A resume interrupt can only occur while the MC68HC05JB4 is in the suspend mode.

10.6.3 End of Packet Interrupt

The USB module can generate a USB interrupt upon detection of an end of packet signal (a single ended 0) for low speed devices. Upon detection of an SE0 sequence, the USB module sets the EOPF bit and will generate an interrupt if the EOPIE bit in the UIR1 register is set.

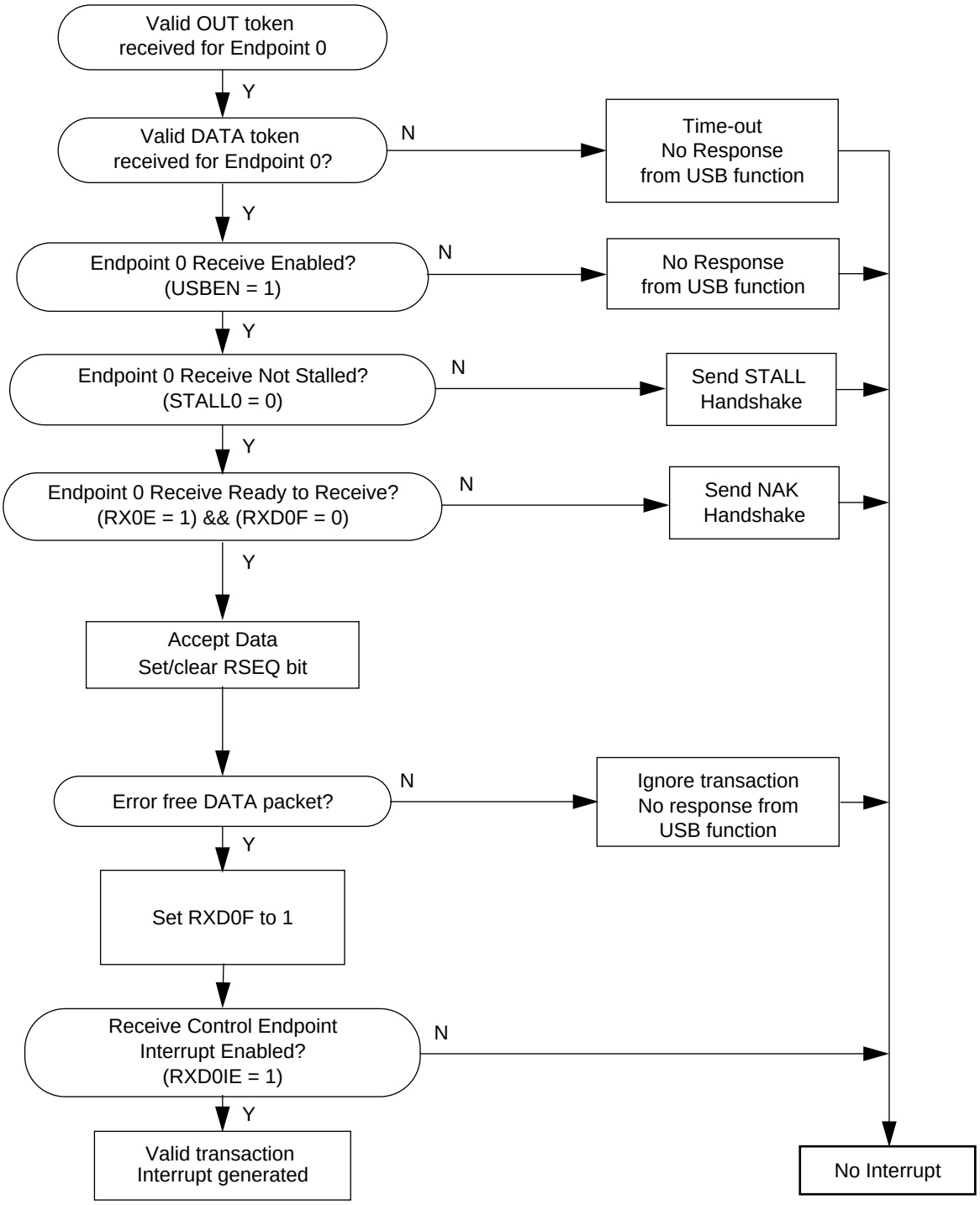


Figure 10-29. OUT Token Data Flow for Receive Endpoint 0

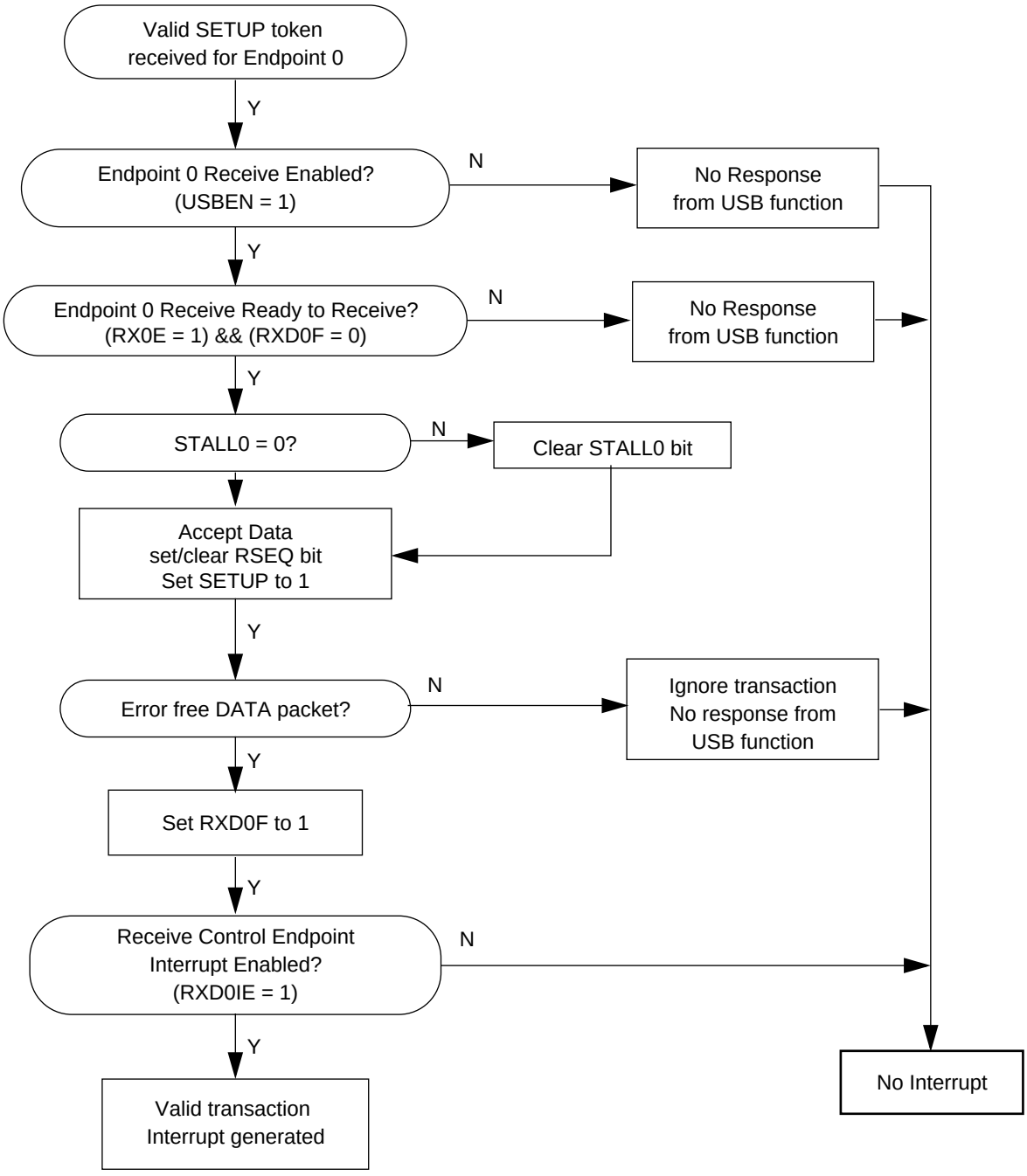


Figure 10-30. SETUP Token Data Flow for Receive Endpoint 0

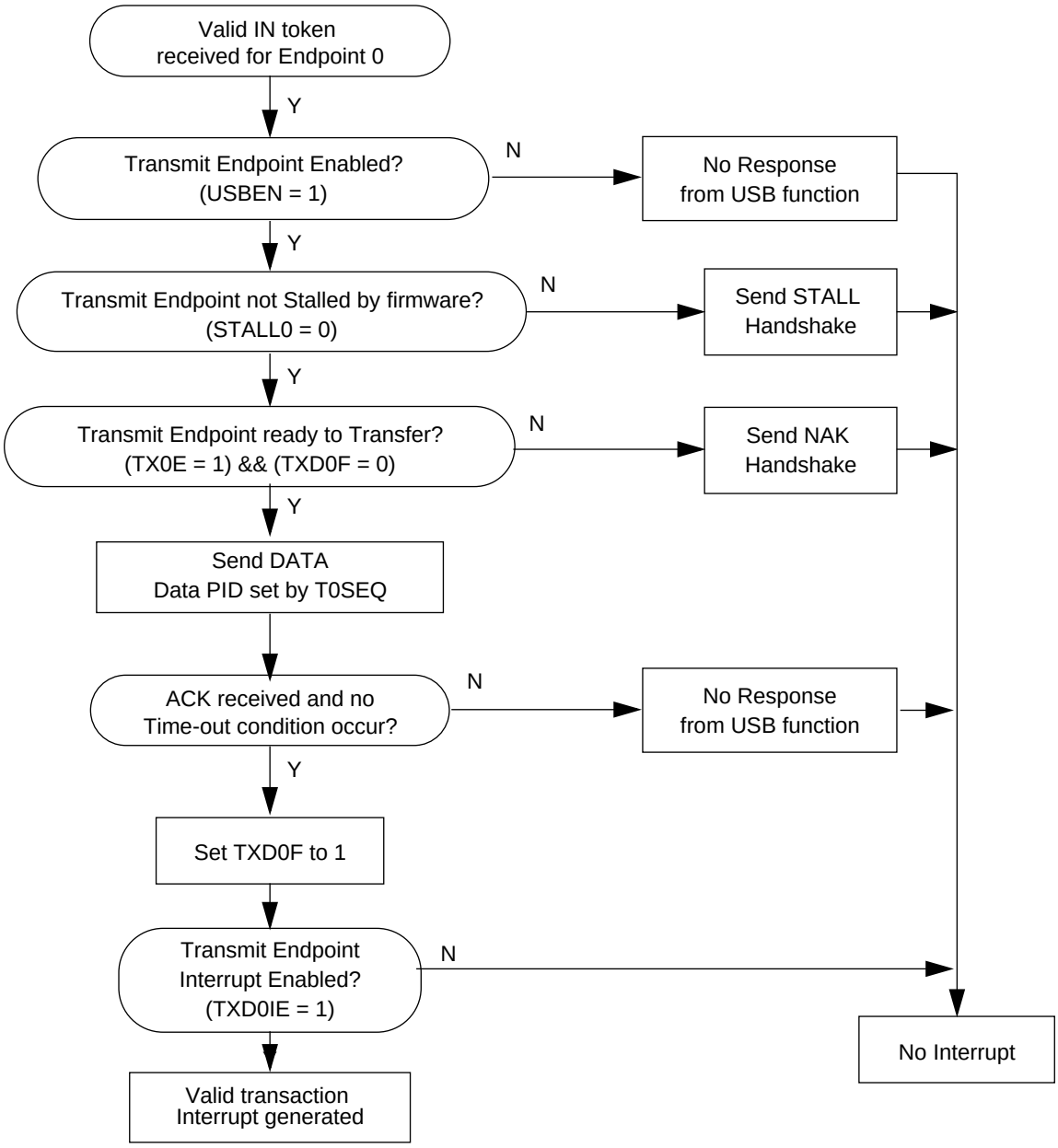


Figure 10-31. IN Token Data Flow for Transmit Endpoint 0

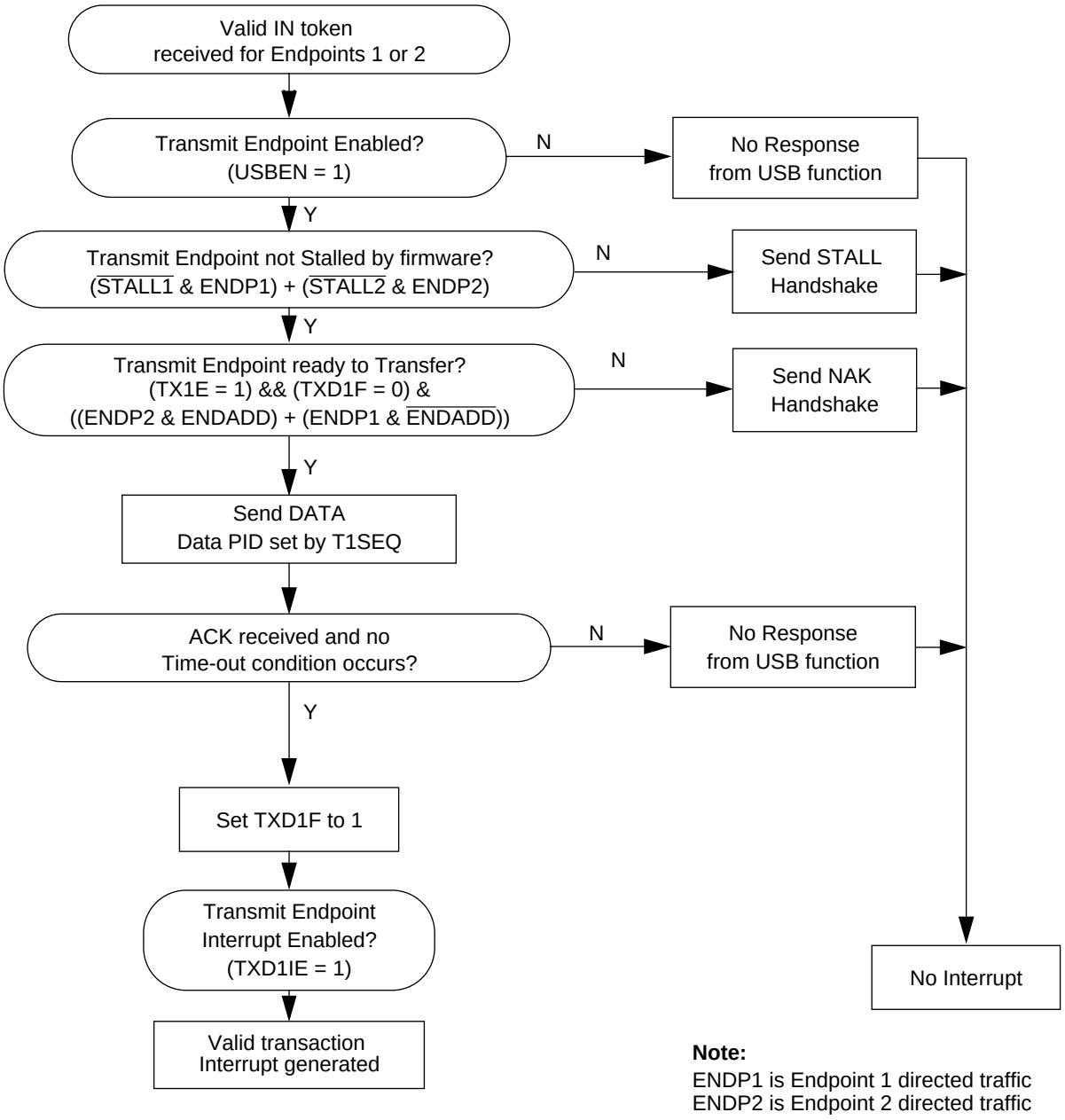


Figure 10-32. IN Token Data Flow for Transmit Endpoint 1/2

SECTION 11
ANALOG TO DIGITAL CONVERTER

The analog to digital converter system consists of a single 8-bit successive approximation converter and an 16-channel analog multiplexer. Six of the channels are available for analog inputs, four channels are dedicated to internal test functions, and the remaining six channels are unused. There is one 8-bit ADC Data Register ($0F) and one 8-bit ADC Status and Control register ($0E). The reference supply, V_{RL} and V_{RH} for the converter uses two input pins (shared with port pins PC4 and PC5) instead of the power supply lines, because drops caused by loading in the power supply lines would degrade the accuracy of the analog to digital conversion. An internal RC oscillator is available if the bus speed is low enough to degrade the ADC accuracy. An ADON bit allows the ADC to be switched off to reduce power consumption, which is particularly useful in the Wait mode.

