

Welcome to [E-XFL.COM](#)

**[Embedded - Microcontrollers - Application Specific](#): Tailored Solutions for Precision and Performance**

**[Embedded - Microcontrollers - Application Specific](#)** represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

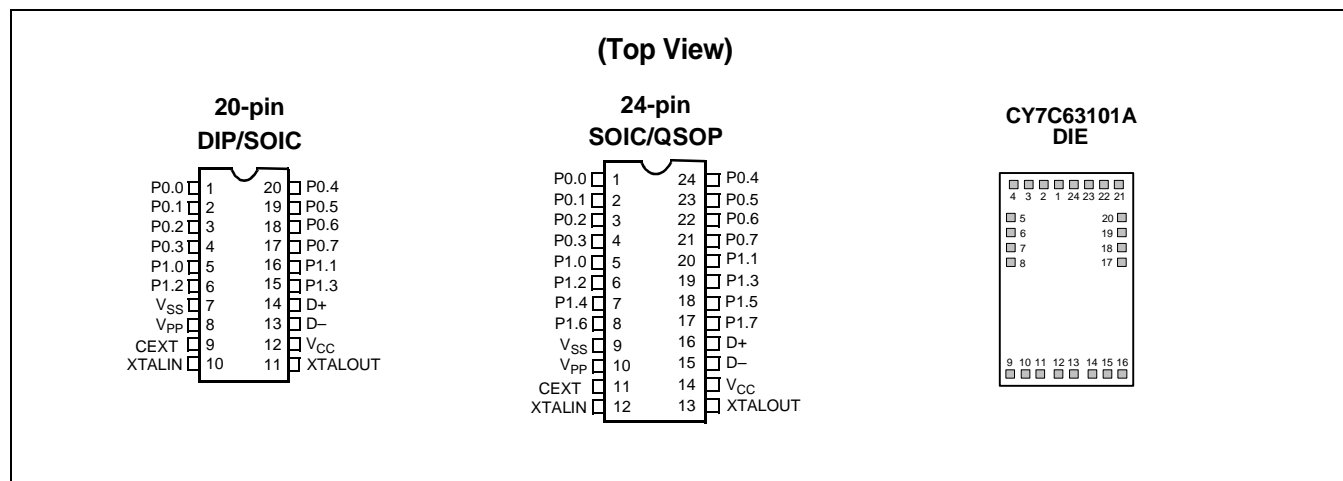
**What Are [Embedded - Microcontrollers - Application Specific](#)?**

Application specific microcontrollers are engineered to

#### Details

Product Status	Obsolete
Applications	USB Microcontroller
Core Processor	M8A
Program Memory Type	OTP (4kB)
Controller Series	CY7C631xx
RAM Size	128 x 8
Interface	USB
Number of I/O	16
Voltage - Supply	4V ~ 5.25V
Operating Temperature	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	24-SSOP (0.154", 3.90mm Width)
Supplier Device Package	24-QSOP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/infineon-technologies/cy7c63101a-qc">https://www.e-xfl.com/product-detail/infineon-technologies/cy7c63101a-qc</a>

## 2.0 Pin Configurations



## 3.0 Functional Overview

The CY7C630/101A is a family of 8-bit RISC One Time Programmable (OTP) microcontrollers with a built-in 1.5-Mbps USB Serial Interface Engine (SIE). The microcontroller features 35 instructions that are optimized for USB applications. In addition, the microcontroller features 128 bytes of internal RAM and four Kbytes of program memory space. The Cypress USB Controller accepts a 6-MHz ceramic resonator as its clock source. This clock signal is doubled within the chip to provide a 12-MHz clock for the microprocessor.

The microcontroller features two ports of up to sixteen general purpose I/Os (GPIOs). Each GPIO pin can be used to generate an interrupt to the microcontroller. Additionally, all

pins in Port 1 are equipped with programmable drivers strong enough to drive LEDs. The GPIO ports feature low EMI emissions as a result of controlled rise and fall times and unique output driver circuits. The Cypress microcontrollers have a range of GPIOs to fit various applications; the CY7C6300XA has twelve GPIOs and the CY7C6310XA has sixteen GPIOs. Notice that each part has eight 'low-current' ports (Port 0) with the remaining ports (Port 1) being 'high-current' ports.

The 12-GPIO CY7C6300XA is available in 20-pin PDIP (-PC) and 20-pin SOIC (-SC) packages. The 26-GPIO CY7C6310XA is available in 24-pin QSOP (-QC) package.

## 4.0 Pin Definitions

Name	I/O	20-Pin	24-pin	Die Pad #	Description
P0.0	I/O	1	1	1	Port 0 bit 0
P0.1	I/O	2	2	2	Port 0 bit 1
P0.2	I/O	3	3	3	Port 0 bit 2
P0.3	I/O	4	4	4	Port 0 bit 3
P0.4	I/O	20	24	24	Port 0 bit 4
P0.5	I/O	19	23	23	Port 0 bit 5
P0.6	I/O	18	22	22	Port 0 bit 6
P0.7	I/O	17	21	21	Port 0 bit 7
P1.0	I/O	5	5	5	Port 1 bit 0
P1.1	I/O	16	20	20	Port 1 bit 1
P1.2	I/O	6	6	6	Port 1 bit 2
P1.3	I/O	15	19	19	Port 1 bit 3
P1.4	I/O	—	7	7	Port 1 bit 4
P1.5	I/O	—	18	18	Port 1 bit 5
P1.6	I/O	—	8	8	Port 1 bit 6
P1.7	I/O	—	17	17	Port 1 bit 7
XTALIN	I	10	12	12	Ceramic resonator in
XTALOUT	O	11	13	13	Ceramic resonator out

## 4.0 Pin Definitions (continued)

Name	I/O	20-Pin	24-pin	Die Pad #	Description
CEXT	I/O	9	11	11	Connects to external R/C timing circuit for optional 'suspend' wakeup
D+	I/O	14	16	16	USB data+
D-	I/O	13	15	15	USB data-
V <sub>PP</sub>	–	8	10	10	Programming voltage supply, tie to ground during normal operation
V <sub>CC</sub>	–	12	14	14	Voltage supply
V <sub>SS</sub>	–	7	9	9	Ground

## 5.0 Pin Description

Name	Description
V <sub>CC</sub>	One pin. Connects to the USB power source or to a nominal 5V power supply. Actual V <sub>CC</sub> range can vary between 4.0V and 5.25V.
V <sub>SS</sub>	One pin. Connects to ground.
V <sub>PP</sub>	One pin. Used in programming the on-chip EPROM. This pin should be tied to ground during normal operations.
XTALIN	One pin. Input from an external ceramic resonator.
XTALOUT	One pin. Return path for the ceramic resonator (leave unconnected if driving XTALIN from an external oscillator).
P0.0–P0.7, P1.0–P1.7	Sixteen pins. P0.0–P0.7 are the 8 I/O lines in Port 0. P1.0–P1.7 are the 8 I/O lines in Port 1. P1.0–P1.3 are supported in the CY7C6300XA. All I/O pins include bit-programmable pull-up resistors. However, the sink current of each pin can be programmed to one of sixteen levels. Besides functioning as GPIO lines, each pin can be programmed as an interrupt input. The interrupt is edge-triggered, with programmable polarity.
D+, D-	Two pins. Bidirectional USB data lines. An external pull-up resistor must be connected between the D pin and V <sub>CC</sub> to select low-speed USB operation.
CEXT	One pin. Open-drain output with Schmitt trigger input. The input is connected to a rising edge-triggered interrupt. CEXT may be connected to an external RC to generate a wake-up from Suspend mode. See Section 6.4.

## 6.0 Functional Description

The Cypress CY7C630/101A USB microcontrollers are optimized for human-interface computer peripherals such as a mouse, joystick, and gamepad. These USB microcontrollers conform to the low-speed (1.5 Mbps) requirements of the USB Specification version 1.1. Each microcontroller is a self-contained unit with: a USB interface engine, USB transceivers, an 8-bit RISC microcontroller, a clock oscillator, timers, and program memory. Each microcontroller supports one USB device address and two endpoints.

The 6-MHz clock is doubled to 12 MHz to drive the microcontroller. A RISC architecture with 35 instructions provides the best balance between performance and product cost.

### 6.1 Memory Organization

The memory in the USB Controller is organized into user program memory in EPROM space and data memory in SRAM space.

#### 6.1.1 Program Memory Organization

The CY7C63001A and CY7C63101A each offer 4 Kbytes of EPROM. The program memory space is divided into two functional groups: interrupt vectors and program code.

The interrupt vectors occupy the first 16 bytes of the program space. Each vector is 2 bytes long. After a reset, the Program Counter points to location zero of the program space. *Figure 6-1* shows the organization of the Program Memory Space.

