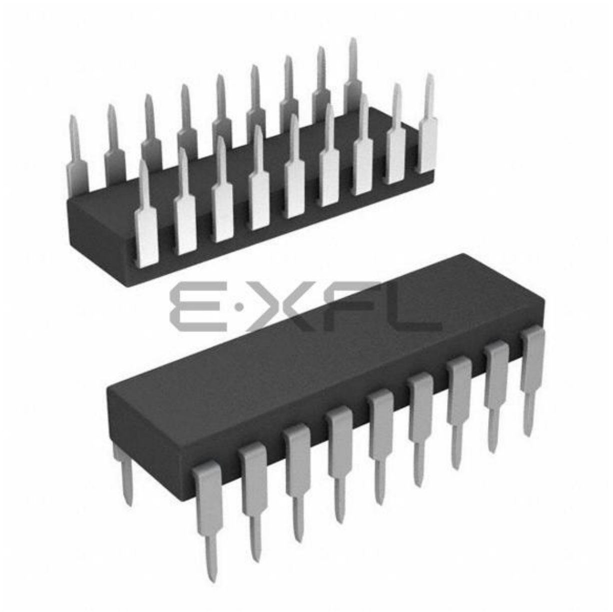
Infineon Technologies - CY7C63723-PC Datasheet



Welcome to E-XFL.COM

Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

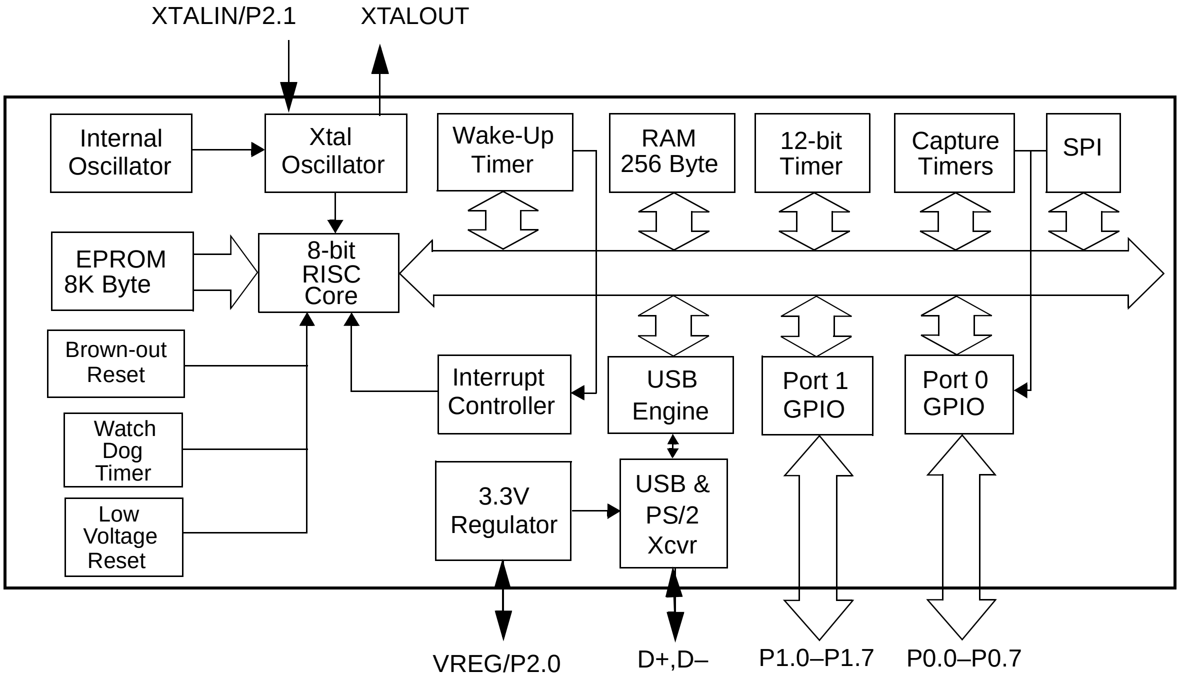
Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering 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	M8B
Program Memory Type	OTP (8kB)
Controller Series	CY7C637xx
RAM Size	256 x 8
Interface	PS/2, USB
Number of I/O	10
Voltage - Supply	4V ~ 5.5V
Operating Temperature	0°C ~ 70°C
Mounting Type	Through Hole
Package / Case	18-DIP (0.300", 7.62mm)
Supplier Device Package	18-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy7c63723-pc

2.0 Logic Block Diagram



3.0 Functional Overview

3.1 enCoRe USB—The New USB Standard

Cypress has reinvented its leadership position in the low-speed USB market with a new family of innovative microcontrollers. Introducing...enCoRe USB—“enhanced Component Reduction.” Cypress has leveraged its design expertise in USB solutions to create a new family of low-speed USB microcontrollers that enables peripheral developers to design new products with a minimum number of components. At the heart of the enCoRe USB technology is the breakthrough design of a crystalless oscillator. By integrating the oscillator into our chip, an external crystal or resonator is no longer needed. We have also integrated other external components commonly found in low-speed USB applications such as pull-up resistors, wake-up circuitry, and a 3.3V regulator. All of this adds up to a lower system cost.

The CY7C637xx is an 8-bit RISC one-time-programmable (OTP) microcontroller. The instruction set has been optimized specifically for USB and PS/2 operations, although the microcontrollers can be used for a variety of other embedded applications.

The CY7C637xx features up to 16 GPIO pins to support USB, PS/2 and other applications. The I/O pins are grouped into two ports (Port 0 to 1) where each pin can be individually configured as inputs with internal pull-ups, open drain outputs, or traditional CMOS outputs with programmable drive strength of up to 50 mA output drive. Additionally, each I/O pin can be used to generate a GPIO interrupt to the microcontroller. Note the GPIO interrupts all share the same “GPIO” interrupt vector.

The CY7C637xx microcontrollers feature an internal oscillator. With the presence of USB traffic, the internal oscillator can be set to precisely tune to USB timing requirements (6 MHz ±1.5%). Optionally, an external 6-MHz ceramic resonator can be used to provide a higher precision reference for USB operation. This clock generator reduces the clock-related noise emissions (EMI). The clock generator provides the 6- and 12-MHz clocks that remain internal to the microcontroller.

The CY7C637xx has 8 Kbytes of EPROM and 256 bytes of data RAM for stack space, user variables, and USB FIFOs.

These parts include low-voltage reset logic, a Watchdog timer, a vectored interrupt controller, a 12-bit free-running timer, and capture timers. The low-voltage reset (LVR) logic detects when power is applied to the device, resets the logic to a known state, and begins executing instructions at EPROM address 0x0000. LVR will also reset the part when V_{CC} drops below the operating voltage range. The Watchdog timer can be used to ensure the firmware never gets stalled for more than approximately 8 ms.

The microcontroller supports 10 maskable interrupts in the vectored interrupt controller. Interrupt sources include the USB Bus-Reset, the 128-µs and 1.024-ms outputs from the free-running timer, three USB endpoints, two capture timers, an internal wake-up timer and the GPIO ports. The timers bits cause periodic interrupts when enabled. The USB endpoints interrupt after USB transactions complete on the bus. The capture timers interrupt whenever a new timer value is saved due to a selected GPIO edge event. The GPIO ports have a level of masking to select which GPIO inputs can cause a GPIO interrupt. For additional flexibility, the input transition polarity that causes an interrupt is programmable for each GPIO pin. The interrupt polarity can be either rising or falling edge.

The free-running 12-bit timer clocked at 1 MHz provides two interrupt sources as noted above (128 µs and 1.024 ms). The timer can be used to measure the duration of an event under firmware control by reading the timer at the start and end of an event, and subtracting the two values. The four capture timers save a programmable 8 bit range of the free-running timer when a GPIO edge occurs on the two capture pins (P0.0, P0.1).

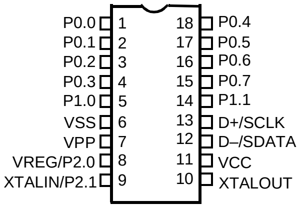
The CY7C637xx includes an integrated USB serial interface engine (SIE) that supports the integrated peripherals. The hardware supports one USB device address with three endpoints. The SIE allows the USB host to communicate with the function integrated into the microcontroller. A 3.3V regulated output pin provides a pull-up source for the external USB resistor on the D– pin.

The USB D+ and D– USB pins can alternately be used as PS/2 SCLK and SDATA signals, so that products can be designed to respond to either USB or PS/2 modes of operation. PS/2 operation is supported with internal pull-up resistors on SCLK and SDATA, the ability to disable the regulator output pin, and an interrupt to signal the start of PS/2 activity. No external components are necessary for dual USB and PS/2 systems, and no GPIO pins need to be dedicated to switching between modes. Slow edge rates operate in both modes to reduce EMI.