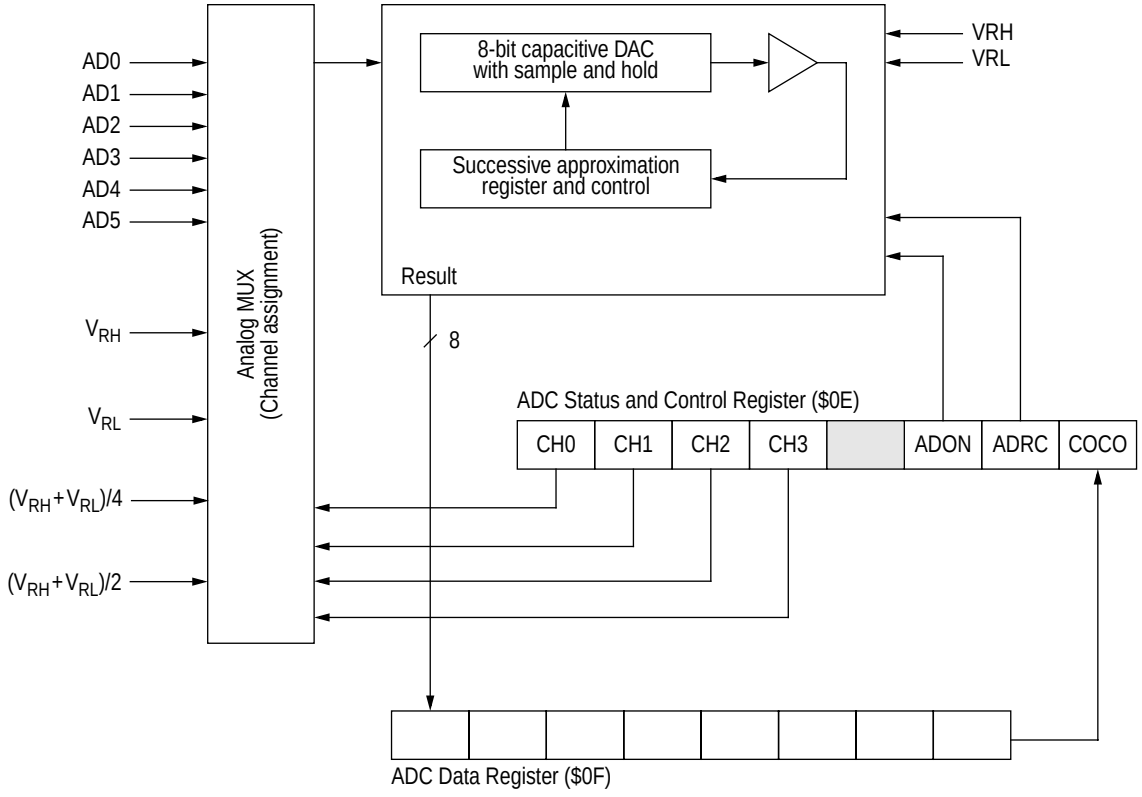


Figure 11-1. ADC Converter Block Diagram

11.1 ADC OPERATION

As shown in **Figure 11-1**, the ADC consists of an analog multiplexer, an 8-bit digital to analog capacitor array, a comparator and a successive approximation register (SAR).

There are ten options that can be selected by the multiplexer; the AD0 to AD5 input pins, V_{RH}, V_{RL}, $(V_{RH}+V_{RL})/4$, or $(V_{RH}+V_{RL})/2$. Selection is done via the CHx bits in the ADC Status and Control Register. AD0 to AD5 are input points for ADC conversion operations; the others are reference points which can be used for test purposes. The converter uses V_{RH} and V_{RL} as reference voltages. An input voltage equal to or greater than V_{RH} converts to $FF. An input voltage equal to or less than V_{RL}, but greater than V_{SS}, converts to $00. Maximum and minimum ratings must not be exceeded. Each analog input source should use V_{RH} as the supply voltage and should be referenced to V_{RL} for the ratiometric conversions. To maintain full accuracy of the ADC, the following should be noted:

1. V_{RH} should be equal to or less than V_{CC};
2. V_{RL} should be equal to or greater than V_{SS} but less than maximum specifications; and
3. $V_{RH}–V_{RL}$ should be equal to or greater than 4 Volts.

The ADC reference inputs (V_{RH} and V_{RL}) are applied to a precision internal digital to analog converter. Control logic drives this D/A converter and the analog output is successively compared with the selected analog input sampled at the beginning of the conversion. The conversion is monotonic with no missing codes.

The result of each successive comparison is stored in the successive approximation register (SAR) and, when the conversion is complete, the contents of the SAR are transferred to the read-only ADC Data Register ($0F), and the conversion complete flag, COCO, is set in the ADC Status and Control Register ($0E).

NOTE

Any write to the ADC Status and Control Register will abort the current conversion, reset the conversion complete flag (COCO) and a new conversion starts on the selected channel.

At power-on or external reset, both the ADRC and ADON bits are cleared, thus the ADC is disabled.

11.2 ADC STATUS AND CONTROL REGISTER (ADSCR)

The ADSCR is a read/write register containing status and control bits for the ADC.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ADSCR	R	COCO	ADRC	ADON	0	CH3	CH2	CH1	CH0
$000E	W								
reset:		0	0	0	0	0	0	0	0

Figure 11-2. A/D Status and Control Register

COCO — COnversion COmplete

1 = An ADC conversion has completed; ADC Data Register ($0F) contains valid conversion result.
0 = ADC conversion not completed.

This read-only status bit is set when a conversion is completed, indicating that the ADC Data Register contains a valid result. This COCO bit is cleared either by a write to the ADSCR or a read of the ADC Data Register. Once the COCO bit is cleared, a new conversion automatically starts. If the COCO bit is not cleared, conversions are initiated every 32 cycles. In this continuous conversion mode the ADC Data Register is refreshed with new data, every 32 cycles, and the COCO bit remains set.

ADRC — ADC RC Oscillator Control

1 = ADC uses RC oscillator as clock source.
0 = ADC uses internal processor clock as clock source.

The RC oscillator option must be used if the internal processor is running below 1MHz. A stabilization time of typically 1ms is required when switching to the RC oscillator option.

ADON — ADC On

1 = ADC is switched ON.
0 = ADC is switched OFF.

When the ADC is turned from OFF to ON, it requires a time t_{ADON} for the current sources to stabilize. During this time ADC conversion results may be inaccurate. Switching the ADC off disables the internal charge pump and RC oscillator (if selected by ADRC=1), and hence saving power.

CH3:CH0 — Channel Select Bits

These four bits selects one of ten ADC channels for the conversion. Channels 0 to 5 correspond to inputs AD0-AD5 on port pins PC0-PC3, PB3 and PB4 respectively. Channels 12 and 13 are the ADC reference inputs V_{RH} and V_{RL}, on port pins PC4 and PC5 respectively. Channels 14 and 15 are used for internal reference points. **Table 11-1** shows the signals selected by the channel select bits.