#### 6.1.2 Security Fuse Bit

The Cypress USB microcontroller includes a security fuse bit. When the security fuse is programmed, the EPROM program memory outputs 0xFF to the EPROM programmer, thus protecting the user's code.

after reset		Address	
PC	→	0x0000	Reset Vector
		0x0002	Interrupt Vector – 128 $\mu$ s
		0x0004	Interrupt Vector – 1.024 ms
		0x0006	Interrupt Vector – USB Endpoint 0
		0x0008	Interrupt Vector – USB Endpoint 1
		0x000A	Reserved
		0x000C	Interrupt Vector – GPIO
		0x000E	Interrupt Vector – Cext
		0x0010	On-chip program Memory
		0x07FF	2K ROM (CY7C63000A, CY7C63100A)
		0x0FFF	4K ROM (CY7C63001A, CY7C63101A)

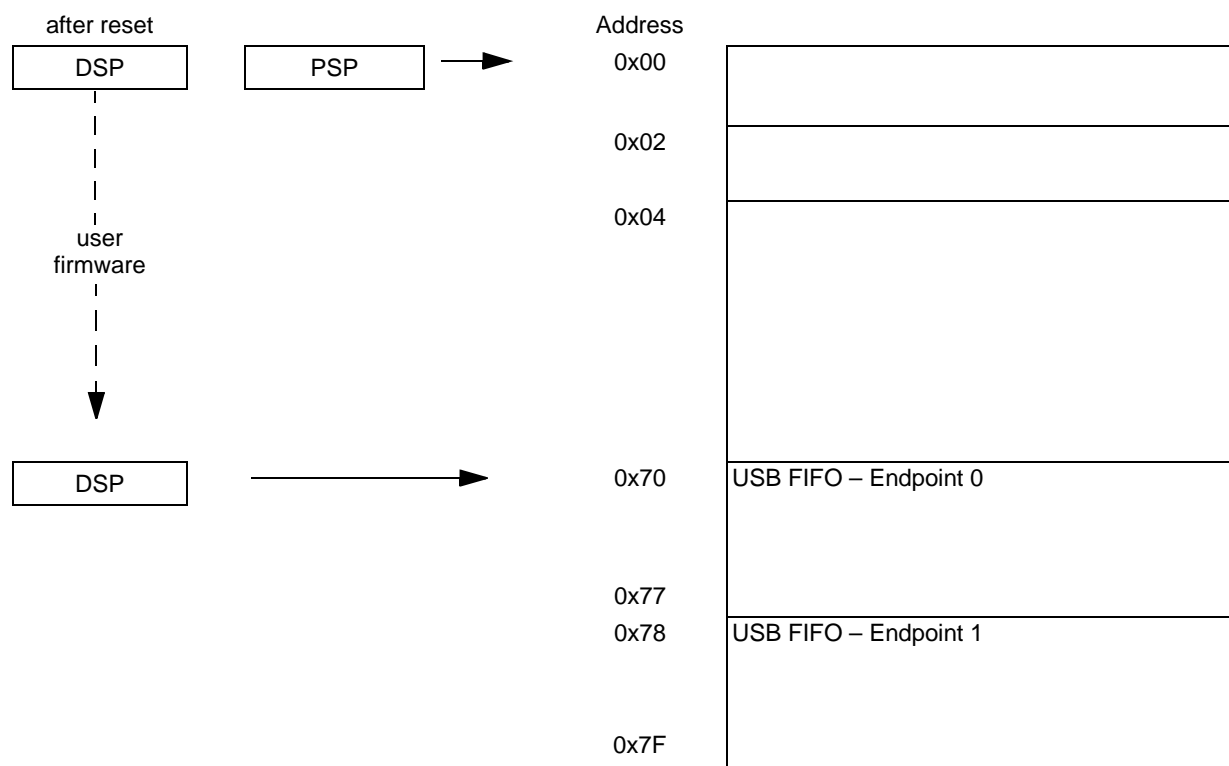
**Figure 6-1. Program Memory Space**

### 6.1.3 Data Memory Organization

The USB Controller includes 128 bytes of data RAM. The upper 16 bytes of the data memory are used as USB FIFOs for Endpoint 0 and Endpoint 1. Each endpoint is associated with an 8-byte FIFO.

The USB controller includes two pointers into data RAM, the Program Stack Pointer (PSP) and the Data Stack Pointer (DSP). The value of PSP after reset is 0x00. The PSP increments by two whenever a CALL instruction is executed and it decrements by two whenever a RET instruction is used.

The DSP pre-decrements by one whenever a PUSH instruction is executed and it increments by one after a POP instruction is used. The default value of the DSP after reset is 0x00, which would cause the first PUSH to write into USB FIFO space for Endpoint 1. Therefore, the DSP should be mapped to a location such as 0x70 before initiating any data stack operations. Refer to the Reset section for more information about DSP remapping after reset. *Figure 6-2* illustrates the Data Memory Space.



**Figure 6-2. Data Memory Space**

## 6.2 I/O Register Summary

I/O registers are accessed via the I/O Read (IORD) and I/O Write (IOWR, IOWX) instructions.

**Table 6-1. I/O Register Summary**

Register Name	I/O Address	Read/Write	Function	Page
P0 Data	0x00	R/W	General purpose I/O Port (low current)	Figure 6-8
P1 Data	0x01	R/W	General purpose I/O Port (high current)	Figure 6-9
P0 IE	0x04	W	Interrupt enable for Port 0 pins	Figure 6-17
P1 IE	0x05	W	Interrupt enable for Port 1 pins	Figure 6-18
P0 Pull-up	0x08	W	Pull-up resistor control for Port 0 pins	Figure 6-11
P1 Pull-up	0x09	W	Pull-up resistor control for Port 1 pins	Figure 6-12
EP0 TX Config.	0x10	R/W	USB Endpoint 0 transmit configuration	Figure 6-22
EP1 TX Config.	0x11	R/W	USB Endpoint 1 transmit configuration	Figure 6-23
USB DA	0x12	R/W	USB device address	Figure 6-20
USB SCR	0x13	R/W	USB status and control	Figure 6-24
EP0 RX Status	0x14	R/W	USB Endpoint 0 receive status	Figure 6-21
GIE	0x20	R/W	Global Interrupt Enable	Figure 6-15
WDT	0x21	W	Watchdog Timer clear	Figure 6-4
Cext	0x22	R/W	External R-C Timing circuit control	Figure 6-5
Timer	0x23	R	Free-running timer	Figure 6-6
P0 Isink	0x30-0x37	W	Input sink current control for Port 0 pins. There is one Isink register for each pin. Address of the Isink register for pin 0 is located at 0x30 and the register address for pin 7 is located at 0x37.	Figure 6-13

**Table 6-1. I/O Register Summary** (continued)

Register Name	I/O Address	Read/Write	Function	Page
P1 Isink	0x38-0x3F	W	Input sink current control for Port 1 pins. There is one Isink register for each pin. Address of the Isink register for pin 0 is located at 0x38 and the register address for pin 7 is located at 0x3F. The number of Port 1 pins depends on package type.	Figure 6-13
SCR	0xFF	R/W	Processor status and control register	Figure 6-3

### 6.3 Reset

The USB Controller supports three types of resets. All registers are restored to their Watchdog default states during a reset. The USB Device Address is set to 0 and all interrupts are disabled. In addition, the Program Stack Pointer (PSP) is set to 0x00 and the Data Stack Pointer (DSP) is set to 0x00. The user should set the DSP to a location such as 0x70 to reserve 16 bytes of USB FIFO space. The assembly instructions to do so are:

```
MOV A, 70h      ; Move 70 hex into Accumulator, use 70
                ; instead of 6F because the dsp is
                ; always decremented by 1 before the
                ; data transfer of the PUSH instruction occurs
SWAP A, DSP     ; Move Accumulator value into dsp
```

The three reset types are:

1. Power-On Reset (POR)
2. Watchdog Reset (WDR)
3. USB Reset

The occurrence of a reset is recorded in the Status and Control Register located at I/O address 0xFF (Figure 6-3). Reading and writing this register are supported by the IORD and IOWR instructions. Bits 1, 2, and 7 are reserved and must be written as zeros during a write. During a read, reserved bit positions should be ignored. Bits 4, 5, and 6 are used to record the occurrence of POR, USB, and WDR Reset respectively. The firmware can interrogate these bits to determine the cause of a reset. If a Watchdog Reset occurs, firmware must clear the WDR bit (bit 6) in the Status and Control Register to re-enable the USB transmitter (please refer to the Watchdog Reset section for further details). Bit 0, the "Run" control, is set to 1 at POR. Clearing this bit stops the microcontroller (firmware normally should not clear this bit). Once this bit is set to LOW, only a reset can set this bit HIGH.

The microcontroller resumes execution from ROM address 0x00 after a reset unless the Suspend bit (bit 3) of the Status and Control Register is set. Setting the Suspend bit stops the clock oscillator and the interrupt timers and powers down the microcontroller. The detection of any USB activity, the occurrence of a GPIO Interrupt, or the occurrence of the Cext Interrupt terminates the suspend condition.

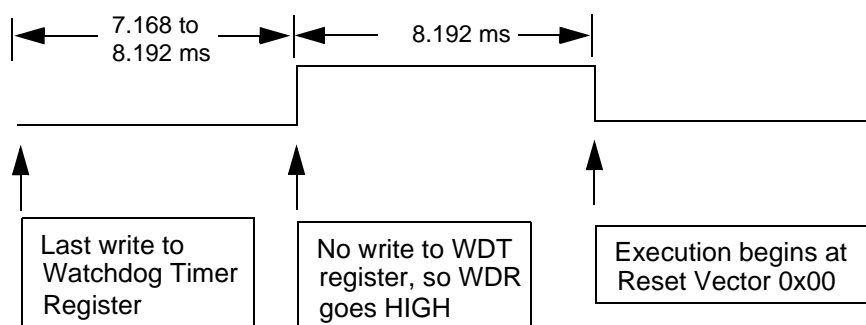
b7	b6	b5	b4	b3	b2	b1	b0
Reserved	WDR	USBR	POR	SUSPEND	Reserved	Reserved	RUN
	R/W	R/W	R/W	R/W			R/W
0	0	0	1	0	0	0	1

**Figure 6-3. Status and Control Register (SCR – Address 0xFF)**

#### 6.3.1 Power-On Reset

Power-On Reset (POR) occurs every time the power to the device is switched on. Bit 4 of the Status and Control Register is set to record this event (the register contents are set to 00011001 by the POR). The USB Controller is placed in

suspended mode at the end of POR to conserve power (the clock oscillator, the timers, and the interrupt logic are turned off in suspend mode). After POR, only a non-idle USB Bus state terminates the suspend mode. The microcontroller then begins execution from ROM address 0x00.



**Figure 6-4. Watchdog Reset**

### 6.3.2 Watchdog Reset (WDR)

The Watchdog Timer Reset (WDR) occurs when the Most Significant Bit of the 4-bit Watchdog Timer Register transitions from LOW to HIGH. Writing any value to the write-only Watchdog Restart Register at 0x21 clears the timer (firmware should periodically write to the Watchdog Restart Register in the 'main loop' of firmware). The Watchdog timer is clocked by a 1.024-ms clock from the free-running timer. If 8 clocks occur between writes to the timer, a WDR occurs and bit 6 of the Status and Control Register is set to record the event. A Watchdog Timer Reset lasts for 8.192 ms, at which time the microcontroller begins execution at ROM address 0x00. The USB transmitter is disabled by a Watchdog Reset because the USB Device Address Register is cleared (otherwise, the USB Controller would respond to all address 0 transactions). The transmitter remains disabled until the WDR bit (bit 6) in the Status and Control Register is reset to 0 by firmware.