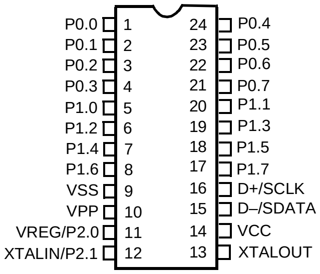
4.0 Pin Configurations

Top View

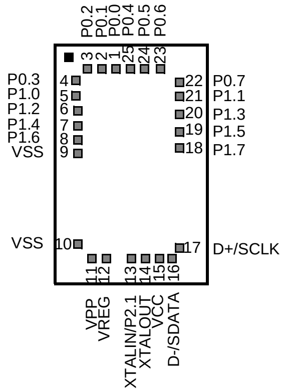
CY7C63723
18-pin SOIC/PDIP



CY7C63743
24-pin SOIC/PDIP/QSOP



CY7C63722-XC
DIE



5.0 Pin Definitions

Name	I/O	CY7C63723	CY7C63743	CY7C63722	Description
		18-Pin	24-Pin	25-Pad	
D–/SDATA, D+/SCLK	I/O	12 13	15 16	16 17	USB differential data lines (D– and D+), or PS/2 clock and data signals (SDATA and SCLK)
P0[7:0]	I/O	1, 2, 3, 4, 15, 16, 17, 18	1, 2, 3, 4, 21, 22, 23, 24	1, 2, 3, 4, 22, 23, 24, 25	GPIO Port 0 capable of sinking up to 50 mA/pin, or sinking controlled low or high programmable current. Can also source 2 mA current, provide a resistive pull-up, or serve as a high-impedance input. P0.0 and P0.1 provide inputs to Capture Timers A and B, respectively.
P1[7:0]	I/O	5, 14	5, 6, 7, 8, 17, 18, 19, 20	5, 6, 7, 8, 18, 19, 20, 21	IO Port 1 capable of sinking up to 50 mA/pin, or sinking controlled low or high programmable current. Can also source 2 mA current, provide a resistive pull-up, or serve as a high-impedance input.
XTALIN/P2.1	IN	9	12	13	6-MHz ceramic resonator or external clock input, or P2.1 input
XTALOUT	OUT	10	13	14	6-MHz ceramic resonator return pin or internal oscillator output
V_{PP}		7	10	11	Programming voltage supply, ground for normal operation
V_{CC}		11	14	15	Voltage supply
VREG/P2.0		8	11	12	Voltage supply for 1.3-kΩ USB pull-up resistor (3.3V nominal). Also serves as P2.0 input.
V_{SS}		6	9	9, 10	Ground

MNEMONIC	Operand	Opcode	Cycles		MNEMONIC	Operand	Opcode	Cycles
MOV A,[expr]	direct	1A	5		CPL		3A	4
MOV A,[X+expr]	index	1B	6		ASL		3B	4
MOV X,expr	data	1C	4		ASR		3C	4
MOV X,[expr]	direct	1D	5		RLC		3D	4
reserved		1E			RRC		3E	4
XPAGE		1F	4		RET		3F	8
MOV A,X		40	4		DI		70	4
MOV X,A		41	4		EI		72	4
MOV PSP,A		60	4		RETI		73	8
CALL	addr	50 - 5F	10					
JMP	addr	80-8F	5		JC	addr	C0-CF	5 (or 4)
CALL	addr	90-9F	10		JNC	addr	D0-DF	5 (or 4)
JZ	addr	A0-AF	5 (or 4)		JACC	addr	E0-EF	7
JNZ	addr	B0-BF	5 (or 4)		INDEX	addr	F0-FF	14

11.0 Suspend Mode

The CY7C637xx parts support a versatile low-power suspend mode. In suspend mode, only an enabled interrupt or a LOW state on the D–/SDATA pin will wake the part. Two options are available. For lowest power, all internal circuits can be disabled, so only an external event will resume operation. Alternatively, a low-power internal wake-up timer can be used to trigger the wake-up interrupt. This timer is described in Section 11.2, and can be used to periodically poll the system to check for changes, such as looking for movement in a mouse, while maintaining a low average power.

The CY7C637xx is placed into a low-power state by setting the Suspend bit of the Processor Status and Control Register (*Figure 20-1*). All logic blocks in the device are turned off except the GPIO interrupt logic, the D–/SDATA pin input receiver, and (optionally) the wake-up timer. The clock oscillators, as well as the free-running and Watchdog timers are shut down. Only the occurrence of an enabled GPIO interrupt, wake-up interrupt, SPI slave interrupt, or a LOW state on the D–/SDATA pin will wake the part from suspend (D– LOW indicates non-idle USB activity). Once one of these resuming conditions occurs, clocks will be restarted and the device returns to full operation after the oscillator is stable and the selected delay period expires. This delay period is determined by selection of internal vs. external clock, and by the state of the Ext. Clock Resume Delay as explained in Section 9.0.

In suspend mode, any enabled and pending interrupt will wake the part up. The state of the Interrupt Enable Sense bit (Bit 2, *Figure 20-1*) does not have any effect. As a result, any interrupts not intended for waking from suspend should be disabled through the Global Interrupt Enable Register and the USB End Point Interrupt Enable Register (Section 21.0).

If a resuming condition exists when the suspend bit is set, the part will still go into suspend and then awake after the appropriate delay time. The Run bit in the Processor Status and Control Register must be set for the part to resume out of suspend.

Once the clock is stable and the delay time has expired, the microcontroller will execute the instruction following the I/O write that placed the device into suspend mode before servicing any interrupt requests.

To achieve the lowest possible current during suspend mode, all I/O should be held at either V_{CC} or ground. In addition, the GPIO *bit* interrupts (*Figure 21-4* and *Figure 21-5*) should be disabled for any pins that are not being used for a wake-up interrupt. This should be done even if the main GPIO Interrupt Enable (*Figure 21-1*) is off.

Typical code for entering suspend is shown below:

```
...           ; All GPIO set to low-power state (no floating
                pins, and bit interrupts disabled unless
                using for wake-up)
...           ; Enable GPIO and/or wake-up timer
                interrupts if desired for wake-up
...           ; Select clock mode for wake-up (see
                Section 11.1)
mov a, 09h    ; Set suspend and run bits
iowr FFh      ; Write to Status and Control Register –
                Enter suspend, wait for GPIO/wake-up
                interrupt or USB activity
nop           ; This executes before any ISR
...           ; Remaining code for exiting suspend
                routine
```

11.1 Clocking Mode on Wake-up from Suspend

When exiting suspend on a wake-up event, the device can be configured to run in either Internal or External Clock mode. The mode is selected by the state of the External Oscillator Enable bit in the Clock Configuration Register (*Figure 9-2*). Using the Internal Clock saves the external oscillator start-up time and keeps that oscillator off for additional power savings. The external oscillator mode can be activated when desired, similar to operation at power-up.

The sequence of events for these modes is as follows:

Wake in Internal Clock Mode:

1. Before entering suspend, clear bit 0 of the Clock Configuration Register. This selects Internal clock mode after suspend.
2. Enter suspend mode by setting the suspend bit of the Processor Status and Control Register.
3. After a wake-up event, the internal clock starts immediately (within 2 µs).
4. A time-out period of 8 µs passes, and then firmware execution begins.
5. At some later point, to activate External Clock mode, set bit 0 of the Clock Configuration Register. This halts the internal clocks while the external clock becomes stable. After an additional time-out (128 µs or 4 ms, see Section 9.0), firmware execution resumes.

Wake in External Clock Mode:

1. Before entering suspend, the external clock must be selected by setting bit 0 of the Clock Configuration Register. Make sure this bit is still set when suspend mode is entered. This selects External clock mode after suspend.
2. Enter suspend mode by setting the suspend bit of the Processor Status and Control Register.
3. After a wake-up event, the external oscillator is started. The clock is monitored for stability (this takes approximately 50–100 µs with a ceramic resonator).
4. After an additional time-out period (128 µs or 4 ms, see Section 9.0), firmware execution resumes.

11.2 Wake-up Timer

The wake-up timer runs whenever the wake-up interrupt is enabled, and is turned off whenever that interrupt is disabled. Operation is independent of whether the device is in suspend mode or if the global interrupt bit is enabled. Only the Wake-up Timer Interrupt Enable bit (*Figure 21-1*) controls the wake-up timer.

Once this timer is activated, it will give interrupts after its time-out period (see below). These interrupts continue periodically until the interrupt is disabled. Whenever the interrupt is disabled, the wake-up timer is reset, so that a subsequent enable always results in a full wake-up time.

The wake-up timer can be adjusted by the user through the Wake-up Timer Adjust bits in the Clock Configuration Register (*Figure 9-2*). These bits clear on reset. In addition to allowing the user to select a range for the wake-up time, a firmware algorithm can be used to tune out initial process and operating condition variations in this wake-up time. This can be done by timing the wake-up interrupt time with the accurate 1.024-ms timer interrupt, and adjusting the Timer Adjust bits accordingly to approximate the desired wake-up time.