Table 11-1. A/D Channel Assignments

CH3	CH2	CH1	CH0	Channel	Signal
0	0	0	0	0	AD0 (PC0)
0	0	0	1	1	AD1 (PC1)
0	0	1	0	2	AD2 (PC2)
0	0	1	1	3	AD3 (PC3)
0	1	0	0	4	AD4 (PB3)
0	1	0	1	5	AD5 (PB4)
0	1	1	0	6	Not Implemented
0	1	1	1	7	Not Implemented
1	0	0	0	8	Not Implemented
1	0	0	1	9	Not Implemented
1	0	1	0	10	Not Implemented
1	0	1	1	11	Not Implemented
1	1	0	0	12	VRH (PC4)
1	1	0	1	13	VRL (PC5)
1	1	1	0	14	(VRH+VRL)/4
1	1	1	1	15	(VRH+VRL)/2

Using a port pin as both an analog and digital input simultaneously is prohibited. When the ADC is enabled (ADON=1) and one of channels is selected, the corresponding port pin will appear as a logic zero when read from the Port Data Register. Note that the pull-up on a port-B pin will be disabled automatically when that port pin is selected as the ADC input and the ADON bit is set to logic “1”.

11.3 ADC DATA REGISTER (ADDR)

The ADDR stores the result of a valid ADC conversion when the COCO bits is set in ADSCR.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ADDR	R	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
$000F	W								
reset:		0	0	0	0	0	0	0	0

Figure 11-3. A/D Data Register

11.4 ADC DURING LOW POWER MODES

The ADC continues normal operation in WAIT mode. To reduce power consumption in WAIT mode, the ADON and ADRC bits in the ADSCR should be cleared if the ADC is not used. If the ADC is in use and the internal bus clock is above 1MHz, it is recommended that the ADRC bit be cleared.

In STOP mode, the ADC stops operation.

SECTION 12
INSTRUCTION SET

This section describes the addressing modes and instruction types.

12.1 ADDRESSING MODES

The CPU uses eight addressing modes for flexibility in accessing data. The addressing modes define the manner in which the CPU finds the data required to execute an instruction. The eight addressing modes are the following:

- Inherent
- Immediate
- Direct
- Extended
- Indexed, No Offset
- Indexed, 8-Bit Offset
- Indexed, 16-Bit Offset
- Relative

12.1.1 Inherent

Inherent instructions are those that have no operand, such as return from interrupt (RTI) and stop (STOP). Some of the inherent instructions act on data in the CPU registers, such as set carry flag (SEC) and increment accumulator (INCA). Inherent instructions require no memory address and are one byte long.

12.1.2 Immediate

Immediate instructions are those that contain a value to be used in an operation with the value in the accumulator or index register. Immediate instructions require no memory address and are two bytes long. The opcode is the first byte, and the immediate data value is the second byte.

12.1.3 Direct

Direct instructions can access any of the first 256 memory addresses with two bytes. The first byte is the opcode, and the second is the low byte of the operand address. In direct addressing, the CPU automatically uses $00 as the high byte of the operand address. BRSET and BRCLR are three-byte instructions that use direct addressing to access the operand and relative addressing to specify a branch destination.

12.1.4 Extended

Extended instructions use only three bytes to access any address in memory. The first byte is the opcode; the second and third bytes are the high and low bytes of the operand address.

When using the Motorola assembler, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.

12.1.5 Indexed, No Offset

Indexed instructions with no offset are one-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the conditional address of the operand. The CPU automatically uses $00 as the high byte, so these instructions can address locations $0000–$00FF.

Indexed, no offset instructions are often used to move a pointer through a table or to hold the address of a frequently used RAM or I/O location.

12.1.6 Indexed, 8-Bit Offset

Indexed, 8-bit offset instructions are two-byte instructions that can access data with variable addresses within the first 511 memory locations. The CPU adds the unsigned byte in the index register to the unsigned byte following the opcode. The sum is the conditional address of the operand. These instructions can access locations $0000–$01FE.

Indexed 8-bit offset instructions are useful for selecting the kth element in an n-element table. The table can begin anywhere within the first 256 memory locations and could extend as far as location 510 ($01FE). The k value is typically in the index register, and the address of the beginning of the table is in the byte following the opcode.

12.1.7 Indexed, 16-Bit Offset

Indexed, 16-bit offset instructions are three-byte instructions that can access data with variable addresses at any location in memory. The CPU adds the unsigned byte in the index register to the two unsigned bytes following the opcode. The sum is the conditional address of the operand. The first byte after the opcode is the high byte of the 16-bit offset; the second byte is the low byte of the offset. These instructions can address any location in memory.

Indexed, 16-bit offset instructions are useful for selecting the kth element in an n-element table anywhere in memory.

As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

12.1.8 Relative

Relative addressing is only for branch instructions. If the branch condition is true, the CPU finds the conditional branch destination by adding the signed byte following the opcode to the contents of the program counter. If the branch condition is not true, the CPU goes to the next instruction. The offset is a signed, two's complement byte that gives a branching range of –128 to +127 bytes from the address of the next location after the branch instruction.

When using the Motorola assembler, the programmer does not need to calculate the offset, because the assembler determines the proper offset and verifies that it is within the span of the branch.

12.1.9 Instruction Types

The MCU instructions fall into the following five categories:

- Register/Memory Instructions
- Read-Modify-Write Instructions
- Jump/Branch Instructions
- Bit Manipulation Instructions
- Control Instructions

12.1.10 Register/Memory Instructions

Most of these instructions use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory. **Table 12-1** lists the register/memory instructions.

Table 12-1. Register/Memory Instructions

Instruction	Mnemonic
Add Memory Byte and Carry Bit to Accumulator	ADC
Add Memory Byte to Accumulator	ADD
AND Memory Byte with Accumulator	AND
Bit Test Accumulator	BIT
Compare Accumulator	CMP
Compare Index Register with Memory Byte	CPX
EXCLUSIVE OR Accumulator with Memory Byte	EOR
Load Accumulator with Memory Byte	LDA
Load Index Register with Memory Byte	LDX
Multiply	MUL
OR Accumulator with Memory Byte	ORA
Subtract Memory Byte and Carry Bit from Accumulator	SBC
Store Accumulator in Memory	STA
Store Index Register in Memory	STX
Subtract Memory Byte from Accumulator	SUB

12.1.11 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to the memory location or to the register. The test for negative or zero instruction (TST) is an exception to the read-modify-write sequence because it does not write a replacement value. **Table 12-2** lists the read-modify-write instructions.

Table 12-2. Read-Modify-Write Instructions

Instruction	Mnemonic
Arithmetic Shift Left	ASL
Arithmetic Shift Right	ASR
Clear Bit in Memory	BCLR
Set Bit in Memory	BSET
Clear	CLR
Complement (One's Complement)	COM
Decrement	DEC
Increment	INC
Logical Shift Left	LSL
Logical Shift Right	LSR
Negate (Two's Complement)	NEG
Rotate Left through Carry Bit	ROL
Rotate Right through Carry Bit	ROR
Test for Negative or Zero	TST

12.1.12 Jump/Branch Instructions

Jump instructions allow the CPU to interrupt the normal sequence of the program counter. The unconditional jump instruction (JMP) and the jump to subroutine instruction (JSR) have no register operand. Branch instructions allow the CPU to interrupt the normal sequence of the program counter when a test condition is met. If the test condition is not met, the branch is not performed. All branch instructions use relative addressing.

Bit test and branch instructions cause a branch based on the state of any readable bit in the first 256 memory locations. These three-byte instructions use a combination of direct addressing and relative addressing. The direct address of the byte to be tested is in the byte following the opcode. The third byte is the signed offset byte. The CPU finds the conditional branch destination by adding the

third byte to the program counter if the specified bit tests true. The bit to be tested and its condition (set or clear) is part of the opcode. The span of branching is from –128 to +127 from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register. **Table 12-3** lists the jump and branch instructions.