### 6.3.3 USB Bus Reset

The USB Controller recognizes a USB Reset when a Single Ended Zero (SE0) condition persists for at least 8–16  $\mu$ s (the Reset may be recognized for an SE0 as short as 8  $\mu$ s, but it is always recognized for an SE0 longer than 16  $\mu$ s). SE0 is the condition in which both the D+ line and the D– line are LOW. Bit 5 of the Status and Control Register is set to record this event. If the USB reset happens while the device is suspended, the suspend condition is cleared and the clock oscillator is restarted. However, the microcontroller is not released until the USB reset is removed.

## 6.4 Instant-on Feature (Suspend Mode)

The USB Controller can be placed in a low-power state by setting the Suspend bit (bit 3) of the Status and Control

register. All logic blocks in the device are turned off except the USB receiver, the GPIO interrupt logic, and the Cext interrupt logic. The clock oscillator and the free-running and Watchdog timers are shut down.

The suspend mode is terminated when one of the following three conditions occur:

1. USB activity
2. A GPIO interrupt
3. Cext interrupt

The clock oscillator, GPIO, and timers restart immediately upon exiting suspend mode. The USB engine and microcontroller return to a fully functional state no more than 256  $\mu$ s later. Before servicing any interrupt requests, the microcontroller executes the instruction following the I/O write that placed the device into suspend mode.

Both the GPIO interrupt and the Cext interrupt allow the USB Controller to wake-up periodically and poll potentiometers, optics, and other system components while maintaining a very low average power consumption. The Cext Interrupt is preferred for lowest power consumption.

For Cext to generate an "Instant-on" interrupt, the pin must be connected to ground with an external capacitor and connected to  $V_{CC}$  with an external resistor. A "0" is written to the Cext register located at I/O address 0x22 to discharge the capacitor. Then, a "1" is written to disable the open-drain output driver. A Schmitt trigger input circuit monitors the input and generates a wake-up interrupt when the input voltage rises above the input threshold. By changing the values of the external resistor and capacitor, the user can fine tune the charge rate of the R-C timing circuit. The format of the Cext register is shown in Figure 6-5. Reading the register returns the value of the Cext pin. During a reset, the Cext pin is HIGH.

b7	b6	b5	b4	b3	b2	b1	b0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	CEXT
							R/W
0	0	0	0	0	0	0	1

**Figure 6-5. The Cext Register (Address 0x22)**

## 6.5 On-Chip Timer

The USB Controller is equipped with a free-running timer driven by a clock one-sixth the resonator frequency. Bits 0 through 7 of the counter are readable from the read-only Timer Register located at I/O address 0x23. The Timer Register is cleared during a Power-On Reset and whenever Suspend

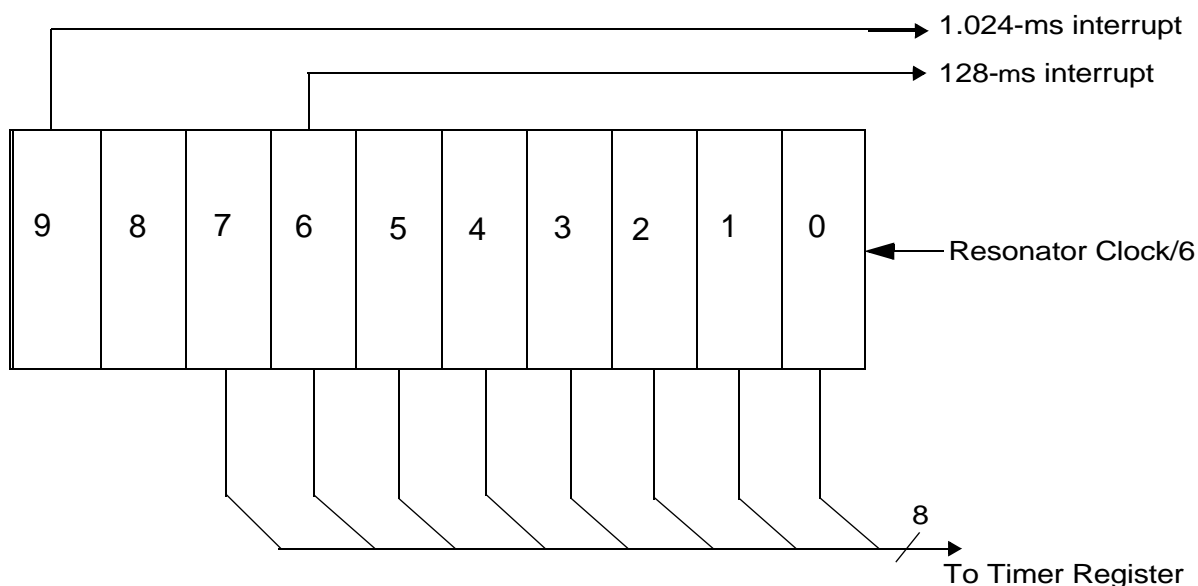
mode is entered. *Figure 6-6* illustrates the format of this register and *Figure 6-7* is its block diagram.

With a 6 MHz resonator, the timer resolution is 1  $\mu$ s.

The timer generates two interrupts: the 128- $\mu$ s interrupt and the 1.024-ms interrupt.

**Figure 6-6. Timer Register (Address 0x23)**

b7	b6	b5	b4	b3	b2	b1	b0
T.7	T.6	T.5	T.4	T.3	T.2	T.1	T.0
R	R	R	R	R	R	R	R
0	0	0	0	0	0	0	0



**Figure 6-7. Timer Block Diagram**

## 6.6 General Purpose I/O Ports

Interface with peripherals is conducted via as many as 16 GPIO signals. These signals are divided into two ports: Port 0 and Port 1. Port 0 contains eight lines (P0.0–P0.7) and Port 1 contains up to eight lines (P1.0–P1.7). The number of external I/O pins depends on the package type. Both ports can be accessed by the IORD, IOWR, and IOWX instructions. The

Port 0 data register is located at I/O address 0x00 while the Port 1 data register is located at I/O address 0x01. The contents of both registers are set HIGH during a reset. Refer to *Figures 6-8* and *6-9* for the formats of the data registers. In addition to supporting general input/output functions, each I/O line can trigger an interrupt to the microcontroller. Please refer to the interrupt section for more details.

b7	b6	b5	b4	b3	b2	b1	b0
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

**Figure 6-8. Port 0 Data Register (Address 0x00)**

b7	b6	b5	b4	b3	b2	b1	b0
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
1	1	1	1	1	1	1	1

**Figure 6-9. Port 1 Data Register (Address 0x01)**



Each GPIO line includes an internal  $R_{up}$  resistor. This resistor provides both the pull-up function and slew control. Two factors govern the enabling and disabling of each resistor: the state of its associated Port Pull-up register bit and the state of the Data Register bit. NOTE: The control bits in the Port Pull-up register are active LOW.

A GPIO line is HIGH when a “1” is written to the Data Register and a “0” is written to the respective Port Pull-up register. Writing a “0” to the port Data Register disables the port’s Pull-up resistor and outputs a LOW on the GPIO line regardless of the setting in the Port Pull-up Register. The output goes to a high-Z state if the Data Register bit and the Port Pull-up Register bit are both “1”. Figure 6-10 illustrates the block diagram of one I/O line.

The Port Isink Register is used to control the output current level and it is described later in this section. NOTE: The Isink logic block is turned off during suspend mode (please refer to the Instant-on Feature section for more details). Therefore, to prevent higher  $I_{CC}$  currents during USB suspend mode, firmware must set ALL Port 0 and Port 1 Data Register bits (which are not externally driven to a known state), **including those that are not bonded out on a particular package**, to “1” and all Port 0 and Port 1 Pull-Up Register data bits to “0” to enable port pull-ups before setting the Suspend bit (bit 3 of the Status and Control Register). Table 6-2 is the Output Control truth table.

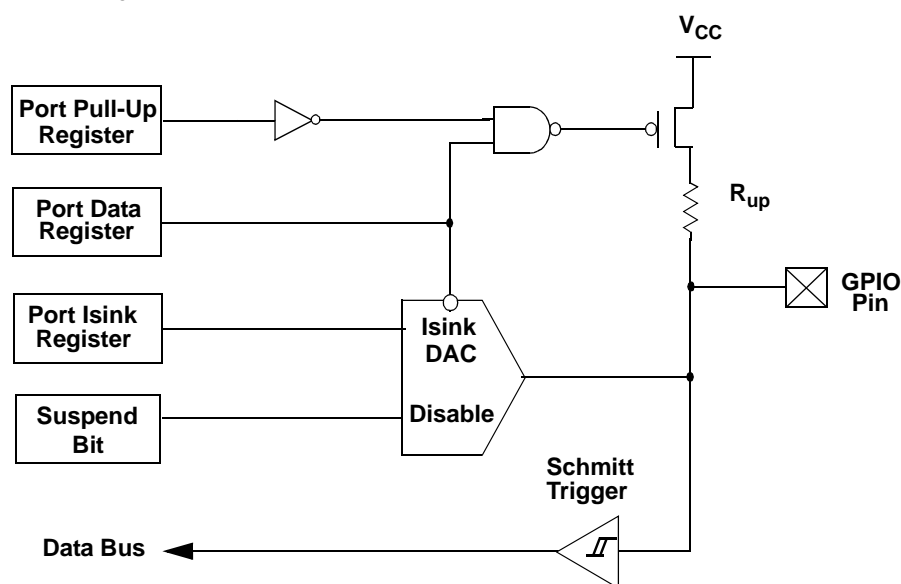


Figure 6-10. Block Diagram of an I/O Line

Table 6-2. Output Control Truth Table

Data Register	Port Pull-up Register	Output at I/O Pin	Interrupt Polarity
0	0	Sink Current ('0')	High to Low
0	1	Sink Current ('0')	Low to High
1	0	Pull-up Resistor ('1')	High to Low
1	1	Hi-Z	Low to High

To configure a GPIO pin as an input, a “1” should be written to the Port Data Register bit associated with that pin to disable the pull-down function of the Isink DAC (see Figure 6-10). When the Port Data Register is read, the bit value is a “1” if the voltage on the pin is greater than the Schmitt trigger threshold, or “0” if it is below the threshold. In applications where an internal pull-up is required, the  $R_{up}$  pull-up resistor can be engaged by writing a “0” to the appropriate bit in the Port Pull-up Register.