Table 12-1. Ports 0 and 1 Output Control Truth Table

Data Register	Mode1	Mode0	Output Drive Strength	Input Threshold
0	0	0	Hi-Z	CMOS
1			Hi-Z	TTL
0	0	1	Medium (8 mA) Sink	CMOS
1			High Drive	CMOS
0	1	0	Low (2 mA) Sink	CMOS
1			Resistive	CMOS
0	1	1	High (50 mA) Sink	CMOS
1			High Drive	CMOS

12.1 Auxiliary Input Port

Port 2 serves as an auxiliary input port as shown in *Figure 12-8*. The Port 2 inputs all have TTL input thresholds.

Bit #	7	6	5	4	3	2	1	0
Bit Name	Reserved		D+ (SCLK) State	D– (SDATA) State	Reserved		P2.1 (Internal Clock Mode Only)	P2.0 VREG Pin State
Read/ Write	-	-	R	R	-	-	R	R
Reset	0	0	0	0	0	0	0	0

Figure 12-8. Port 2 Data Register (Address 0x02)

Bit [7:6]: Reserved

Bit [5:4]: D+ (SCLK) and D– (SDATA) States

The state of the D+ and D– pins can be read at Port 2 Data Register. Performing a read from the port pins returns their logic values.

1 = Port Pin is logic HIGH

0 = Port Pin is logic LOW

Bit [3:2]: Reserved

Bit 1: P2.1 (Internal Clock Mode Only)

In the Internal Clock mode, the XTALIN pin can serve as a general purpose input, and its state can be read at Port 2, Bit 1 (P2.1). See Section 9.1 for more details.

1 = Port Pin is logic HIGH

0 = Port Pin is logic LOW

Bit 0: P2.0/VREG Pin State

In PS/2 mode, the VREG pin can be used as an input and its state can be read at port P2.0. Section 15.0 for more details.

1 = Port Pin is logic HIGH

0 = Port Pin is logic LOW

13.0 USB Serial Interface Engine (SIE)

The SIE allows the microcontroller to communicate with the USB host. The SIE simplifies the interface between the microcontroller and USB by incorporating hardware that handles the following USB bus activity independently of the microcontroller:

- Translate the encoded received data and format the data to be transmitted on the bus.
- CRC checking and generation. Flag the microcontroller if errors exist during transmission.
- Address checking. Ignore the transactions not addressed to the device.
- Send appropriate ACK/NAK/STALL handshakes.
- Token type identification (SETUP, IN, or OUT). Set the appropriate token bit once a valid token is received.
- Place valid received data in the appropriate endpoint FIFOs.
- Send and update the data toggle bit (Data1/0).
- Bit stuffing/unstuffing.

Firmware is required to handle the rest of the USB interface with the following tasks:

- Coordinate enumeration by decoding USB device requests.
- Fill and empty the FIFOs.
- Suspend/Resume coordination.
- Verify and select Data toggle values.

13.1 USB Enumeration

A typical USB enumeration sequence is shown below. In this description, 'Firmware' refers to embedded firmware in the CY7C637xx controller.

1. The host computer sends a SETUP packet followed by a DATA packet to USB address 0 requesting the Device descriptor.
2. Firmware decodes the request and retrieves its Device descriptor from the program memory tables.
3. The host computer performs a control read sequence and Firmware responds by sending the Device descriptor over the USB bus, via the on-chip FIFO.
4. After receiving the descriptor, the host sends a SETUP packet followed by a DATA packet to address 0 assigning a new USB address to the device.
5. Firmware 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. Firmware decodes the request and retrieves the Device descriptor from program memory tables.
8. The host performs a control read sequence and Firmware responds by sending its Device descriptor over the USB bus.
9. The host generates control reads from the device to request the Configuration and Report descriptors.
10. Once the device receives a Set Configuration request, its functions may now be used.
11. Firmware should take appropriate action for Endpoint 1 and/or 2 transactions, which may occur from this point.

17.0 Serial Peripheral Interface (SPI)

SPI is a four-wire, full-duplex serial communication interface between a master device and one or more slave devices. The CY7C637xx SPI circuit supports byte serial transfers in either Master or Slave modes. The block diagram of the SPI circuit is shown in *Figure 17-1*. The block contains buffers for both transmit and receive data for maximum flexibility and throughput. The CY7C637xx can be configured as either an SPI Master or Slave. The external interface consists of Master-Out/Slave-In (MOSI), Master-In/Slave-Out (MISO), Serial Clock (SCK), and Slave Select ($\overline{SS}$).

SPI modes are activated by setting the appropriate bits in the SPI Control Register, as described below.

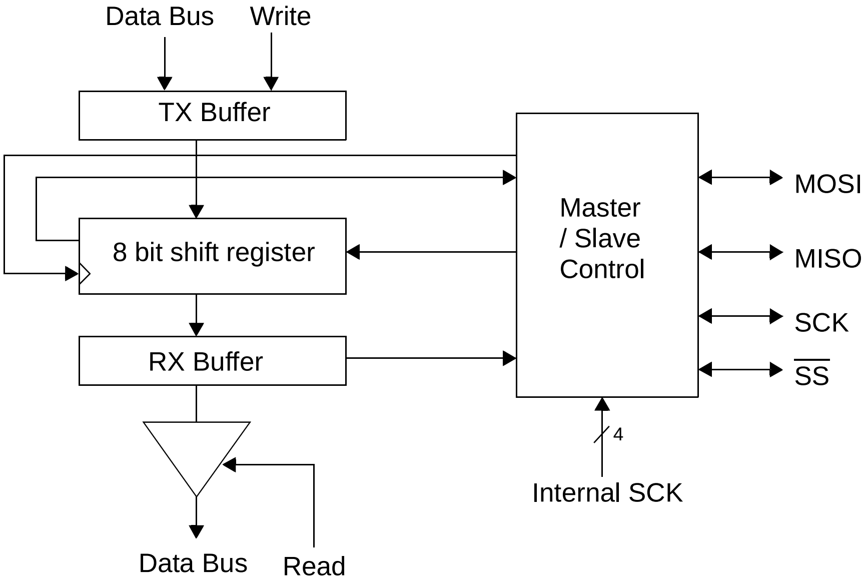


Figure 17-1. SPI Block Diagram

The SPI Data Register below serves as a transmit and receive buffer.

Bit #	7	6	5	4	3	2	1	0
Bit Name	Data I/O							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Figure 17-2. SPI Data Register (Address 0x60)

Bit [7:0]: Data I/O[7:0]

Writes to the SPI Data Register load the transmit buffer, while reads from this register read the receive buffer contents.

1 = Logic HIGH

0 = Logic LOW

17.1 Operation as an SPI Master

Only an SPI Master can initiate a byte/data transfer. This is done by the Master writing to the SPI Data Register. The Master shifts out 8 bits of data (MSB first) along with the serial clock SCK for the Slave. The Master's outgoing byte is replaced with an incoming one from a Slave device. When the last bit is received, the shift register contents are transferred to the receive buffer and an interrupt is generated. The receive data must be read from the SPI Data Register before the next byte of data is transferred to the receive buffer, or the data will be lost.

When operating as a Master, an active LOW Slave Select ($\overline{SS}$) must be generated to enable a Slave for a byte transfer. This Slave Select is generated under firmware control, and is not part of the SPI internal hardware. Any available GPIO can be used for the Master's Slave Select output.

When the Master writes to the SPI Data Register, the data is loaded into the transmit buffer. If the shift register is not busy shifting a previous byte, the TX buffer contents will be automatically transferred into the shift register and shifting will begin. If the shift register is busy, the new byte will be loaded into the shift register only after the active byte has finished and is transferred to the receive buffer. The new byte will then be shifted out. The Transmit Buffer Full (TBF) bit will be set HIGH until the transmit buffer's data-byte is transferred to the shift register. Writing to the transmit buffer while the TBF bit is HIGH will overwrite the old byte in the transmit buffer.

The byte shifting and SCK generation are handled by the hardware (based on firmware selection of the clock source). Data is shifted out on the MOSI pin (P0.5) and the serial clock is output on the SCK pin (P0.7). Data is received from the slave on the MISO pin (P0.6). The output pins must be set to the desired drive strength, and the GPIO data register must be set to 1 to enable a bypass mode for these pins. The MISO pin must be configured in the desired GPIO input mode. See Section 12.0 for GPIO configuration details.

17.2 Master SCK Selection

The Master's SCK is programmable to one of four clock settings, as shown in *Figure 17-1*. The frequency is selected with the Clock Select Bits of the SPI control register. The hardware provides 8 output clocks on the SCK pin (P0.7) for each byte transfer. Clock phase and polarity are selected by the CPHA and CPOL control bits (see *Figure 17-1* and *17-4*).

The master SCK duty cycle is nominally 33% in the fastest (2 Mbps) mode, and 50% in all other modes.