Table 12-3. Jump and Branch Instructions

Instruction	Mnemonic
Branch if Carry Bit Clear	BCC
Branch if Carry Bit Set	BCS
Branch if Equal	BEQ
Branch if Half-Carry Bit Clear	BHCC
Branch if Half-Carry Bit Set	BHCS
Branch if Higher	BHI
Branch if Higher or Same	BHS
Branch if $\overline{\text{IRQ}}$ Pin High	BIH
Branch if $\overline{\text{IRQ}}$ Pin Low	BIL
Branch if Lower	BLO
Branch if Lower or Same	BLS
Branch if Interrupt Mask Clear	BMC
Branch if Minus	BMI
Branch if Interrupt Mask Set	BMS
Branch if Not Equal	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit Clear	BRCLR
Branch Never	BRN
Branch if Bit Set	BRSET
Branch to Subroutine	BSR
Unconditional Jump	JMP
Jump to Subroutine	JSR

12.1.13 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory. Port registers, port data direction registers, timer registers, and on-chip RAM locations are in the first 256 bytes of memory. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations. Bit manipulation instructions use direct addressing. **Table 12-4** lists these instructions.

Table 12-4. Bit Manipulation Instructions

Instruction	Mnemonic
Clear Bit	BCLR
Branch if Bit Clear	BRCLR
Branch if Bit Set	BRSET
Set Bit	BSET

12.1.14 Control Instructions

These register reference instructions control CPU operation during program execution. Control instructions, listed in **Table 12-5**, use inherent addressing.

Table 12-5. Control Instructions

Instruction	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
No Operation	NOP
Reset Stack Pointer	RSP
Return from Interrupt	RTI
Return from Subroutine	RTS
Set Carry Bit	SEC
Set Interrupt Mask	SEI
Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin	STOP
Software Interrupt	SWI
Transfer Accumulator to Index Register	TAX
Transfer Index Register to Accumulator	TXA
Stop CPU Clock and Enable Interrupts	WAIT

12.1.15 Instruction Set Summary

Table 12-6 is an alphabetical list of all M68HC05 instructions and shows the effect of each instruction on the condition code register.

Table 12-6. Instruction Set Summary

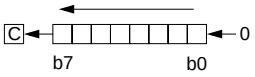
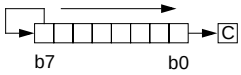
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X	Add without Carry	A ← (A) + (M)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	AB BB CB DB EB FB	ii dd hh ll ee ff ff	2 3 4 5 4 3
AND #*opr* AND *opr* AN*D opr* AND *opr*,X AND *opr*,X AND ,X	Logical AND	A ← (A) ∧ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A4 B4 C4 D4 E4 F4	ii dd hh ll ee ff ff	2 3 4 5 4 3
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X	Arithmetic Shift Left (Same as LSL)		—	—	↕	↕	↕	↕	38 48 58 68 78	dd ff	5 3 3 6 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR ,X	Arithmetic Shift Right		—	—	↕	↕	↕	DIR INH INH IX1 IX	37 47 57 67 77	dd ff	5 3 3 6 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BCLR *n opr*	Clear Bit n	Mn ← 0	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? Z = 1	—	—	—	—	—	REL	27	rr	3
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BIH *rel*	Branch if $\overline{\text{IRQ}}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if $\overline{\text{IRQ}}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 0	—	—	—	—	—	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X	Bit Test Accumulator with Memory Byte	(A) ∧ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A5 B5 C5 D5 E5 F5	ii dd hh ll ee ff ff p	2 3 4 5 4 3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? C ∨ Z = 1	—	—	—	—	—	REL	23	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? I = 0	—	—	—	—	—	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? N = 1	—	—	—	—	—	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? I = 1	—	—	—	—	—	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? Z = 0	—	—	—	—	—	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? N = 0	—	—	—	—	—	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel* ? 1 = 1	—	—	—	—	—	REL	20	rr	3
BRCLR *n opr rel*	Branch if bit n clear	PC ← (PC) + 2 + *rel* ? Mn = 0	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRSET *n opr rel*	Branch if Bit n Set	PC ← (PC) + 2 + *rel* ? Mn = 1	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BSET *n opr*	Set Bit n	Mn ← 1	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	—	—	—	—	—	REL	AD	rr	6
CLC	Clear Carry Bit	C ← 0	—	—	—	—	0	INH	98		2
CLI	Clear Interrupt Mask	I ← 0	—	0	—	—	—	INH	9A		2
CLR *opr* CLRA CLRX CLR *opr*,X CLR ,X	Clear Byte	M ← $00 A ← $00 X ← $00 M ← $00 M ← $00	—	—	0	1	—	DIR INH INH IX1 IX	3F 4F 5F 6F 7F	dd ff 	5 3 3 6 5
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X	Compare Accumulator with Memory Byte	(A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A1 B1 C1 D1 E1 F1	ii dd hh ll ee ff ff 	2 3 4 5 4 3
COM *opr* COMA COMX COM *opr*,X COM ,X	Complement Byte (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (M) X ← ($\overline{X}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	—	—	↕	↕	1	DIR INH INH IX1 IX	33 43 53 63 73	dd ff 	5 3 3 6 5
CPX #*opr* CPX *opr* CPX *opr* CPX *opr*,X CPX *opr*,X CPX ,X	Compare Index Register with Memory Byte	(X) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A3 B3 C3 D3 E3 F3	ii dd hh ll ee ff ff 	2 3 4 5 4 3
DEC *opr* DECA DECX DEC *opr*,X DEC ,X	Decrement Byte	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1	—	—	↕	↕	—	DIR INH INH IX1 IX	3A 4A 5A 6A 7A	dd ff 	5 3 3 6 5
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X	EXCLUSIVE OR Accumulator with Memory Byte	A ← (A) ⊕ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A8 B8 C8 D8 E8 F8	ii dd hh ll ee ff ff 	2 3 4 5 4 3

Table 12-6. Instruction Set Summary (Continued)

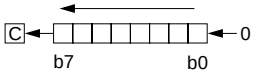
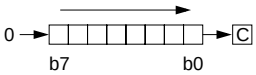
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
INC *opr* INCA INCX INC *opr*,X INC ,X	Increment Byte	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1	—	—	↕	↕	—	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff 	5 3 3 6 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Unconditional 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 *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Conditional Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff 	5 6 7 6 5
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X	Load Accumulator with Memory Byte	A ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff 	2 3 4 5 4 3
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X	Load Index Register with Memory Byte	X ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff 	2 3 4 5 4 3
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X	Logical Shift Left (Same as ASL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
LSR *opr* LSRA LSRX LSR *opr*,X LSR ,X	Logical Shift Right		—	—	0	↕	↕	DIR INH INH IX1 IX	34 44 54 64 74	dd ff 	5 3 3 6 5
MUL	Unsigned Multiply	X : A ← (X) × (A)	0	—	—	—	0	INH	42		11
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (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	30 40 50 60 70	ii ff 	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2

Table 12-6. Instruction Set Summary (Continued)

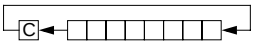
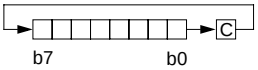
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	39 49 59 69 79	dd ff	5 3 3 6 5
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	36 46 56 66 76	dd ff	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	INH	80		9
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	—	—	—	—	—	INH	81		6
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X	Store Accumulator in Memory	M ← (A)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff	4 5 6 5 4
STOP	Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin		—	0	—	—	—	INH	8E		2
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X	Store Index Register In Memory	M ← (X)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff	4 5 6 5 4

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X	Subtract Memory Byte from Accumulator	A ← (A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff	2 3 4 5 4 3
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		10
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2
TST *opr* TSTA TSTX TST *opr*,X TST ,X	Test Memory Byte for Negative or Zero	(M) – $00	—	—	↕	↕	—	DIR INH INH IX1 IX	3D 4D 5D 6D 7D	dd ff	4 3 3 5 4
TXA	Transfer Index Register to Accumulator	A ← (X)	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	0	—	—	—	INH	8F		2