Both Port 0 and Port 1 Pull-up Registers are write only (see Figures 6-11 and 6-12). The Port 0 Pull-up Register is located at I/O address 0x08 and Port 1 Pull-up Register is mapped to address 0x09. The contents of the Port Pull-up Registers are cleared during reset, allowing the outputs to be controlled by the state of the Data Registers. The Port Pull-up Registers also select the polarity of transition that generates a GPIO interrupt. A “0” selects a HIGH to LOW transition while a “1” selects a LOW to HIGH transition.

b7	b6	b5	b4	b3	b2	b1	b0
PULL0.7	PULL0.6	PULL0.5	PULL0.4	PULL0.3	PULL0.2	PULL0.1	PULL0.0
W	W	W	W	W	W	W	W
0	0	0	0	0	0	0	0

Figure 6-11. Port 0 Pull-up Register (Address 0x08)

When servicing an interrupt, the hardware first disables all interrupts by clearing the Global Interrupt Enable Register. Next, the interrupt latch of the current interrupt is cleared. This is followed by a CALL instruction to the ROM address associated with the interrupt being serviced (i.e., the interrupt vector). The instruction in the interrupt table is typically a JMP instruction to the address of the Interrupt Service Routine (ISR). The user can re-enable interrupts in the interrupt service routine by writing to the appropriate bits in the Global Interrupt Enable Register. Interrupts can be nested to a level limited only by the available stack space.

The Program Counter (PC) value and the Carry and Zero flags (CF, ZF) are automatically stored onto the Program Stack by the CALL instruction as part of the interrupt acknowledge

process. The user firmware is responsible for ensuring that the processor state is preserved and restored during an interrupt. For example the PUSH A instruction should be used as the first command in the ISR to save the accumulator value. And, the IPRET instruction should be used to exit the ISR with the accumulator value restored and interrupts enabled. The PC, CF, and ZF are restored when the IPRET or RET instructions are executed.

The Interrupt Vectors supported by the USB Controller are listed in Table 6-3. Interrupt Vector 0 (Reset) has the highest priority, Interrupt Vector 7 has the lowest priority. Because the JMP instruction is two bytes long, the interrupt vectors occupy two bytes.

**Table 6-3. Interrupt Vector Assignments**

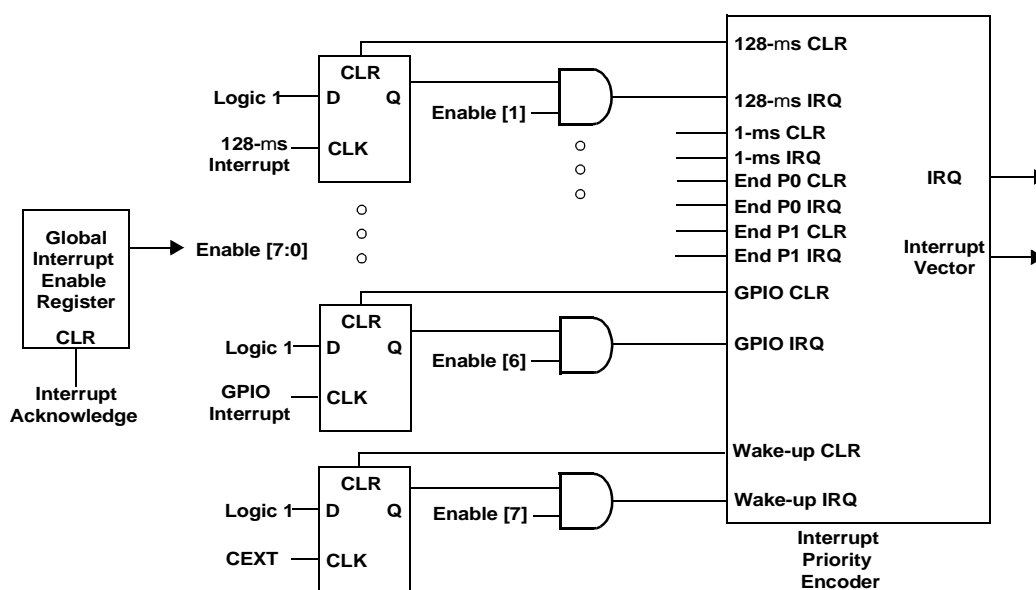
Interrupt Priority	ROM Address	Function
0 (Highest)	0x00	Reset
1	0x02	128-μs timer interrupt
2	0x04	1.024-ms timer interrupt
3	0x06	USB endpoint 0 interrupt
4	0x08	USB endpoint 1 interrupt
5	0x0A	Reserved
6	0x0C	GPIO interrupt
7 (Lowest)	0x0E	Wake-up interrupt

### 6.8.1 Interrupt Latency

Interrupt latency can be calculated from the following equation:

Interrupt Latency = (Number of clock cycles remaining in the current instruction) + (10 clock cycles for the CALL instruction) + (5 clock cycles for the JMP instruction)

For example, if a 5-clock-cycle instruction such as JC is being executed when an interrupt occurs, the first instruction of the Interrupt Service Routine executes a minimum of 16 clock cycles (1+10+5) or a maximum of 20 clock cycles (5+10+5) after the interrupt is issued. Therefore, the interrupt latency in this example will be = 20 clock periods = 20 / (12 MHz) = 1.667 μs. The interrupt latches are sampled at the rising edge of the last clock cycle in the current instruction.



**Figure 6-16. Interrupt Controller Logic Block Diagram**

### 6.8.2 GPIO Interrupt

The General Purpose I/O interrupts are generated by signal transitions at the Port 0 and Port 1 I/O pins. GPIO interrupts are edge sensitive with programmable interrupt polarities. Setting a bit HIGH in the Port Pull-up Register (see *Figure 6-11* and *6-12*) selects a LOW to HIGH interrupt trigger for the corresponding port pin. Setting a bit LOW activates a HIGH to

LOW interrupt trigger. Each GPIO interrupt is maskable on a per-pin basis by a dedicated bit in the Port Interrupt Enable Register. Writing a “1” enables the interrupt. *Figure 6-17* and *Figure 6-18* illustrate the format of the Port Interrupt Enable Registers for Port 0 and Port 1 located at I/O address 0x04 and 0x05 respectively. These write only registers are cleared during reset, thus disabling all GPIO interrupts.

b7	b6	b5	b4	b3	b2	b1	b0
IE0.7	IE0.6	IE0.5	IE0.4	IE0.3	IE0.2	IE0.1	IE0.0
W	W	W	W	W	W	W	W
0	0	0	0	0	0	0	0

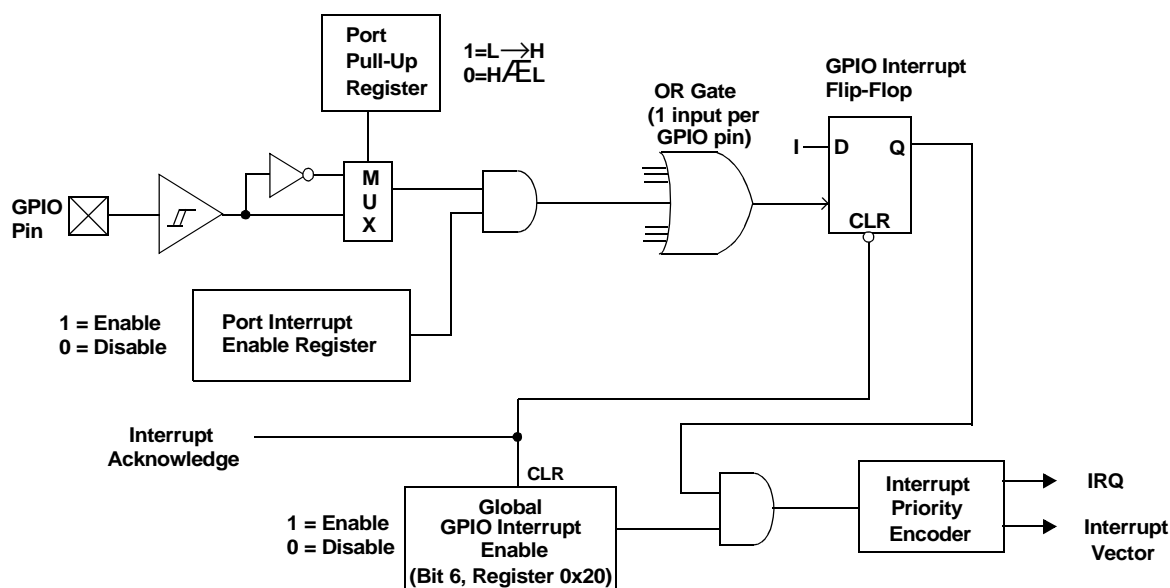
**Figure 6-17. Port 0 Interrupt Enable Register (P0 IE – Address 0x04)**

b7	b6	b5	b4	b3	b2	b1	b0
IE1.7	IE1.6	IE1.5	IE1.4	IE1.3	IE1.2	IE1.1	IE1.0
W	W	W	W	W	W	W	W
0	0	0	0	0	0	0	0

**Figure 6-18. Port 1 Interrupt Enable Register (P1 IE – Address 0x05)**

A block diagram of the GPIO interrupt logic is shown in *Figure 6-19*. The bit setting in the Port Pull-up Register selects the interrupt polarity. If the selected signal polarity is detected on the I/O pin, a HIGH signal is generated. If the Port Interrupt Enable bit for this pin is HIGH and no other port pins are requesting interrupts, the OR gate issues a LOW to HIGH

signal to clock the GPIO interrupt flip-flop. The output of the flip-flop is further qualified by the Global GPIO Interrupt Enable bit before it is processed by the Interrupt Priority Encoder. Both the GPIO interrupt flip-flop and the Global GPIO Enable bit are cleared by on-chip hardware during GPIO interrupt acknowledge.



**Figure 6-19. GPIO Interrupt Logic Block Diagram**

**Note.** If one port pin triggers an interrupt, no other port pin can cause a GPIO interrupt until the port pin that triggered the interrupt has returned to its inactive (non-trigger) state or until its corresponding port interrupt enable bit is cleared (these events ‘reset’ the clock of the GPIO Interrupt flip-flop, which must be ‘reset’ to ‘0’ before another GPIO interrupt event can ‘clock’ the GPIO Interrupt flip-flop and produce an IRQ).