17.3 Operation as an SPI Slave

In slave mode, the chip receives SCK from an external master on pin P0.7. Data from the master is shifted in on the MOSI pin (P0.5), while data is being shifted out of the slave on the MISO pin (P0.6). In addition, the active LOW Slave Select must be asserted to enable the slave for transmit. The Slave Select pin is P0.4. These pins must be configured in appropriate GPIO modes, with the GPIO data register set to 1 to enable bypass mode selected for the MISO pin.

In Slave mode, writes to the SPI Data Register load the Transmit buffer. If the Slave Select is asserted ($\overline{SS}$ LOW) and the shift register is not busy shifting a previous byte, the transmit buffer contents will be automatically transferred into the shift register. If the shift register is busy, the new byte will be loaded into the shift register only after the active byte has finished and is transferred to the receive buffer. The new byte is then ready to be shifted out (shifting waits for SCK from the Master). If the Slave Select is not active when the transmit buffer is loaded, data is not transferred to the shift register until Slave Select is asserted. The Transmit Buffer Full (TBF) bit will be set to ‘1’ until the transmit buffer’s data-byte is transferred to the shift register. Writing to the transmit buffer while the TBF bit is HIGH will overwrite the old byte in the Transmit Buffer.

If the Slave Select is deasserted before a byte transfer is complete, the transfer is aborted and no interrupt is generated. Whenever Slave Select is asserted, the transmit buffer is automatically reloaded into the shift register.

Clock phase and polarity must be selected to match the SPI master, using the CPHA and CPOL control bits (see *Figure 17-3* and *Figure 17-4*).

The SPI slave logic continues to operate in suspend, so if the SPI interrupt is enabled, the device can go into suspend during a SPI slave transaction, and it will wake up at the interrupt that signals the end of the byte transfer.

17.4 SPI Status and Control

The SPI Control Register is shown in *Figure 17-3*. The timing diagram in *Figure 17-4* shows the clock and data states for the various SPI modes.

Bit #	7	6	5	4	3	2	1	0
Bit Name	TCMP	TBF	Comm Mode[1:0]		CPOL	CPHA	SCK Select	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Figure 17-3. SPI Control Register (Address 0x61)

Bit 7: TCMP

1 = TCMP is set to 1 by the hardware when 8-bit transfer is complete. The SPI interrupt is asserted at the same time TCMP is set to 1.

0 = This bit is only cleared by firmware.

Bit 6: TBF

Transmit Buffer Full bit.

1 = Indicates data in the transmit buffer has not transferred to the shift register.

0 = Indicates data in the transmit buffer has transferred to the shift register.

Bit [5:4] Comm Mode[1:0]

00 = All communications functions disabled (default).

01 = SPI Master Mode.

10 = SPI Slave Mode.

11 = Reserved.

Bit 3: CPOL

SPI Clock Polarity bit.

1 = SCK idles HIGH.

0 = SCK idles LOW.

Bit 2: CPHA

SPI Clock Phase bit (see *Figure 17-4*)

Bit [1:0]: SCK Select

Master mode SCK frequency selection (no effect in Slave Mode):

00 = 2 Mbit/s

01 = 1 Mbit/s

10 = 0.5 Mbit/s

11 = 0.0625 Mbit/s

18.0 12-bit Free-running Timer

The 12-bit timer operates with a 1-µs tick, provides two interrupts (128-µs and 1.024-ms) and allows the firmware to directly time events that are up to 4 ms in duration. The lower eight bits of the timer can be read directly by the firmware. Reading the lower eight bits latches the upper four bits into a temporary register. When the firmware reads the upper four bits of the timer, it is actually reading the count stored in the temporary register. The effect of this is to ensure a stable 12-bit timer value can be read, even when the two reads are separated in time.

Bit #	7	6	5	4	3	2	1	0
Bit Name	Timer [7:0]							
Read/Write	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Figure 18-1. Timer LSB Register (Address 0x24)

Bit [7:0]: Timer lower eight bits

Bit #	7	6	5	4	3	2	1	0
Bit Name	Reserved				Timer [11:8]			
Read/Write	-	-	-	-	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Figure 18-2. Timer MSB Register (Address 0x25)

Bit [7:4]: Reserved

Bit [3:0]: Timer upper four bits

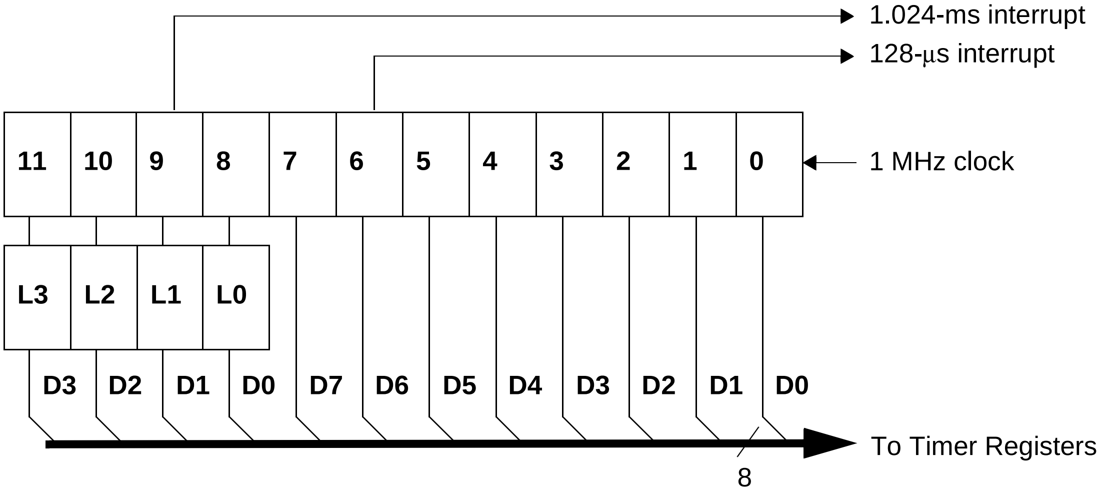


Figure 18-3. Timer Block Diagram

19.0 Timer Capture Registers

Four 8-bit capture timer registers provide both rising- and falling-edge event timing capture on two pins. Capture Timer A is connected to Pin 0.0, and Capture Timer B is connected to Pin 0.1. These can be used to mark the time at which a rising or falling event occurs at the two GPIO pins. Each timer will capture eight bits of the free-running timer into its Capture Timer Data Register if a rising or falling edge event that matches the specified rising or falling edge condition at the pin. A prescaler allows selection of the capture timer tick size. Interrupts can be individually enabled for the four capture registers. A block diagram is shown in *Figure 19-1*.