A	Accumulator	*opr*	Operand (one or two bytes)
C	Carry/borrow flag	PC	Program counter
CCR	Condition code register	PCH	Program counter high byte
dd	Direct address of operand	PCL	Program counter low byte
dd rr	Direct address of operand and relative offset of branch instruction	REL	Relative addressing mode
DIR	Direct addressing mode	*rel*	Relative program counter offset byte
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	rr	Relative program counter offset byte
EXT	Extended addressing mode	SP	Stack pointer
ff	Offset byte in indexed, 8-bit offset addressing	X	Index register
H	Half-carry flag	Z	Zero flag
hh ll	High and low bytes of operand address in extended addressing	#	Immediate value
I	Interrupt mask	∧	Logical AND
ii	Immediate operand byte	∨	Logical OR
IMM	Immediate addressing mode	⊕	Logical EXCLUSIVE OR
INH	Inherent addressing mode	( )	Contents of
IX	Indexed, no offset addressing mode	–( )	Negation (two's complement)
IX1	Indexed, 8-bit offset addressing mode	←	Loaded with
IX2	Indexed, 16-bit offset addressing mode	?	If
M	Memory location	:	Concatenated with
N	Negative flag	↕	Set or cleared
n	Any bit	—	Not affected

Table 12-7. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	DIR	DIR	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
MSB / LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	MSB / LSB
0	BRSET0 5 3 DIR	BSET0 5 2 DIR	BRA 3 2 REL	NEG 5 2 DIR	NEGA 3 1 INH	NEGX 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0
1	BRCLR0 5 3 DIR	BCLR0 5 2 DIR	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1
2	BRSET1 5 3 DIR	BSET1 5 2 DIR	BHI 3 2 REL		MUL 11 1 INH						SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2
3	BRCLR1 5 3 DIR	BCLR1 5 2 DIR	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3
4	BRSET2 5 3 DIR	BSET2 5 2 DIR	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4
5	BRCLR2 5 3 DIR	BCLR2 5 2 DIR	BCS/BLO 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5
6	BRSET3 5 3 DIR	BSET3 5 2 DIR	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6
7	BRCLR3 5 3 DIR	BCLR3 5 2 DIR	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7
8	BRSET4 5 3 DIR	BSET4 5 2 DIR	BHCC 3 2 REL	ASL/LSL 5 2 DIR	ASLA/LSLA 3 1 INH	ASLX/LSLX 3 1 INH	ASL/LSL 6 2 IX1	ASL/LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8
9	BRCLR4 5 3 DIR	BCLR4 5 2 DIR	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9
A	BRSET5 5 3 DIR	BSET5 5 2 DIR	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A
B	BRCLR5 5 3 DIR	BCLR5 5 2 DIR	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B
C	BRSET6 5 3 DIR	BSET6 5 2 DIR	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C
D	BRCLR6 5 3 DIR	BCLR6 5 2 DIR	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D
E	BRSET7 5 3 DIR	BSET7 5 2 DIR	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E
F	BRCLR7 5 3 DIR	BCLR7 5 2 DIR	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F

INH = Inherent
IMM = Immediate
DIR = Direct
EXT = Extended
REL = Relative
IX = Indexed, No Offset
IX1 = Indexed, 8-Bit Offset
IX2 = Indexed, 16-Bit Offset

LSB of Opcode in Hexadecimal

MSB / LSB	0
0	5 BRSET0 3 DIR

MSB of Opcode in Hexadecimal
Number of Cycles
Opcode Mnemonic
Number of Bytes/Addressing Mode

SECTION 13
ELECTRICAL SPECIFICATIONS

This section describes the electrical and timing specifications of the MC68HC05JB4.

13.1 MAXIMUM RATINGS

(Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	–0.3 to +7.0	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Input Voltage	V_{IN}	V_{SS} – 0.3 to V_{SS} + 0.3	V
$\overline{IRQ}$/V_{PP} Pin	V_{PP}	V_{SS} –0.3 to 17	V
Storage Temperature Range	T_{STG}	–65 to +150	°C

NOTE

Maximum ratings are the extreme limits the device can be exposed to without causing permanent damage to the chip. The device is **not** intended to operate at these conditions.

The MCU contains circuitry that protect the inputs against damage from high static voltages; however, do not apply voltages higher than those shown in the table below. Keep V_{IN} and V_{OUT} within the range from $V_{SS} \le (V_{IN}$ or $V_{OUT}) \le V_{DD}$. Connect unused inputs to the appropriate voltage level, either V_{SS} or V_{DD}.

13.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance			
28-pin PDIP	θ_{JA}	60	°C/W
28-pin SOIC	θ_{JA}	60	°C/W

13.3 DC ELECTRICAL CHARACTERISTICS

Table 13-1. DC Electrical Characteristics

(V_{DD} = 4.2V to 5.5V, V_{SS} = 0 Vdc, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage I_{Load} = 10.0 µA	V_{OL} V_{OH}	— V_{DD}–0.1	— —	0.1 —	V
Output High Voltage (I_{Load} =–0.8 mA) PA0-7, PB0-4, PC0-5	V_{OH}	V_{DD}–0.8	—	—	V
Output Low Voltage (I_{Load} = 1.6 mA) PA0-4, PB0-4, PC0-5 (I_{Load} = 10.0 mA) PA5-7 (I_{Load} = 25.0 mA) PA6, PA7 (mask option)	V_{OL}	— — —	— — —	0.4 0.4 0.5	V
Input High Voltage PA0-7, PB0-4, PC0-5, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7×V_{DD}	—	V_{DD}	V
Input Low Voltage PA0-7, PB0-4, PC0-5, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2×V_{DD}	V
Supply Current (see Notes) Run (USB active) Run (USB suspended) Wait (USB active) Wait (USB suspended) Stop LVR off at 25°C (not inlcude 15k to GND)	I_{DD}	 — — — — —	 4.0 3.5 1.5 1.0 130	 4.5 4.0 2.0 1.5 180	 mA mA mA mA µA
I/O Ports Hi-Z Leakage Current PA0-7, PB0-4, PC0-5 (without individual pulldown/up activated)	I_Z	—	—	±10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1	I_{in}	—	—	5	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, OSC1, OSC2	C_{out} C_{in}	— —	— —	12 8	pF pF
Crystal/Ceramic Resonator Oscillator Mode Internal Resistor OSC1 to OSC2	R_{OSC}	1.0	2.0	3.0	MΩ
Pullup Resistor PA0-7, PB0-4	R_{PULLUP}	30	50	75	KΩ
LVR Inhibit	V_{LVRI}	—	3.3	—	V
LVR Recover	V_{LVRR}	—	3.5	—	V

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only MFT and Timer1 active.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source to OSC1 (f_{OSC} = 6.0 MHz), all inputs 0.2 VDC from rail; no DC loads, less than 50pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 VDC, V_{IH} = V_{DD}–0.2 VDC.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

13.4 USB DC ELECTRICAL CHARACTERISTICS

Table 13-2. USB DC Electrical Characteristics

(V_{DD} = 4.2V to 5.5V, V_{SS} = 0 Vdc, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Conditions	Min	Typ	Max	Unit
Hi-Z State Data Line Leakage	I_{LO}	0V<Vin<3.3V	–10		+10	µA
Differential Input Sensitivity	V_{DI}	|(D+)–(D–)|	0.2			V
Differential Common Mode Range	V_{CM}	Includes V_{DI} range	0.8		2.5	V
Single Ended Receiver Threshold	V_{SE}		0.8		2.0	V
Static Output Low	V_{OL}	R_L of 1.5k to 3.6V			0.3	V
Static Output High	V_{OH}	R_L of 15k to GND	2.8		3.6	V
3.3V External Reference Pin	$V_{3.3}$	I_L=200µA	3.0	3.3	3.6	V