**Note.** If the port pin that triggered an interrupt is held in its active (trigger) state while its corresponding port interrupt

enable bit is cleared and then set, a GPIO interrupt event occurs as the GPIO Interrupt flip-flop clock transitions from ‘1’ to ‘0’ and then back to ‘1’ (please refer to *Figure 6-19*). The USB Controller does not assign interrupt priority to different port pins and the Port Interrupt Enable Registers are not cleared during the interrupt acknowledge process. When a GPIO interrupt is serviced, the ISR must poll the ports to determine which pin caused the interrupt.

### 6.8.3 USB Interrupt

A USB Endpoint 0 interrupt is generated after the host has written data to Endpoint 0 or after the USB Controller has transmitted a packet from Endpoint 0 and receives an ACK from the host. An OUT packet from the host which is NAKED by the USB Controller does not generate an interrupt. This interrupt is masked by the USB EP0 Interrupt Enable bit (bit 3) of the Global Interrupt Enable Register.

A USB Endpoint 1 interrupt is generated after the USB Controller has transmitted a packet from Endpoint 1 and has received an ACK from the host. This interrupt is masked by the USB EP1 Interrupt Enable bit (bit 4) of the Global Interrupt Enable Register.

### 6.8.4 Timer Interrupt

There are two timer interrupts: the 128- $\mu$ s interrupt and the 1.024-ms interrupt. They are masked by bits 1 and 2 of the Global Interrupt Enable Register respectively. The user should disable both timer interrupts before going into the suspend mode to avoid possible conflicts from timer interrupts occurring just as suspend mode is entered.

### 6.8.5 Wake-Up Interrupt

A wake-up interrupt is generated when the Cext pin goes HIGH. This interrupt is latched in the interrupt controller. It can be masked by the Wake-up Interrupt Enable bit (bit 7) of the Global Interrupt Enable Register. This interrupt can be used to perform periodic checks on attached peripherals when the USB Controller is placed in the low-power suspend mode. See the Instant-On Feature section for more details.

## 6.9 USB Engine

The USB engine includes the Serial Interface Engine (SIE) and the low-speed USB I/O transceivers. The SIE block performs most of the USB interface functions with only minimal support from the microcontroller core. Two endpoints are supported. Endpoint 0 is used to receive and transmit control (including setup) packets while Endpoint 1 is only used to transmit data packets.

The USB SIE processes USB bus activity at the transaction level independently. It does all the NRZI encoding/decoding and bit stuffing/unstuffing. It also determines token type, checks address and endpoint values, generates and checks CRC values, and controls the flow of data bytes between the bus and the Endpoint FIFOs. NOTE: the SIE stalls the CPU for three cycles per byte when writing data to the endpoint FIFOs (or  $3 * 1/12 \text{ MHz} * 8 \text{ bytes} = 2 \mu\text{s}$  per 8-byte transfer).

The firmware handles higher level and function-specific tasks. During control transfers the firmware must interpret device requests and respond correctly. It also must coordinate Suspend/Resume, verify and select DATA toggle values, and perform function specific tasks.

The USB engine and the firmware communicate through the Endpoint FIFOs, USB Endpoint interrupts, and the USB registers described in the sections below.

### 6.9.1 USB Enumeration Process

The USB Controller provides a USB Device Address Register at I/O location 0x12. Reading and writing this register is achieved via the IORD and IOWR instructions. The register contents are cleared during a reset, setting the USB address of the USB Controller to 0. Figure 6-20 shows the format of the USB Address Register.

b7	b6	b5	b4	b3	b2	b1	b0
Reserved	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

**Figure 6-20. USB Device Address Register (USB DA – Address 0x12)**

Typical enumeration steps:

1. The host computer sends a SETUP packet followed by a DATA packet to USB address 0 requesting the Device descriptor.
2. The USB Controller decodes the request and retrieves its Device descriptor from the program memory space.
3. The host computer performs a control read sequence and the USB Controller responds by sending the Device descriptor over the USB bus.
4. After receiving the descriptor, the host computer sends a SETUP packet followed by a DATA packet to address 0 assigning a new USB address to the device.
5. The USB Controller stores the new address in its USB Device Address Register after the no-data control sequence completes.
6. The host sends a request for the Device descriptor using the new USB address.
7. The USB Controller decodes the request and retrieves the Device descriptor from the program memory.
8. The host performs a control read sequence and the USB Controller responds by sending its Device descriptor over the USB bus.
9. The host generates control reads to the USB Controller to request the Configuration and Report descriptors.
10. The USB Controller retrieves the descriptors from its program space and returns the data to the host over the USB.
11. Enumeration is complete after the host has received all the descriptors.

### 6.9.2 Endpoint 0

All USB devices are required to have an endpoint number 0 that is used to initialize and manipulate the device. Endpoint 0 provides access to the device's configuration information and allows generic USB status and control accesses.

Endpoint 0 can receive and transmit data. Both receive and transmit data share the same 8-byte Endpoint 0 FIFO located at data memory space 0x70 to 0x77. Received data may overwrite the data previously in the FIFO.

#### 6.9.2.1 Endpoint 0 Receive

After receiving a packet and placing the data into the Endpoint 0 FIFO, the USB Controller updates the USB Endpoint 0 RX register to record the receive status and then generates a USB Endpoint 0 interrupt. The format of the Endpoint 0 RX Register is shown in *Figure 6-21*.

b7	b6	b5	b4	b3	b2	b1	b0
COUNT3	COUNT2	COUNT1	COUNT0	TOGGLE	IN	OUT	SETUP
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
0	0	0	0	0	0	0	0

**Figure 6-21. USB Endpoint 0 RX Register (Address 0x14)**

This is a read/write register located at I/O address 0x14. Any write to this register clears all bits except bit 3 which remains unchanged. All bits are cleared during reset.

Bit 0 is set to 1 when a SETUP token for Endpoint 0 is received. Once set to a 1, this bit remains HIGH until it is cleared by an I/O write or a reset. While the data following a SETUP is being received by the USB engine, this bit is not cleared by an I/O write. User firmware writes to the USB FIFOs are disabled when bit 0 is set. This prevents SETUP data from being overwritten.

Bits 1 and 2 are updated whenever a valid token is received on Endpoint 0. Bit 1 is set to 1 if an OUT token is received and cleared to 0 if any other token is received. Bit 2 is set to 1 if an IN token is received and cleared to 0 if any other token is received.

Bit 3 shows the Data Toggle status of DATA packets received on Endpoint 0. This bit is updated for DATA following SETUP tokens and for DATA following OUT tokens if Stall (bit 5 of 0x10) is not set and either EnableOuts or StatusOuts (bits 3 and 4 of 0x13) are set.

Bits 4 to 7 are the count of the number of bytes received in a DATA packet. The two CRC bytes are included in the count, so the count value is two greater than the number of data bytes

received. The count is always updated and the data is always stored in the FIFO for DATA packets following a SETUP token. The count for DATA following an OUT token is updated if Stall (bit 5 of 0x10) is 0 and either EnableOuts or StatusOuts (bits 3 and 4 of 0x13) are 1. The DATA following an OUT is written into the FIFO if EnableOuts is set to 1 and Stall and StatusOuts are 0.

A maximum of eight bytes are written into the Endpoint 0 FIFO. If there are less than eight bytes of data the CRC is written into the FIFO.

Due to register space limitations, the Receive Data Invalid bit is located in the USB Endpoint 0 TX Configuration Register. Refer to the Endpoint 0 Transmit section for details. This bit is set by the SIE if an error is detected in a received DATA packet.

*Table 6-4* summarizes the USB Engine response to SETUP and OUT transactions on Endpoint 0. In the Data Packet column 'Error' represents a packet with a CRC, PID or bit-stuffing error, or a packet with more than eight bytes of data. 'Valid' is a packet without an Error. 'Status' is a packet that is a valid control read Status stage, while 'N/Status' is not a correct Status stage (see section 6.9.4). The 'Stall' bit is described in Section 6.9.2.2. The 'StatusOuts' and 'EnableOuts' bits are described in section 6.9.4.

**Table 6-4. USB Engine Response to SETUP and OUT Transactions on Endpoint 0**

Control Bit Settings			Received Packets		USB Engine Response				
Stall	Status Out	Enable Out	Token Type	Data Packet	FIFO Write	Toggle Update	Count Update	Interrupt	Reply
–	–	–	SETUP	Valid	Yes	Yes	Yes	Yes	ACK
–	–	–	SETUP	Error	Yes	Yes	Yes	Yes	None
0	0	1	OUT	Valid	Yes	Yes	Yes	Yes	ACK
0	0	1	OUT	Error	Yes	Yes	Yes	Yes	None
0	0	0	OUT	Valid	No	No	No	No	NAK
0	0	0	OUT	Error	No	No	No	No	None
1	0	0	OUT	Valid	No	No	No	No	STALL
1	0	0	OUT	Error	No	No	No	No	None
0	1	0	OUT	Status	No	Yes	Yes	Yes	ACK
0	1	0	OUT	N/Status	No	Yes	Yes	Yes	STALL
0	1	0	OUT	Error	No	Yes	No	No	None

### 6.9.2.2 Endpoint 0 Transmit

The USB Endpoint 0 TX Register located at I/O address 0x10 controls data transmission from Endpoint 0 (see *Figure 6-22*). This is a read/write register. All bits are cleared during reset.

b7	b6	b5	b4	b3	b2	b1	b0
INEN	DATA1/0	STALL	ERR	COUNT3	COUNT2	COUNT1	COUNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

**Figure 6-22. USB Endpoint 0 TX Configuration Register (Address 0x10)**

Bits 0 to 3 indicate the numbers of data bytes to be transmitted during an IN packet, valid values are 0 to 8 inclusive. Bit 4 indicates that a received DATA packet error (CRC, PID, or bitstuffing error) occurred during a SETUP or OUT data phase. Setting the Stall bit (bit 5) stalls IN and OUT packets. This bit is cleared whenever a SETUP packet is received by Endpoint 0. Bit 6 (Data 1/0) must be set to 0 or 1 to select the DATA packet's toggle state (0 for DATA0, 1 for DATA1).