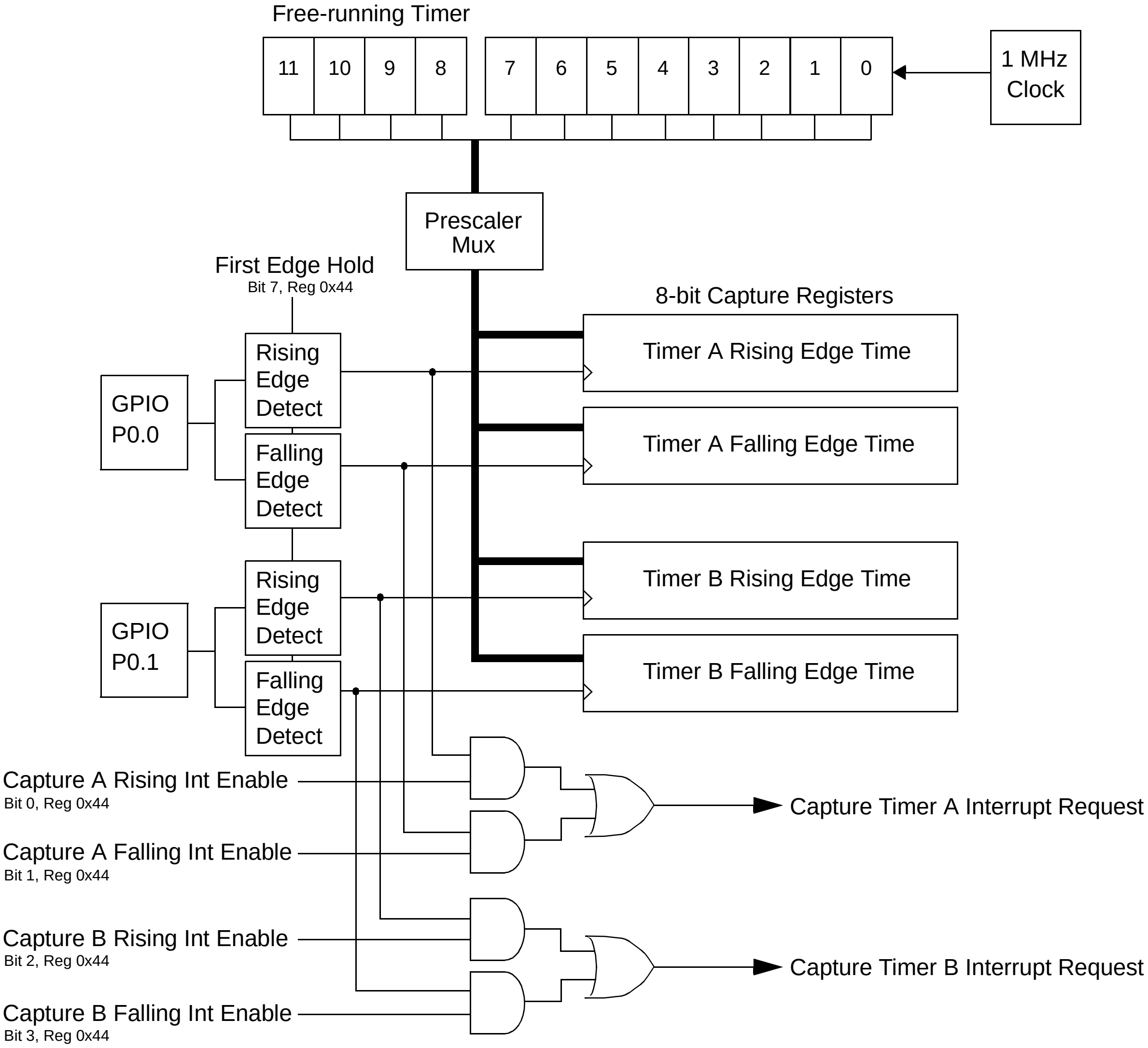


Figure 19-1. Capture Timers Block Diagram

The four Capture Timer Data Registers are read-only, and are shown in *Figure 19-2* through *Figure 19-5*.

Out of the 12-bit free running timer, the 8-bit captured in the Capture Timer Data Registers are determined by the Prescale Bit [2:0] in the Capture Timer Configuration Register (*Figure 19-7*).

Bit #	7	6	5	4	3	2	1	0
Bit Name	Capture A Rising Data							
Read/Write	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Figure 19-2. Capture Timer A-Rising, Data Register (Address 0x40)

Bit #	7	6	5	4	3	2	1	0
Bit Name	Capture A Falling Data							
Read/Write	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Figure 19-3. Capture Timer A-Falling, Data Register (Address 0x41)

Bit #	7	6	5	4	3	2	1	0
Bit Name	Capture B Rising Data							
Read/Write	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Figure 19-4. Capture Timer B-Rising, Data Register (Address 0x42)

Bit #	7	6	5	4	3	2	1	0
Bit Name	Capture B Falling Data							
Read/Write	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Figure 19-5. Capture Timer B-Falling, Data Register (Address 0x43)

Bit #	7	6	5	4	3	2	1	0
Bit Name	Reserved				Capture B Falling Event	Capture B Rising Event	Capture A Falling Event	Capture A Rising Event
Read/ Write	-	-	-	-	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Figure 19-6. Capture Timer Status Register (Address 0x45)

Bit [7:4]: Reserved.

Bit [3:0]: Capture A/B, Falling/Rising Event

These bits record the occurrence of any rising or falling edges on the capture GPIO pins. Bits in this register are cleared by reading the corresponding data register.

1 = A rising or falling event that matches the pin's rising/falling condition has occurred.

0 = No event that matches the pin's rising or falling edge condition.

Because both Capture A events (rising and falling) share an interrupt, user's firmware needs to check the status of both Capture A Falling and Rising Event bits to determine what caused the interrupt. This is also true for Capture B events.

Bit #	7	6	5	4	3	2	1	0
Bit Name	First Edge Hold	Prescale Bit [2:0]			Capture B Falling Int Enable	Capture B Rising Int Enable	Capture A Falling Int Enable	Capture A Rising Int Enable
Read/ Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Figure 19-7. Capture Timer Configuration Register (Address 0x44)

Bit 7: First Edge Hold

1 = The time of the first occurrence of an edge is held in the Capture Timer Data Register until the data is read. Subsequent edges are ignored until the Capture Timer Data Register is read.

0 = The time of the most recent edge is held in the Capture Timer Data Register. That is, if multiple edges have occurred before reading the capture timer, the time for the last one will be read (default state).

The First Edge Hold function applies globally to all four capture timers.

Bit [6:4]: Prescale Bit [2:0]

Three prescaler bits allow the capture timer clock rate to be selected among 5 choices, as shown in *Table 19-1* below.

Bit [3:0]: Capture A/B, Rising/Falling Interrupt Enable

Each of the four Capture Timer registers can be individually enabled to provide interrupts.

Both Capture A events share a common interrupt request, as do the two Capture B events. In addition to the event enables, the main Capture Interrupt Enables bit in the Global Interrupt Enable register (Section 21.0) must be set to activate a capture interrupt.

1 = Enable interrupt

0 = Disable interrupt

Table 19-1. Capture Timer Prescalar Settings (Step size and range for F_{CLK} = 6 MHz)

Prescale 2:0	Captured Bits	LSB Step Size	Range
000	Bits 7:0 of free-running timer	1 μs	256 μs
001	Bits 8:1 of free-running timer	2 μs	512 μs
010	Bits 9:2 of free-running timer	4 μs	1.024 ms
011	Bits 10:3 of free-running timer	8 μs	2.048 ms
100	Bits 11:4 of free-running timer	16 μs	4.096 ms

21.3 Interrupt Sources

The following sections provide details on the different types of interrupt sources.

Bit #	7	6	5	4	3	2	1	0
Bit Name	Wake-up Interrupt Enable	GPIO Interrupt Enable	Capture Timer B Intr. Enable	Capture Timer A Intr. Enable	SPI Interrupt Enable	1.024-ms Interrupt Enable	128-µs Interrupt Enable	USB Bus Reset / PS/2 Activity Intr. Enable
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Figure 21-1. Global Interrupt Enable Register (Address 0x20)

Bit 7: Wake-up Interrupt Enable

The internal wake-up timer is normally used to wake the part from suspend mode, but it can also provide an interrupt when the part is awake. The wake-up timer is cleared whenever the Wake-up Interrupt Enable bit is written to a 0, and runs whenever that bit is written to a 1. When the interrupt is enabled, the wake-up timer provides periodic interrupts at multiples of period, as described in Section 11.2.

1 = Enable wake-up timer for periodic wake-up.

0 = Disable and power-off wake-up timer.

Bit 6: GPIO Interrupt Enable

Each GPIO pin can serve as an interrupt input. During a reset, GPIO interrupts are disabled by clearing all GPIO interrupt enable registers. Writing a ‘1’ to a GPIO Interrupt Enable bit enables GPIO interrupts from the corresponding input pin. These registers are shown in *Figure 21-4* for Port 0 and *Figure 21-5* for Port 1. In addition to enabling the desired individual pins for interrupt, the main GPIO interrupt must be enabled, as explained in Section 21.0.

The polarity that triggers an interrupt is controlled independently for each GPIO pin by the GPIO Interrupt Polarity Registers. Setting a Polarity bit to ‘0’ allows an interrupt on a falling GPIO edge, while setting a Polarity bit to ‘1’ allows an interrupt on a rising GPIO edge. The Polarity Registers reset to 0 and are shown in *Figure 21-6* for Port 0 and *Figure 21-7* for Port 1.

All of the GPIO pins share a single interrupt vector, which means the firmware will need to read the GPIO ports with enabled interrupts to determine which pin or pins caused an interrupt.The GPIO interrupt structure is illustrated in *Figure 21-8*.

Note that if one port pin triggered an interrupt, no other port pins can cause a GPIO interrupt until that port pin has returned to its inactive (non-trigger) state or its corresponding port interrupt enable bit is cleared. The CY7C637xx does not assign interrupt priority to different port pins and the Port Interrupt Enable Registers are not affected by the interrupt acknowledge process.

1 = Enable

0 = Disable

Bit [5:4]: Capture Timer A and B Interrupts

There are two capture timer interrupts, one for each associated pin. Each of these interrupts occurs on an enabled edge of the selected GPIO pin(s). For each pin, rising and/or falling edge capture interrupts can be in selected. Refer to Section 19.0. These interrupts are independent of the GPIO interrupt, described in the next section.

1 = Enable

0 = Disable

Bit 3: SPI Interrupt Enable

The SPI interrupt occurs at the end of each SPI byte transaction, at the final clock edge, as shown in *Figure 17-4*. After the interrupt, the received data byte can be read from the SPI Data Register, and the TCMP control bit will be high

1 = Enable

0 = Disable

Bit 2: 1.024-ms Interrupt Enable

The 1.024-ms interrupts are periodic timer interrupts from the free-running timer (based on the 6-MHz clock). The user should disable this interrupt before going into the suspend mode to avoid possible conflicts between servicing the timer interrupts (128-µs interrupt and 1.024-ms interrupt) first or the suspend request first when waking up.

1 = Enable. Periodic interrupts will be generated approximately every 1.024 ms.

0 = Disable.

Bit 1: 128-µs Interrupt Enable

The 128-µs interrupt is another source of timer interrupt from the free-running timer. The user should disable both timer interrupts (128-µs and 1.024-ms) before going into the suspend mode to avoid possible conflicts between servicing the timer interrupts first or the suspend request first when waking up.