13.5 USB LOW SPEED SOURCE ELECTRICAL CHARACTERISTICS

Table 13-3. USB Low Speed Source Electrical Characteristics

Parameter	Symbol	Conditions (Notes 1,2,3)	Min	Typ	Max	Unit
Transition time: Rise Time Fall Time	 T_R T_F	Notes 4, 5, 8 C_L=50pF C_L=350pF C_L=50pF C_L=350pF	 75 75	 — —	 — 300 — 300	 ns ns ns ns
Rise/Fall Time Matching	T_{RFM}	T_R/T_F	80	—	120	%
Output Signal Crossover Voltage	V_{CRS}		1.3	—	2.0	V
Low Speed Data Rate	T_{DRATE}	1.5Mbs ±1.5%	1.4775 676.8	1.500 666.0	1.5225 656.8	Mbs ns
Source Differential Driver Jitter To Next Transition For Paired Transitions	 T_{UDJ1} T_{UDJ2}	C_L=350pF Notes 6 and 7	 –25 –10	 — —	 25 10	 ns ns
Receiver Data Jitter Tolerance To Next Transition For Paired Transitions	 T_{DJR1} T_{DJR2}	C_L=350pF Notes 7	 –75 –45	 — —	 75 45	 ns ns
Source EOP Width	TEOPT	Note 7	1.25	—	1.50	µs
Differential to EOP Transition Skew	TDEOP	Note 7	–40	—	100	ns
Receiver EOP Width Must Reject as EOP Must Accept	 T_{EOPR1} T_{EOPR2}	Note 7	 330 675	 — —	 — —	 ns ns

NOTES:

1. All voltages measured from local ground, unless otherwise specified.
2. All timings use a capacitive load of 50pF, unless otherwise specified.
3. Low speed timings have a 1.5k pull-up to 2.8V on the D– data line.
4. Measured from 10% to 90% of the data signal.
5. The rising and falling edges should be smooth transitions (monotonic).
6. Timing differences between the differential data signals.
7. Measured at crossover point of differential data signals.
8. Capacitive loading includes 50pF of tester capacitance.

13.6 CONTROL TIMING

Table 13-4. Control Timing

(V_{DD} = 4.2V to 5.5V, V_{SS} = 0 Vdc, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Units
Frequency of Operation Crystal Oscillator Option External Clock Source	 f_{OSC} f_{OSC}	 — DC	 6 6	 MHz MHz
Internal Operating Frequency Crystal Oscillator ($f_{OSC} \div 2$) External Clock ($f_{OSC} \div 2$)	 f_{OP} f_{OP}	 — DC	 3 3	 MHz MHz
Cycle Time ($1/f_{OP}$)	t_{CYC}	330	—	ns
$\overline{\text{RESET}}$ Pulse Width Low	t_{RL}	1.5	—	t_{CYC}
$\overline{\text{IRQ}}$ Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	0.5	—	t_{CYC}
$\overline{\text{IRQ}}$ Interrupt Pulse Period	t_{ILIL}	note 1	—	t_{CYC}
PA0 to PA3 Interrupt Pulse Width High (Edge-Triggered)	t_{IHIL}	0.5	—	t_{CYC}
PA0 to PA3 Interrupt Pulse Period	t_{IHIH}	note 1	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	—	—	ns
Output High to Low Transition Period on PA6, PA7, PB0-4	t_{SLOW}	0.5 (typical)		t_{CYC}

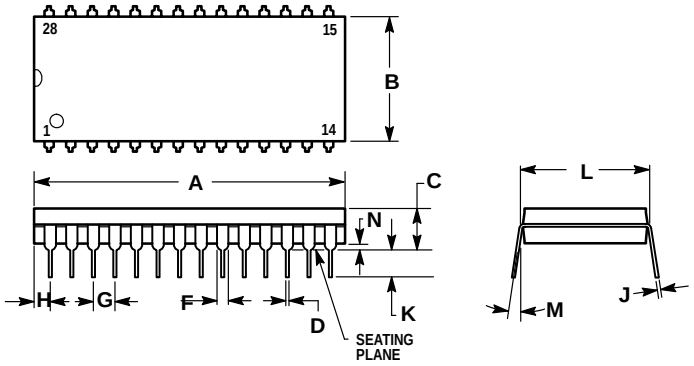
NOTES:

1. The minimum period t_{ILIL} or t_{IHIH} should not be less than the number of cycles it takes to execute the interrupt service routine plus 19 t_{CYC}.

SECTION 14
MECHANICAL SPECIFICATIONS

This section provides the mechanical dimensions for the 28-pin PDIP and 28-pin SOIC packages.

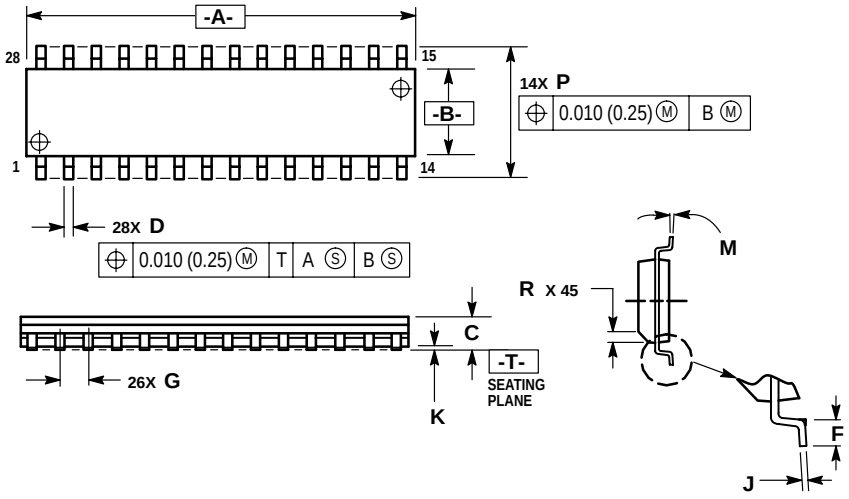
14.1 28-PIN PDIP (CASE 710)



NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

14.2 28-PIN SOIC (CASE 751F)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 (0.005) TOTAL IN EXCESS OF D DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	17.80	18.05	0.701	0.711
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.41	0.90	0.016	0.035
G	1.27 BSC		0.050 BSC	
J	0.23	0.32	0.009	0.013
K	0.13	0.29	0.005	0.011
M	0°	8°	0°	8°
P	10.01	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

APPENDIX A
MC68HC705JB4

This appendix describes the MC68HC705JB4, the emulation part for MC68HC05JB4. The entire MC68HC05JB4 data sheet applies to the MC68HC705JB4, with exceptions outlined in this appendix.

A.1 INTRODUCTION

The MC68HC705JB4 is an EPROM version of the MC68HC05JB4, and is available for user system evaluation and debugging. The MC68HC705JB4 is functionally identical to the MC68HC05JB4 with the exception of the 3584 bytes user ROM is replaced by 3584 bytes user EPROM. Also, the mask options available on the MC68HC05JB4 are implemented using the Mask Option Register (MOR) in the MC68HC705JB4.

A.2 MEMORY

The MC68HC705JB4 memory map is shown in **Figure A-1**.

A.3 MASK OPTION REGISTER (MOR)

The Mask Option Register (MOR) is a byte of EPROM used to select the features controlled by mask options on the MC68HC05JB4. In order to program this register the MORON bit in PCR need to be set to “1” before doing the EPROM programming process.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
MOR	R			COPEN	IRQTRIG	HIGHCURRA	PAINTEN	OSCDLY	LVREN
$007F	W								
reset:		0	0	1	1	1	1	1	1

COPEN – COP Enable

1 = COP watchdog function disabled.
0 = COP watchdog function enabled.

IRQTRIG – IRQ, PA0-PA3 Interrupt Options

1 = Edge-trigger only.
0 = Edge-and-level-triggered.

HIGHCURRA – PA6 and PA7 High Current Enable

1 = High current capability disabled on PA6 and PA7.
0 = High current capability enabled on PA6 and PA7.

PAINTEN – PA0-PA3 External Interrupt Options

1 = External interrupt capability on PA0-PA3 disabled.
0 = External interrupt capability on PA0-PA3 enabled.

OSCDLY – Oscillator Delay Option

1 = 128 internal clock cycles.
0 = 4064 internal clock cycles.

LVREN – LVR Option

1 = Low Voltage Reset circuit enabled.
0 = Low Voltage Reset circuit disabled.