After the transmit data has been loaded into the FIFO, bit 6 should be set according to the data toggle state and bit 7 set to "1". This enables the USB Controller to respond to an IN packet. Bit 7 is cleared and an Endpoint 0 interrupt is generated by the SIE once the host acknowledges the data transmission. Bit 7 is also cleared when a SETUP token is

received. The Interrupt Service Routine can check bit 7 to confirm that the data transfer was successful.

### 6.9.3 Endpoint 1

Endpoint 1 is capable of transmit only. The data to be transmitted is stored in the 8-byte Endpoint 1 FIFO located at data memory space 0x78 to 0x7F.

#### 6.9.3.1 Endpoint 1 Transmit

Transmission is controlled by the USB Endpoint 1 TX Register located at I/O address 0x11 (see *Figure 6-23*). This is a read/write register. All bits are cleared during reset.

b7	b6	b5	b4	b3	b2	b1	b0
INEN	DATA1/0	STALL	EP1EN	COUNT3	COUNT2	COUNT1	COUNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
0	0	0	0	0	0	0	0

**Figure 6-23. USB Endpoint 1 TX Configuration Register (Address 0x11)**

Bits 0 to 3 indicate the numbers of data bytes to be transmitted during an IN packet, valid values are 0 to 8 inclusive.

Bit 4 must be set before Endpoint 1 can be used. If this bit is cleared, the USB Controller ignores all traffic to Endpoint 1.

Setting the Stall bit (bit 5) stalls IN and OUT packets until this bit is cleared.

Bit 6 (Data 1/0) must be set to either 0 or 1 depending on the data packet's toggle state, 0 for DATA0, 1 for DATA1.

After the transmit data has been loaded into the FIFO, bit 6 should be set according to the data toggle state and bit 7 set

to "1". This enables the USB Controller to respond to an IN packet. Bit 7 is cleared and an Endpoint 1 interrupt is generated by the SIE once the host acknowledges the data transmission.

### 6.9.4 USB Status and Control

USB status and control is regulated by USB Status and Control Register located at I/O address 0x13 as shown in *Figure 6-24*. This is a read/write register. All reserved bits must be written to zero. All bits in the register are cleared during reset.

b7	b6	b5	b4	b3	b2	b1	b0
Reserved	Reserved	Reserved	ENOUTS	STATOUTS	FORCEJ	FORCEK	BUSACT
			R/W	R/W		R/W	R/W
0	0	0	0	0	0	0	0

**Figure 6-24. USB Status and Control Register (USB SCR – Address 0x13)**

Bit 0 is set by the SIE if any USB activity except idle (D+ LOW, D- HIGH) is detected. The user program should check and clear this bit periodically to detect any loss of bus activity. Writing a 0 to this bit clears it. Writing a 1 does not change its value.

Bit 1 is used to force the on-chip USB transmitter to the K state which sends a Resume signal to the host. Bit 2 is used to force the transmitter to the J state. This bit should normally be set to

zero. However, for resume signaling, force a J state for one instruction before forcing resume.

Bit 3 is used to automatically respond to the Status stage OUT of a control read transfer on Endpoint 0. A valid Status stage OUT contains a DATA1 packet with 0 bytes of data. If the StatusOuts bit is set, the USB engine responds to a valid Status stage OUT with an ACK, and any other OUT with a STALL.

The data is not written into the FIFO when this bit is set. This bit is cleared when a SETUP token is received by Endpoint 0.

Bit 4 is used to enable the receiving of Endpoint 0 OUT packets. When this bit is set to 1, the data from an OUT transaction is written into the Endpoint 0 FIFO. If this bit is 0, data is not written to the FIFO and the SIE responds with a NAK. This bit is cleared following a SETUP or ACKed OUT transaction. **Note.** After firmware decodes a SETUP packet and prepares for a subsequent OUT transaction by setting bit 4, bit 4 is not cleared until the hand-shake phase of an ACKed OUT transaction (a NAKed OUT transaction does not clear this bit).

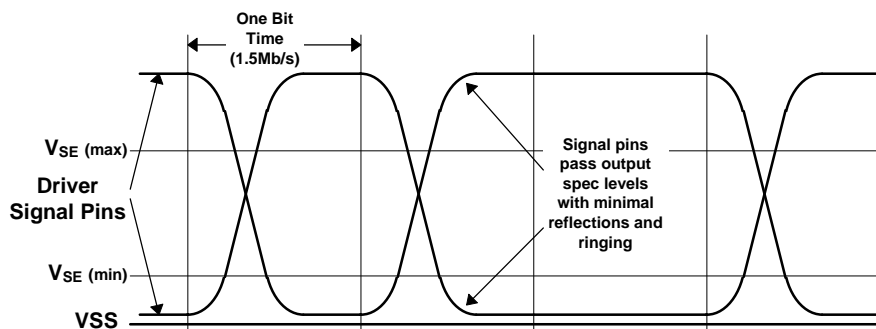
## 6.10 USB Physical Layer Characteristics

The following section describes the CY7C630/101A compliance to the Chapter 7 Electrical section of the USB Specification, Revision 1.1. The section contains all signaling, power distribution, and physical layer specifications necessary to describe a low-speed USB function.

### 6.10.1 Low-Speed Driver Characteristics

The CY7C630/101A devices use a differential output driver to drive the Low-speed USB data signal onto the USB cable, as shown in *Figure 6-25*. The output swings between the differential HIGH and LOW state are well balanced to minimize signal skew. Slew rate control on the driver minimizes the radiated noise and cross talk on the USB cable. The driver's outputs support three-state operation to achieve bidirectional half duplex operation. The CY7C630/101A driver tolerates a voltage on the signal pins of  $-0.5\text{V}$  to  $3.8\text{V}$  with respect to local ground reference without damage. The driver tolerates this voltage for  $10.0\text{ }\mu\text{s}$  while the driver is active and driving, and tolerates this condition indefinitely when the driver is in its high-impedance state.

A low-speed USB connection is made through an unshielded, untwisted wire cable a maximum of three meters in length. The rise and fall time of the signals on this cable are well controlled to reduce RFI emissions while limiting delays, signaling skews and distortions. The CY7C630/101A driver reaches the specified static signal levels with smooth rise and fall times, resulting in minimal reflections and ringing when driving the USB cable. This cable and driver are intended to be used only on network segments between low-speed devices and the ports to which they are connected.

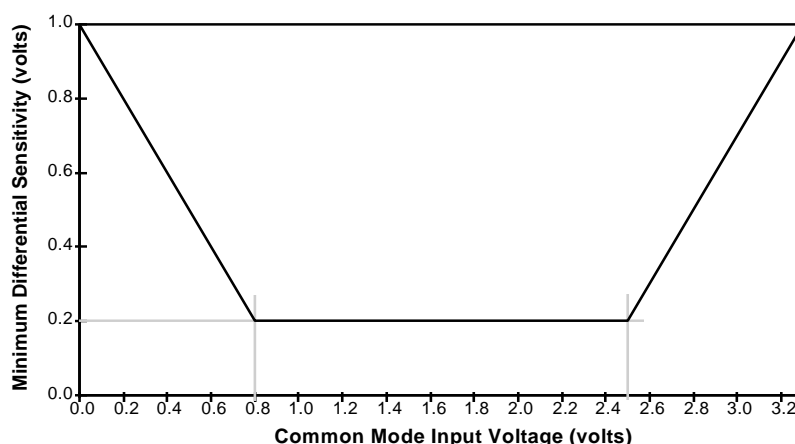


**Figure 6-25. Low-speed Driver Signal Waveforms**

### 6.10.2 Receiver Characteristics

The CY7C630/101A has a differential input receiver which is able to accept the USB data signal. The receiver features an input sensitivity of at least  $200\text{ mV}$  when both differential data inputs are in the range of at least  $0.8\text{V}$  to  $2.5\text{V}$  with respect to its local ground reference. This is the common mode input voltage range. Proper data reception is also guaranteed when

the differential data lines are outside the common mode range, as shown in *Figure 6-26*. The receiver tolerates static input voltages between  $-0.5\text{V}$  and  $3.8\text{V}$  with respect to its local ground reference without damage. In addition to the differential receiver, there is a single-ended receiver for each of the two data lines. The single-ended receivers have a switching threshold between  $0.8\text{V}$  and  $2.0\text{V}$  (TTL inputs).

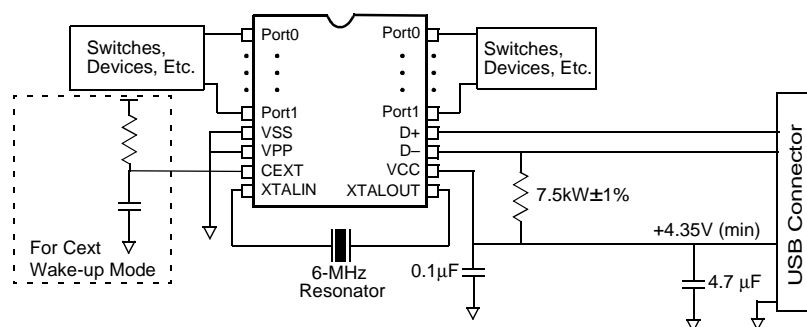


**Figure 6-26. Differential Input Sensitivity Over Entire Common Mode Range**

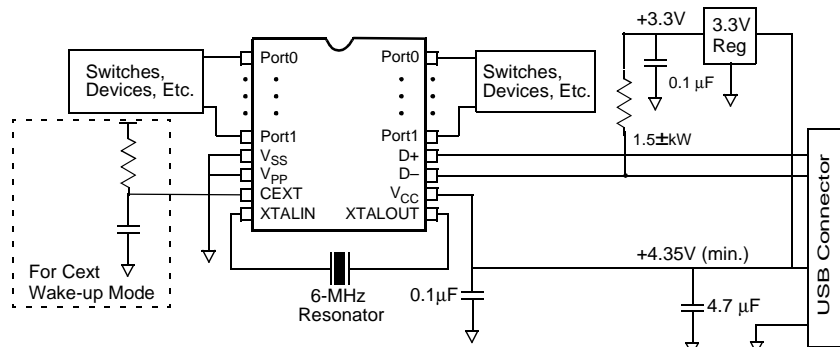
### 6.11 External USB Pull-Up Resistor

The USB system specifies that a pull-up resistor be connected on the D<sup>-</sup> pin of low-speed peripherals as shown in *Figure 6-27*. To meet the USB 1.1 spec (section 7.1.6), which states that the termination must charge the D<sup>-</sup> line from 0 to 2.0 V in 2.5  $\mu$ s, the total load capacitance on the D<sup>+</sup>/D<sup>-</sup> lines of the low-speed USB device (Cypress device capacitance + PCB

trace capacitance + integrated cable capacitance) must be less than 250 pF. As Cypress D<sup>+</sup>/D<sup>-</sup> transceiver input capacitance is 20pF max, up to 230 pF of capacitance is allowed for in the low speed device's integrated cable and PCB. If the cable + PCB capacitance on the D<sup>+</sup>/D<sup>-</sup> lines will be greater than approximately 230 pF, an external 3.3V regulator must be used as shown in *Figure 6-28*.