1 = Enable. Periodic interrupts will be generated approximately every 128 µs.

0 = Disable.

Bit 0: USB Bus Reset - PS/2 Interrupt Enable

The function of this interrupt is selectable between detection of either a USB bus reset condition, or PS/2 activity. The selection is made with the USB-PS/2 Interrupt Mode bit in the USB Status and Control Register (*Figure 13-1*). In either case, the interrupt will occur if the selected condition exists for 256 µs, and may occur as early as 128 µs.

Bit [7:0]: P1[7:0] Interrupt Polarity

1 = Rising GPIO edge

0 = Falling GPIO edge

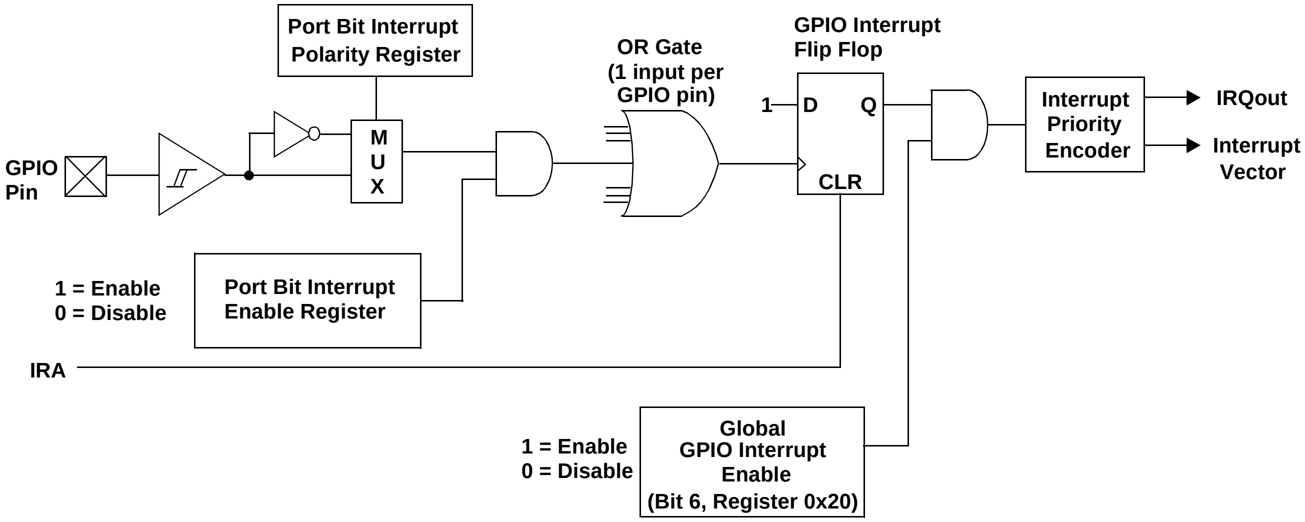

Port Bit Interrupt Polarity Register
GPIO Pin
M U X
OR Gate (1 input per GPIO pin)
GPIO Interrupt Flip Flop
1
D
Q
CLR
Interrupt Priority Encoder
IRQout
Interrupt Vector
1 = Enable
0 = Disable
Port Bit Interrupt Enable Register
IRA
1 = Enable
0 = Disable
Global GPIO Interrupt Enable (Bit 6, Register 0x20)

Figure 21-8. GPIO Interrupt Diagram

22.0 USB Mode Tables

The following tables give details on mode setting for the USB Serial Interface Engine (SIE) for both the control endpoint (EP0) and non-control endpoints (EP1 and EP2).

Table 22-1. USB Register Mode Encoding for Control and Non-Control Endpoints

Mode	Encoding	SETUP	IN	OUT	Comments
Disable	**0000**	Ignore	Ignore	Ignore	Ignore all USB traffic to this endpoint
NAK IN/OUT	**0001**	Accept	NAK	NAK	On Control endpoint, after successfully sending an ACK handshake to a SETUP packet, the SIE forces the endpoint mode (from modes other than 0000) to 0001. The mode is also changed by the SIE to 0001 from mode 1011 on issuance of ACK handshake to an OUT.
Status OUT Only	**0010**	Accept	STALL	Check	For Control endpoints
STALL IN/OUT	**0011**	Accept	STALL	STALL	For Control endpoints
Ignore IN/OUT	**0100**	Accept	Ignore	Ignore	For Control endpoints
Reserved	**0101**	Ignore	Ignore	Always	Reserved
Status IN Only	**0110**	Accept	TX 0 Byte	STALL	For Control Endpoints
Reserved	**0111**	Ignore	TX Count	Ignore	Reserved
NAK OUT	**1000**	Ignore	Ignore	NAK	In mode 1001, after sending an ACK handshake to an OUT, the SIE changes the mode to 1000
ACK OUT(STALL[3]=0) ACK OUT(STALL[3]=1)	**1001** **1001**	Ignore Ignore	Ignore Ignore	ACK STALL	This mode is changed by the SIE to mode 1000 on issuance of ACK handshake to an OUT
NAK OUT - Status IN	**1010**	Accept	TX 0 Byte	NAK	
ACK OUT - NAK IN	**1011**	Accept	NAK	ACK	This mode is changed by the SIE to mode 0001 on issuance of ACK handshake to an OUT
NAK IN	**1100**	Ignore	NAK	Ignore	An ACK from mode 1101 changes the mode to 1100
ACK IN(STALL[3]=0) ACK IN(STALL[3]=1)	**1101** **1101**	Ignore Ignore	TX Count STALL	Ignore Ignore	This mode is changed by the SIE to mode 1100 on issuance of ACK handshake to an IN
NAK IN - Status OUT	**1110**	Accept	NAK	Check	An ACK from mode 1111 changes the mode to 1110
ACK IN - Status OUT	**1111**	Accept	TX Count	Check	This mode is changed by the SIE to mode 1110 on issuance of ACK handshake to an IN

Note:

[3] STALL bit is the bit 7 of the USB Non-Control Device Endpoint Mode registers. Refer to Section 14.3 for more explanation.

Mode Column:

The ’Mode’ column contains the mnemonic names given to the modes of the endpoint. The mode of the endpoint is determined by the four-bit binaries in the ’Encoding’ column as discussed below. The Status IN and Status OUT modes represent the status IN or OUT stage of the control transfer.

Encoding Column:

The contents of the ’Encoding’ column represent the Mode Bits [3:0] of the Endpoint Mode Registers (*Figure 14-2* and *Figure 14-3*). The endpoint modes determine how the SIE responds to different tokens that the host sends to the endpoints. For example, if the Mode Bits [3:0] of the Endpoint 0 Mode Register (*Figure 14-2*) are set to ’0001’, which is NAK IN/OUT mode as shown in *Table 22-1* above, the SIE of the part will send an ACK handshake in response to SETUP tokens and NAK any IN or OUT tokens. For more information on the functionality of the Serial Interface Engine (SIE), see Section 13.0.

SETUP, IN, and OUT Columns:

Depending on the mode specified in the ’Encoding’ column, the ’SETUP’, ’IN’, and ’OUT’ columns contain the device SIE’s responses when the endpoint receives SETUP, IN, and OUT tokens respectively.

A ’Check’ in the Out column means that upon receiving an OUT token the SIE checks to see whether the OUT is of zero length and has a Data Toggle (Data1/0) of 1. If these conditions are true, the SIE responds with an ACK. If any of the above conditions is not met, the SIE will respond with either a STALL or Ignore. *Table 22-3* gives a detailed analysis of all possible cases.

A ’TX Count’ entry in the IN column means that the SIE will transmit the number of bytes specified in the Byte Count Bit [3:0] of the Endpoint Count Register (*Figure 14-4*) in response to any IN token.

A ’TX 0 Byte’ entry in the IN column means that the SIE will transmit a zero byte packet in response to any IN sent to the endpoint. Sending a 0 byte packet is to complete the status stage of a control transfer.

An ’Ignore’ means that the device sends no handshake tokens.

An ’Accept’ means that the SIE will respond with an ACK to a valid SETUP transaction.