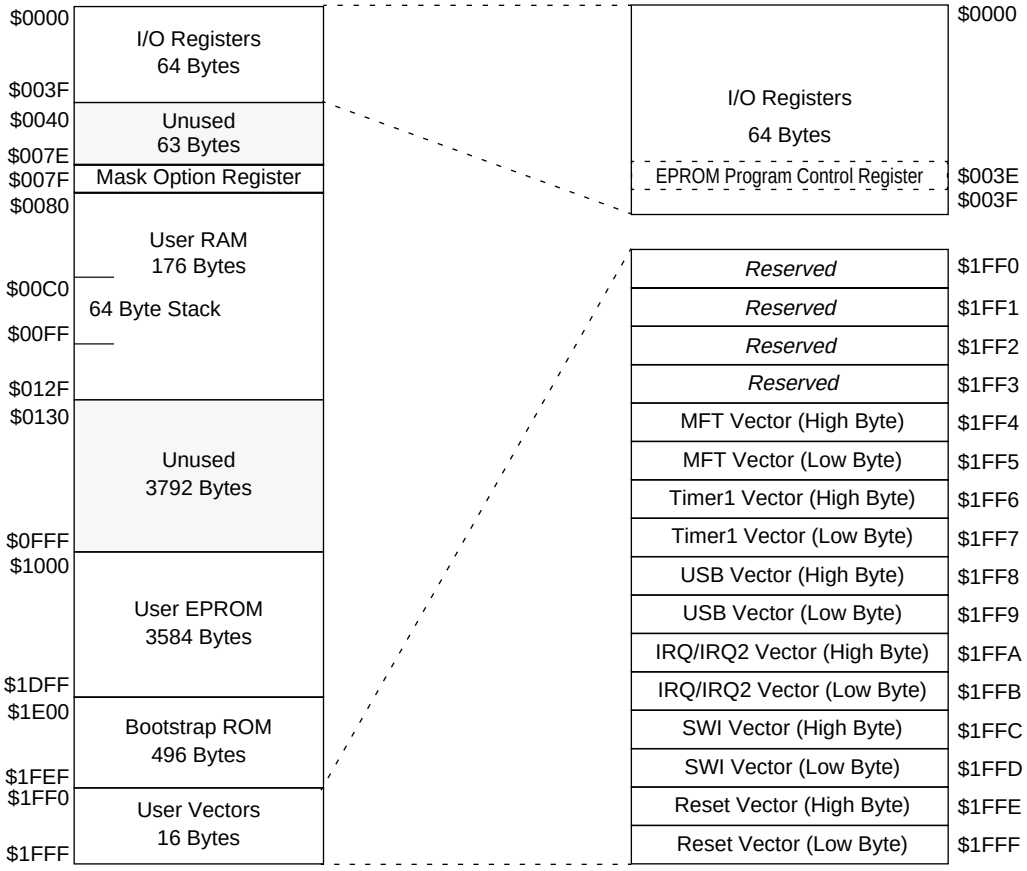


Figure A-1. MC68HC705JB4 Memory Map

A.4 BOOTSTRAP MODE

Bootloader mode is entered upon the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}/V_{PP}$ pin is at V_{TST} and the PB0 pin is at logic zero. The Bootloader program is masked in the ROM area from $1E00 to $1FEF. This program handles copying of user code from an external EPROM into the on-chip EPROM. The bootload function has to be done from an external EPROM. The bootloader performs one programming pass at 1ms per byte then does a verify pass.

The user code must be a one-to-one correspondence with the internal EPROM addresses.

A.5 EPROM PROGRAMMING

Programming the on-chip EPROM is achieved by using the Program Control Register located at address $3E.

Please contact Motorola for programming board availability.

A.5.1 EPROM Program Control Register (PCR)

This register is provided for programming the on-chip EPROM in the MC68HC705JB4.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PCR	R	RESERVED					MORON	ELAT	PGM
$003E	W								
reset:		0	0	0	0	0	0	0	0

MORON – Mask Option Register ON

0 = Disable programming to Mask Option Register ($007F)
1 = Enable programming to Mask Option Register ($007F)

ELAT – EPROM LATch control

0 = EPROM address and data bus configured for normal reads
1 = EPROM address and data bus configured for programming (writes to EPROM cause address and data to be latched). EPROM is in programming mode and cannot be read if ELAT is 1. This bit should not be set when no programming voltage is applied to the V_{pp} pin.

PGM – EPROM ProGraM command

0 = Programming power is switched OFF from EPROM array.
1 = Programming power is switched ON to EPROM array. If ELAT ≠ 1, then PGM = 0.

A.5.2 Programming Sequence

The EPROM programming sequence is:

1. Set the ELAT bit
2. Write the data to the address to be programmed
3. Set the PGM bit
4. Delay for a time t_{PGMR}
5. Clear the PGM bit
6. Clear the ELAT bit

The last two steps must be performed with separate CPU writes.

CAUTION

It is important to remember that an external programming voltage must be applied to the V_{PP} pin while programming, but it should be equal to V_{DD} during normal operations.

Figure A-2 shows the flow required to successfully program the EPROM.

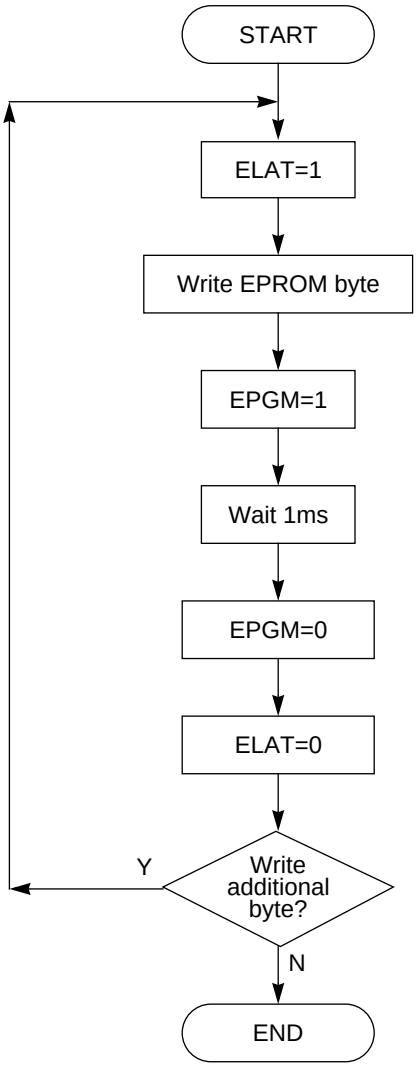


Figure A-2. EPROM Programming Sequence

A.6 EPROM PROGRAMMING SPECIFICATIONS

Table A-1. EPROM Programming Electrical Characteristics

(V_{DD} = 4.2V to 5.5V, V_{SS} = 0 Vdc, T_A = 0°C to +40°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage $\overline{IRQ}/V_{PP}$	V_{PP}	—	13.5	—	V
Programming Current $\overline{IRQ}/V_{PP}$	I_{PP}	—	5	10	mA
Programming Time per byte	t_{EPGM}	—	1	—	ms

A.7 DC ELECTRICAL CHARACTERISTICS

Table A-2. DC Electrical Characteristics (705JB4)

(V_{DD} = 4.2V to 5.5V, V_{SS} = 0 Vdc, T_A = 0°C to +40°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Supply Current (see Notes)					
Run (USB active)	I_{DD}	—	—	7.5	mA
Run (USB suspended)		—	—	6.5	mA
Wait (USB active)		—	—	4.0	mA
Wait (USB suspended)		—	—	3.0	mA
Stop					
LVR off at 25°C (not include 15k to GND)		—	130	180	µA

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado 80217. 1-800-441-2447 or 1-303-675-2140

JAPAN: Nippon Motorola Ltd. SPD, Strategic Planning Office 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan. 03-5487-8488

Mfax™, Motorola Fax Back System: RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/; TOUCHTONE 1-602-244-6609; US and Canada ONLY 1-800-774-1848

HOME PAGE: http://motorola.com/sps/

MOTOROLA