**Figure 6-27. Application Showing 7.5kW  $\pm$ 1% Pull-Up Resistor**



**Figure 6-28. Application Showing 1.5-kW  $\pm$ 5% Pull-Up Resistor**



## 7.0 Absolute Maximum Ratings

Storage Temperature ..... -65°C to +150°C  
 Ambient Temperature with Power Applied..... -0°C to +70°C  
 Supply Voltage on  $V_{CC}$  Relative to  $V_{SS}$  ..... -0.5V to +7.0V  
 DC Input Voltage..... -0.5V to + $V_{CC}+0.5V$

DC Voltage Applied to Outputs in  
 High-Z state ..... -0.5V to + $V_{CC}+0.5V$   
 Max. Output Current into Port 1 Pins ..... 60 mA  
 Max. Output Current into Non-Port 1 Pins..... 10 mA  
 Power Dissipation..... 300 mW  
 Static Discharge Voltage ..... >2000V  
 Latch-up Current<sup>[1]</sup> ..... > 200 mA

## 8.0 Electrical Characteristics $f_{OSC} = 6$ MHz; Operating Temperature = 0 to 70°C, $V_{CC} = 4.0$ to 5.25V

Parameter	Description	Conditions	Min.	Max.	Unit
<b>General</b>					
$I_{CC}$	$V_{CC}$ Operating Supply Current			25	mA
$I_{SB1}$	Supply Current—Suspend Mode	Resonator off, D- > $V_{oh}$ min <sup>[2]</sup>		20	μA
$I_{SB2}$	Supply Current—Start-up Mode			4	mA
$V_{PP}$	Programming Voltage (disabled)		-0.4	0.4	V
$t_{start}$	Resonator Start-up Interval	Ceramic resonator		256	μs
$t_{watch}$	Watchdog Timer Period		7.168	8.192	ms
<b>Power On Reset</b>					
$t_{VCCS}$	$V_{CC}$ Slew	Linear ramp on $V_{CC}$ pin to $V_{CC}$ <sup>[3, 4]</sup>	0.010	1000	ms
<b>USB Interface</b>					
$V_{oh}$	Static Output High	15kΩ ± 5% to Gnd <sup>[5, 6]</sup>	2.8	3.6	V
$V_{ol}$	Static Output Low	See Notes 5 and 6		0.3	V
$V_{di}$	Differential Input Sensitivity	$(D+) - (D-)$  , and Figure 6-26	0.2		V
$V_{cm}$	Differential Input Common Mode Range	Figure 6-26	0.8	2.5	V
$V_{se}$	Single Ended Receiver Threshold		0.8	2.0	V
$C_{in}$	Transceiver Input Capacitance	D+ to Vss; D- to Vss		20	pF
$I_{lo}$	Data Line (D+, D-) Leakage	0 V < (D+, D-) < 3.3 V, Hi-Z State	-10	10	μA
$R_{pu1}$	External Bus Pull-up Resistance, D- pin	1.5 kΩ ± 5% to 3.3V supply	1.425	1.575	kΩ
$R_{pu2}$	External Bus Pull-up Resistance, D- pin	7.5 kΩ ± 1% to $V_{CC}$ <sup>[7]</sup>	7.425	7.575	kΩ
$R_{pd}$	External Bus Pull-down Resistance	15 kΩ ± 5%	14.25	15.75	kΩ
<b>General Purpose I/O Interface</b>					
$R_{up}$	Pull-up Resistance		8	24	kΩ
$I_{sink0(0)}$	Port 0 Sink Current (0), lowest current	$V_{out} = 2.0V$ DC, Port 0 only <sup>[5]</sup>	0.1	0.3	mA
$I_{sink0(F)}$	Port 0 Sink Current (F), highest current	$V_{out} = 2.0V$ DC, Port 0 only <sup>[5]</sup>	0.5	1.5	mA
$I_{sink1(0)}$	Port 1 Sink Current (0), lowest current	$V_{out} = 2.0V$ DC, Port 1 only <sup>[5]</sup>	1.6	4.8	mA
$I_{sink1(F)}$	Port 1 Sink Current (F), highest current	$V_{out} = 2.0V$ DC, Port 1 only <sup>[5]</sup> $V_{out} = 0.4V$ DC, Port 1 only <sup>[5]</sup>	8 5	24	mA mA
$I_{range}$	Sink Current max./min.	$V_{out} = 2.0V$ DC, Port 0 or 1 <sup>[5, 8]</sup>	4.5	5.5	
$I_{lin}$	Differential Nonlinearity	Port 0 or Port 1 <sup>[11]</sup>		0.5	$I_{SB}$
$T_{ratio}$	Tracking Ratio Port1 to Port0	$V_{out} = 2.0V$ <sup>[12]</sup>	14.4	19.6	
$t_{sink}$	Current Sink Response Time	Full scale transition		0.8	μs
$I_{max}$	Port 1 Max Sink Current	Summed over all Port 1 bits		60	mA

### Notes:

1. All pins specified for >200 mA positive and negative injection, except P1.0 is specified for >50 mA negative injection.
2. Cext at  $V_{CC}$  or Gnd, Port 0 and Port1 at  $V_{CC}$ .
3. Part powers up in suspend mode, able to be reset by USB Bus Reset.
4. POR may re-occur whenever  $V_{CC}$  drops to approximately 2.5V.
5. Level guaranteed for range of  $V_{CC} = 4.35V$  to 5.25V.
6. With  $R_{pu1}$  of 1.5 kΩ ± 5% on D- to 3.3V regulator.
7. Maximum matched capacitive loading allowed on D+ and D- (including USB cable and host/hub) is approximately 230 pF.
8.  $I_{range} = I_{sink(F)}/I_{sink(0)}$  for each port 0 or 1 output.

## 8.0 Electrical Characteristics $f_{OSC} = 6 \text{ MHz}$ ; Operating Temperature = 0 to 70°C, $V_{CC} = 4.0$ to 5.25V (continued)

Parameter	Description	Conditions	Min.	Max.	Unit
$P_{max}$	Port 1 & Cext Sink Mode Dissipation	Per pin		25	mW
$V_{ith}$	Input Threshold Voltage	All ports and Cext <sup>[13]</sup>	45%	65%	$V_{CC}$
$V_H$	Input Hysteresis Voltage	Port 0 and Port 1 <sup>[14]</sup>	6%	12%	$V_{CC}$
$V_{HCext}$	Input Hysteresis Voltage, Cext	Cext Pin Only <sup>[14]</sup>	12%	30%	$V_{CC}$
$I_{in}$	Input Leakage Current, GPIO Pins	Port 0 and Port 1, $V_{out} = 0$ or $V_{CC}$ <sup>[15]</sup>	-1	1	$\mu A$
$I_{inCx}$	Input Leakage Current, Cext Pin	$V_{Cext} = 0$ or $V_{CC}$		50	nA
$I_{Cext}$	Sink Current, Cext Pin	$V_{Cext} = V_{CC}$	6	18	mA
$V_{ol1}$	Output LOW Voltage, Cext Pin	$V_{CC} = \text{Min.}$ , $I_{ol} = 2 \text{ mA}$		0.4	V
$V_{ol2}$	Output LOW Voltage, Cext Pin	$V_{CC} = \text{Min.}$ , $I_{ol} = 5 \text{ mA}$		2.0	V

## 9.0 Switching Characteristics

Parameter	Description	Conditions	Min.	Max.	Unit
<b>Clock</b>					
$t_{CYC}$	Input Clock Cycle Time		166.67	166.67	ns
$t_{CH}$	Clock HIGH Time		$0.45 t_{CYC}$		ns
$t_{CL}$	Clock LOW Time		$0.45 t_{CYC}$		ns
<b>USB Driver Characteristics</b>					
$t_r$	USB Data Transition Rise Time	See Notes 5, 6, and 9	75	300	ns
$t_f$	USB Data Transition Fall Time	See Notes 5, 6, and 9	75	300	ns
$t_{rfm}$	Rise/Fall Time Matching	$t_r/t_f$	80	125	%
$V_{crs}$	Output Signal Crossover Voltage	See Note 5	1.3	2.0	V
<b>USB Data Timing</b>					
$t_{drate}$	Low Speed Data Rate	Ave. Bit Rate ( $1.5 \text{ Mb/s} \pm 1.5\%$ )	1.4775	1.5225	Mb/s
$t_{djr1}$	Receiver Data Jitter Tolerance	To Next Transition, <i>Figure 9-3</i> <sup>[10]</sup>	-75	75	ns
$t_{djr2}$	Receiver Data Jitter Tolerance	For Paired Transitions, <i>Figure 9-3</i> <sup>[10]</sup>	-45	45	ns
$t_{deop}$	Differential to EOP Transition Skew	<i>Figure 9-4</i> <sup>[10]</sup>	-40	100	ns
$t_{eopr}$	EOP Width at Receiver	Accepts as EOP <sup>[10]</sup>	670		ns
$t_{lst}$	Width of SE0 Interval During Differential Transition			210	ns
$t_{eopt}$	Source EOP Width		1.25	1.50	$\mu s$
$t_{udj1}$	Differential Driver Jitter	To next transition, <i>Figure 9-5</i>	-95	95	ns
$t_{udj2}$	Differential Driver Jitter	To paired transition, <i>Figure 9-5</i>	-150	150	ns

### Notes:

9.  $C_{load}$  of 200 (75 ns) to 600 pF (300 ns).
10. Measured at crossover point of differential data signals.
11. Measured as largest step size vs. nominal according to measured full scale and zero programmed values
12.  $T_{ratio} = I_{sink1(n)}/I_{sink0(n)}$  for the same n.
13. Low to High transition.
14. This parameter is guaranteed, but not tested.
15. With Ports configured in Hi-Z mode.