Comments Column:

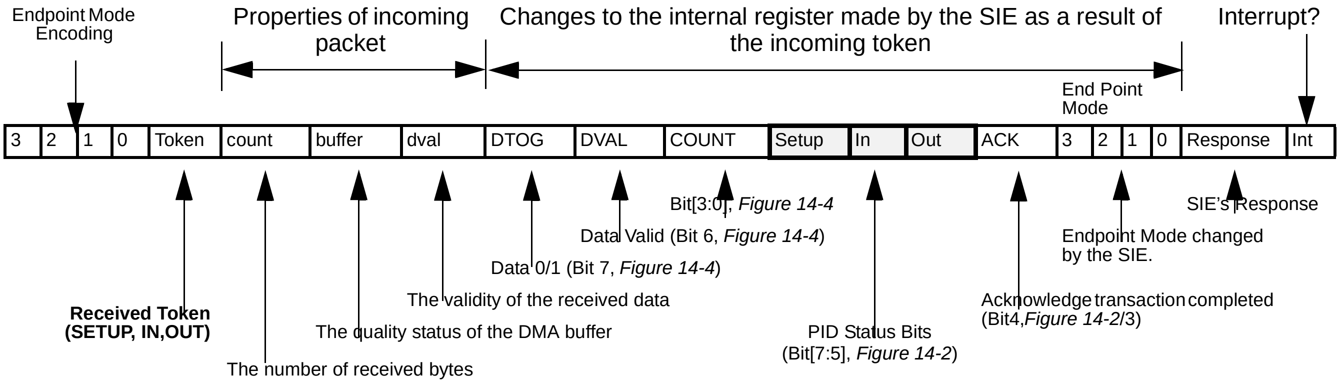
Some Mode Bits are automatically changed by the SIE in response to many USB transactions. For example, if the Mode Bits [3:0] are set to ’1111’ which is ACK IN-Status OUT mode as shown in *Table 22-1*, the SIE will change the endpoint Mode Bits [3:0] to NAK IN-Status OUT mode (1110) after ACKing a valid status stage OUT token. The firmware needs to update the mode for the SIE to respond appropriately. See *Table 22-1* for more details on what modes will be changed by the SIE.

Any SETUP packet to an enabled endpoint with mode set to accept SETUPs will be changed by the SIE to 0001 (NAKing). Any mode set to accept a SETUP will send an ACK handshake to a valid SETUP token.

A disabled endpoint will remain disabled until changed by firmware, and all endpoints reset to the Disabled mode (0000). Firmware normally enables the endpoint mode after a SetConfiguration request.

The control endpoint has three status bits for identifying the token type received (SETUP, IN, or OUT), but the endpoint must be placed in the correct mode to function as such. Non-control endpoints should not be placed into modes that accept SETUPs.

Table 22-2. Decode table for *Table 22-3*: “Details of Modes for Differing Traffic Conditions”


Endpoint Mode Encoding
Properties of incoming packet
Changes to the internal register made by the SIE as a result of the incoming token
Interrupt?
End Point Mode
3
2
1
0
Token
count
buffer
dval
DTOG
DVAL
COUNT
Setup
In
Out
ACK
3
2
1
0
Response
Int
Bit[3:0], Figure 14-4
SIE’s Response
Data Valid (Bit 6, Figure 14-4)
Endpoint Mode changed by the SIE.
Data 0/1 (Bit 7, Figure 14-4)
The validity of the received data
Acknowledge transaction completed (Bit4,Figure 14-2/3)
Received Token (SETUP, IN,OUT)
The quality status of the DMA buffer
PID Status Bits (Bit[7:5], Figure 14-2)
The number of received bytes

Legend:

UC: unchanged	TX: transmit	TX0: transmit 0-length packet
x: don’t care	RX: receive	

▢ available for Control endpoint only

Table 22-3. Details of Modes for Differing Traffic Conditions (continued)

Control Read																				
ACK IN/Status OUT																				
1	1	1	1	OUT	2	UC	valid	1	1	updates	UC	UC	1	1	NoChange				ACK	yes
1	1	1	1	OUT	2	UC	valid	0	1	updates	UC	UC	1	UC	0	0	1	1	STALL	yes
1	1	1	1	OUT	!=2	UC	valid	updates	1	updates	UC	UC	1	UC	0	0	1	1	STALL	yes
1	1	1	1	OUT	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
1	1	1	1	OUT	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
1	1	1	1	IN	x	UC	x	UC	UC	UC	UC	1	UC	1	1	1	1	0	ACK (back)	yes
NAK IN/Status OUT																				
1	1	1	0	OUT	2	UC	valid	1	1	updates	UC	UC	1	1	NoChange				ACK	yes
1	1	1	0	OUT	2	UC	valid	0	1	updates	UC	UC	1	UC	0	0	1	1	STALL	yes
3	2	1	0	token	count	buffer	dval	DTOG	DVAL	COUNT	SETUP	IN	OUT	ACK	3	2	1	0	response	int
1	1	1	0	OUT	!=2	UC	valid	updates	1	updates	UC	UC	1	UC	0	0	1	1	STALL	yes
1	1	1	0	OUT	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
1	1	1	0	OUT	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
1	1	1	0	IN	x	UC	x	UC	UC	UC	UC	1	UC	UC	NoChange				NAK	yes
Status OUT Only																				
0	0	1	0	OUT	2	UC	valid	1	1	updates	UC	UC	1	1	NoChange				ACK	yes
0	0	1	0	OUT	2	UC	valid	0	1	updates	UC	UC	1	UC	0	0	1	1	STALL	yes
0	0	1	0	OUT	!=2	UC	valid	updates	1	updates	UC	UC	1	UC	0	0	1	1	STALL	yes
0	0	1	0	OUT	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
0	0	1	0	OUT	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
0	0	1	0	IN	x	UC	x	UC	UC	UC	UC	1	UC	UC	0	0	1	1	STALL	yes
OUT Endpoint																				
ACK OUT, STALL Bit = 0 (*Figure 14-3*)																				
1	0	0	1	OUT	<= 10	data	valid	updates	1	updates	UC	UC	1	1	1	0	0	0	ACK	yes
1	0	0	1	OUT	> 10	junk	x	updates	updates	updates	UC	UC	1	UC	NoChange				Ignore	yes
1	0	0	1	OUT	x	junk	invalid	updates	0	updates	UC	UC	1	UC	NoChange				Ignore	yes
1	0	0	1	IN	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
ACK OUT, STALL Bit = 1 (*Figure 14-3*)																				
1	0	0	1	OUT	<= 10	UC	valid	UC	UC	UC	UC	UC	1	UC	NoChange				STALL	yes
1	0	0	1	OUT	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
1	0	0	1	OUT	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
1	0	0	1	IN	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
NAK OUT																				
1	0	0	0	OUT	<= 10	UC	valid	UC	UC	UC	UC	UC	1	UC	NoChange				NAK	yes
1	0	0	0	OUT	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
1	0	0	0	OUT	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
1	0	0	0	IN	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
Reserved																				
0	1	0	1	OUT	x	updates	updates	updates	updates	updates	UC	UC	1	1	NoChange				RX	yes
0	1	0	1	IN	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
IN Endpoint																				
ACK IN, STALL Bit = 0 (*Figure 14-3*)																				
1	1	0	1	OUT	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
1	1	0	1	IN	x	UC	x	UC	UC	UC	UC	1	UC	1	1	1	0	0	ACK (back)	yes
ACK IN, STALL Bit = 1 (*Figure 14-3*)																				
1	1	0	1	OUT	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				Ignore	no
1	1	0	1	IN	x	UC	x	UC	UC	UC	UC	1	UC	UC	NoChange				STALL	yes
NAK IN																				

24.0 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
Maximum Total Sink Output Current into Port 0 and 1 and Pins	70 mA
Maximum Total Source Output Current into Port 0 and 1 and Pins	30 mA
Maximum On-chip Power Dissipation on any GPIO Pin	50 mW
Power Dissipation	300 mW
Static Discharge Voltage	> 2000V
Latch-up Current	> 200 mA

25.0 DC Characteristics FOSC = 6 MHz; Operating Temperature = 0 to 70°C

	Parameter	Conditions	Min.	Max.	Unit
	General				
V_{CC1}	Operating Voltage	Note 4	V_{LVR}	5.5	V
V_{CC2}	Operating Voltage	Note 4	4.35	5.25	V
I_{CC1}	V_{CC} Operating Supply Current – Internal Oscillator Mode	V_{CC} = 5.5V, no GPIO loading		20	mA
	Typical I_{CC1} = 16 mA[5]	V_{CC} = 5.0V. T = Room Temperature			
I_{CC2}	V_{CC} Operating Supply Current – External Oscillator Mode	V_{CC} = 5.5V, no GPIO loading		17	mA
	Typical I_{CC2}= 13 mA[5]	V_{CC} = 5.0V. T = Room Temperature			
I_{SB1}	Standby Current – No Wake-up Osc	Oscillator off, D– > 2.7V		25	µA
I_{SB2}	Standby Current – With Wake-up Osc	Oscillator off, D– > 2.7V		75	µA
V_{PP}	Programming Voltage (disabled)		–0.4	0.4	V
T_{RSNTR}	Resonator Start-up Interval	V_{CC} = 5.0V, ceramic resonator		256	µs
I_{IL}	Input Leakage Current	Any I/O pin		1	µA
I_{SNK}	Max I_{SS} GPIO Sink Current	Cumulative across all ports[6]		70	mA
I_{SRC}	Max I_{CC} GPIO Source Current	Cumulative across all ports[6]		30	mA
	Low-Voltage and Power-on Reset				
V_{LVR}	Low-Voltage Reset Trip Voltage	V_{CC} below V_{LVR} for >100 ns[7]	3.5	4.0	V
t_{VCCS}	V_{CC} Power-on Slew Time	linear ramp: 0 to 4V[8]		100	ms
	USB Interface				
V_{REG}	VREG Regulator Output Voltage	Load = R_{PU} +R_{PD}[9, 10]	3.0	3.6	V
C_{REG}	Capacitance on VREG Pin	External cap not required		300	pF
V_{OHU}	Static Output High, driven	R_{PD} to Gnd[4]	2.8	3.6	V

Notes:

[4] Full functionality is guaranteed in V_{CC1} range, except USB transmitter specifications and GPIO output currents are guaranteed for V_{CC2} range.