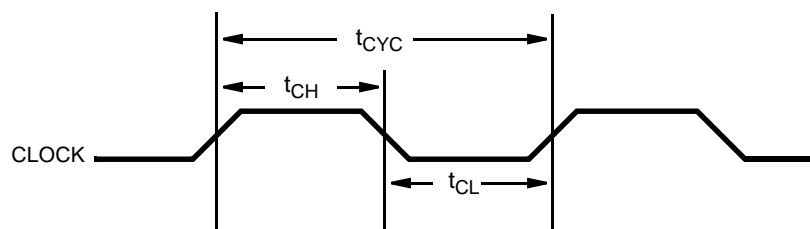


Figure 9-1. Clock Timing

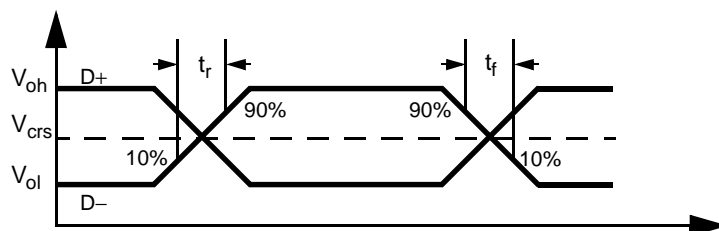


Figure 9-2. USB Data Signal Timing and Voltage Levels

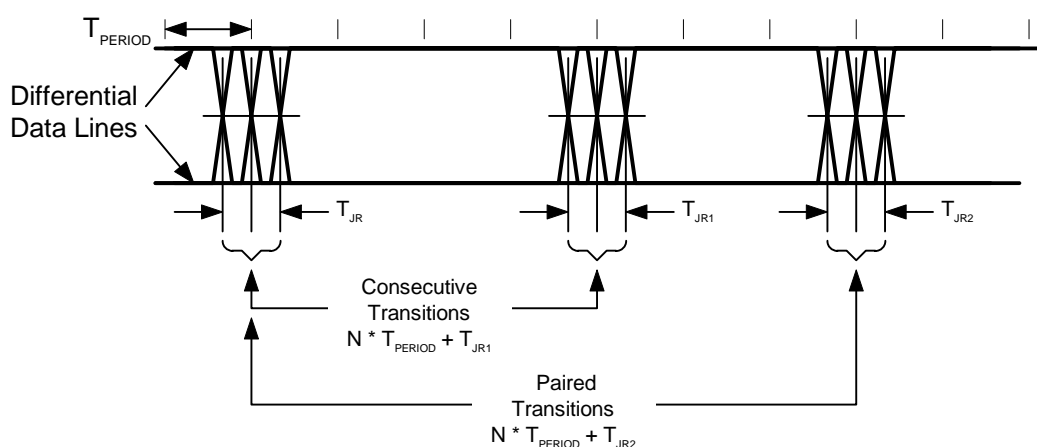


Figure 9-3. Receiver Jitter Tolerance

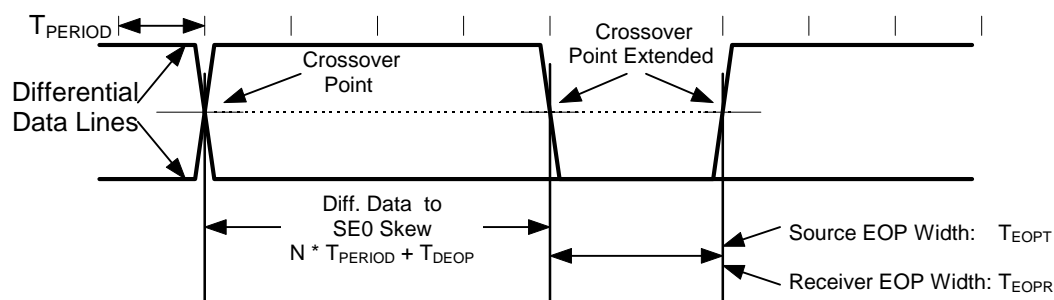
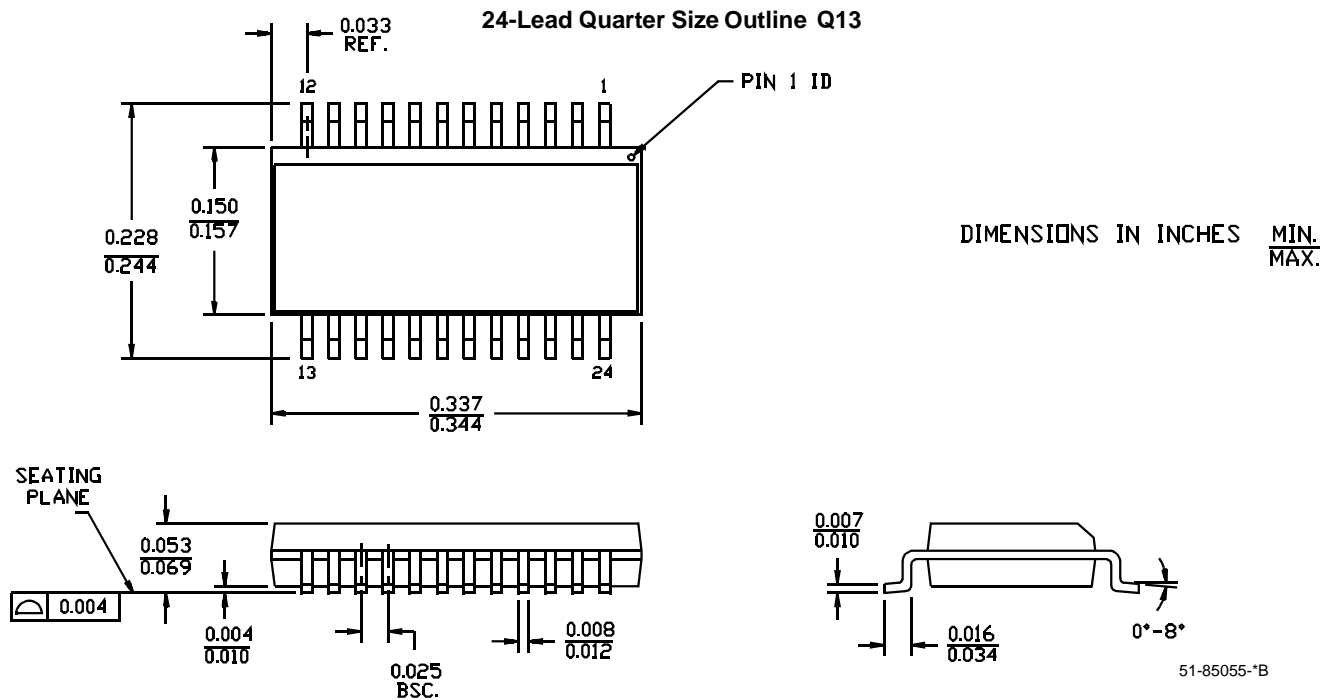
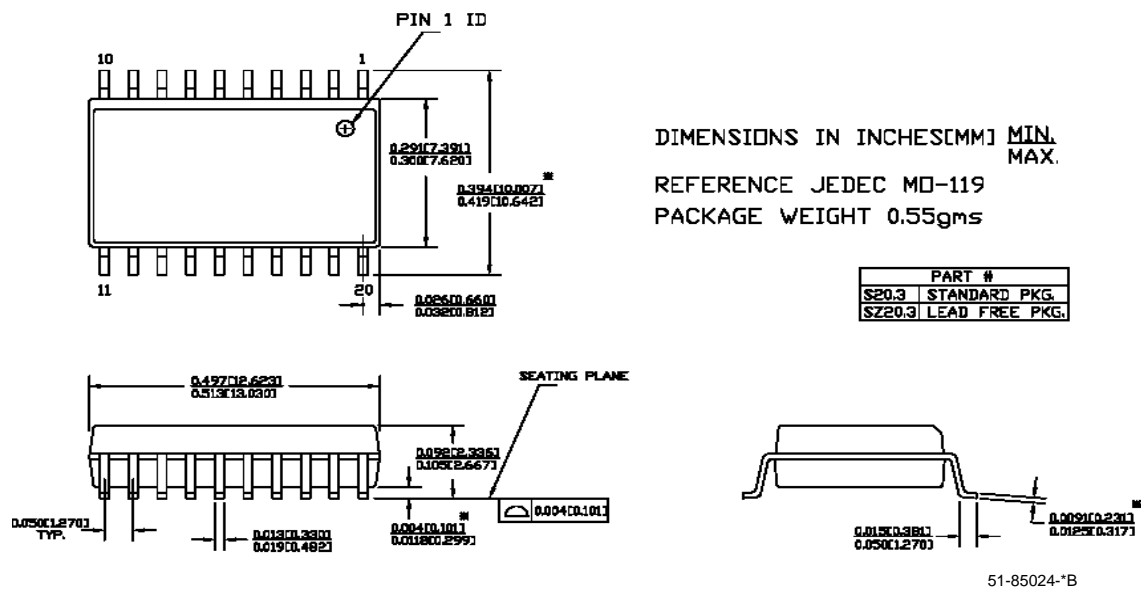


Figure 9-4. Differential to EOP Transition Skew and EOP Width

## 11.0 Package Diagrams (continued)



### 20 Lead (300 Mil) SOIC - S5



**Document History Page**

Document Title: CY7C63001A, CY7C63101A Universal Serial Bus Microcontroller Document Number: 38-08026				
REV.	ECN NO.	Issue Date	Orig. of Change	Description of Change
**	116223	06/12/02	DSG	Change from Spec number: 38-00662 to 38-08026
*A	276070	See ECN	BON	Added die form and bond pad information. Added lead-free packages. Removed obsolete packages and their references.