[5] Bench measurements taken under nominal operating conditions. Spec cannot be guaranteed at final test.

[6] Total current cumulative across all Port pins, limited to minimize Power and Ground-Drop noise effects.

[7] LVR is automatically disabled during suspend mode.

[8] LVR will re-occur whenever V_{CC} drops below V_{LVR}. In suspend or with LVR disabled, BOR occurs whenever V_{CC} drops below approximately 2.5V.

[9] V_{REG} specified for regulator enabled, idle conditions (i.e., no USB traffic), with load resistors listed. During USB transmits from the internal SIE, the VREG output is not regulated, and should not be used as a general source of regulated voltage in that case. During receive of USB data, the VREG output drops when D– is LOW due to internal series resistance of approximately 200Ω at the VREG pin.

[10] In suspend mode, V_{REG} is only valid if R_{PU} is connected from D– to VREG pin, and R_{PD} is connected from D– to ground.

26.0 Switching Characteristics

Parameter	Description	Conditions	Min.	Max.	Unit
	Internal Clock Mode				
F_{ICLK}	Internal Clock Frequency	Internal Clock Mode enabled	5.7	6.3	MHz
F_{ICLK2}	Internal Clock Frequency, USB mode	Internal Clock Mode enabled, Bit 2 of register 0xF8h is set (Precision USB Clocking)[12]	5.91	6.09	MHz
	External Oscillator Mode				
T_{CYC}	Input Clock Cycle Time	USB Operation, with External ±1.5% Ceramic Resonator or Crystal	164.2	169.2	ns
T_{CH}	Clock HIGH Time		0.45 t_{CYC}		ns
T_{CL}	Clock LOW Time		0.45 t_{CYC}		ns
	Reset Timing				
t_{START}	Time-out Delay after LVR/BOR		24	60	ms
t_{WAKE}	Internal Wake-up Period	Enabled Wake-up Interrupt[13]	1	5	ms
t_{WATCH}	WatchDog Timer Period	F_{OSC} = 6 MHz	10.1	14.6	ms
	USB Driver Characteristics				
T_R	Transition Rise Time	CLoad = 200 pF (10% to 90%[4])	75		ns
T_R	Transition Rise Time	CLoad = 600 pF (10% to 90%[4])		300	ns
T_F	Transition Fall Time	CLoad = 200 pF (10% to 90%[4])	75		ns
T_F	Transition Fall Time	CLoad = 600 pF (10% to 90%[4])		300	ns
T_{RFM}	Rise/Fall Time Matching	t_r/t_f[4, 14]	80	125	%
V_{CRS}	Output Signal Crossover Voltage[18]	CLoad = 200 to 600 pF[4]	1.3	2.0	V
	USB Data Timing				
T_{DRATE}	Low Speed Data Rate	Ave. Bit Rate (1.5 Mb/s ±1.5%)	1.4775	1.5225	Mb/s
T_{DJR1}	Receiver Data Jitter Tolerance	To Next Transition[15]	–75	75	ns
T_{DJR2}	Receiver Data Jitter Tolerance	For Paired Transitions[15]	–45	45	ns
T_{DEOP}	Differential to EOP transition Skew	Note 15	–40	100	ns
T_{EOPR2}	EOP Width at Receiver	Accepts as EOP[15]	670		ns
T_{EOPT}	Source EOP Width		1.25	1.50	µs
T_{UDJ1}	Differential Driver Jitter	To next transition, *Figure 26-5*	–95	95	ns
T_{UDJ2}	Differential Driver Jitter	To paired transition, *Figure 26-5*	–150	150	ns
T_{LST}	Width of SE0 during Diff. Transition			210	ns
	Non-USB Mode Driver Characteristics	Note 16			
T_{FPS2}	SDATA/SCK Transition Fall Time	CLoad = 150 pF to 600 pF	50	300	ns
	SPI Timing	See *Figures 26-6 to 26-9*[17]			
T_{SMCK}	SPI Master Clock Rate	F_{CLK}/3; see *Figure 17-1*		2	MHz
T_{SSCK}	SPI Slave Clock Rate			2.2	MHz

Notes:

12. Initially $F_{ICLK2} = F_{ICLK}$ until a USB packet is received.
13. Wake-up time for Wake-up Adjust Bits cleared to 000b (minimum setting)
14. Tested at 200 pF.
15. Measured at cross-over point of differential data signals.
16. Non-USB Mode refers to driving the D–/SDATA and/or D+/SCLK pins with the Control Bits of the USB Status and Control Register, with Control Bit 2 HIGH.
17. SPI timing specified for capacitive load of 50 pF, with GPIO output mode = 01 (medium low drive, strong high drive).
18. Per the USB 2.0 Specification, Table 7.7, Note 10, the first transition from the Idle state is excluded.

26.0 Switching Characteristics (continued)

Parameter	Description	Conditions	Min.	Max.	Unit
T_{SCKH}	SPI Clock High Time	High for CPOL = 0, Low for CPOL = 1	125		ns
T_{SCKL}	SPI Clock Low Time	Low for CPOL = 0, High for CPOL = 1	125		ns
T_{MDO}	Master Data Output Time	SCK to data valid	–25	50	ns
T_{MDO1}	Master Data Output Time, First bit with CPHA = 1	Time before leading SCK edge	100		ns
T_{MSU}	Master Input Data Set-up time		50		ns
T_{MHD}	Master Input Data Hold time		50		ns
T_{SSU}	Slave Input Data Set-up Time		50		ns
T_{SHD}	Slave Input Data Hold Time		50		ns
T_{SDO}	Slave Data Output Time	SCK to data valid		100	ns
T_{SDO1}	Slave Data Output Time, First bit with CPHA = 1	Time after $\overline{SS}$ LOW to data valid		100	ns
T_{SSS}	Slave Select Set-up Time	Before first SCK edge	150		ns
T_{SSH}	Slave Select Hold Time	After last SCK edge	150		ns

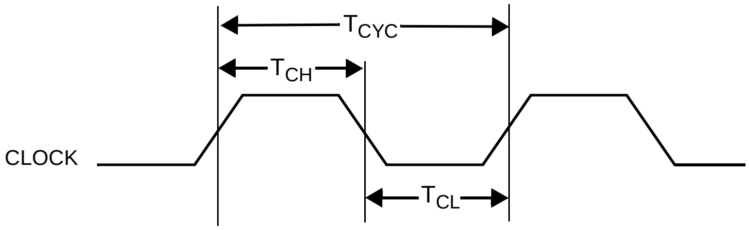


Figure 26-1. Clock Timing

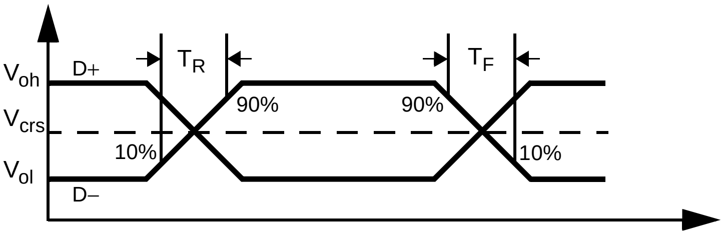


Figure 26-2. USB Data Signal Timing

27.0 Ordering Information

Ordering Code	EPROM Size	Package Name	Package Type	Operating Range
CY7C63723-PC	8 KB	P3	18-Pin (300-Mil) PDIP	Commercial
CY7C63723-PXC	8 KB	P3	18-Pin (300-Mil) Lead-free PDIP	Commercial
CY7C63723-SC	8 KB	S3	18-Pin Small Outline Package	Commercial
CY7C63723-SXC	8 KB	S3	18-Pin Small Outline Lead-free Package	Commercial
CY7C63743-QXC	8 KB	Q13	24 QSOP Lead-free Package	Commercial
CY7C63743-PC	8 KB	P13	24-Pin (300-Mil) PDIP	Commercial
CY7C63743-PXC	8 KB	P13	24-Pin (300-Mil) Lead-free PDIP	Commercial
CY7C63743-SC	8 KB	S13	24-Pin Small Outline Package	Commercial
CY7C63743-SXC	8 KB	S13	24-Pin Small Outline Lead-free Package	Commercial
CY7C63722-XC	8 KB	–	25-Pad DIE Form	Commercial
CY7C63722-XWC	8 KB	–	25-Pad DIE Form Lead-free	Commercial

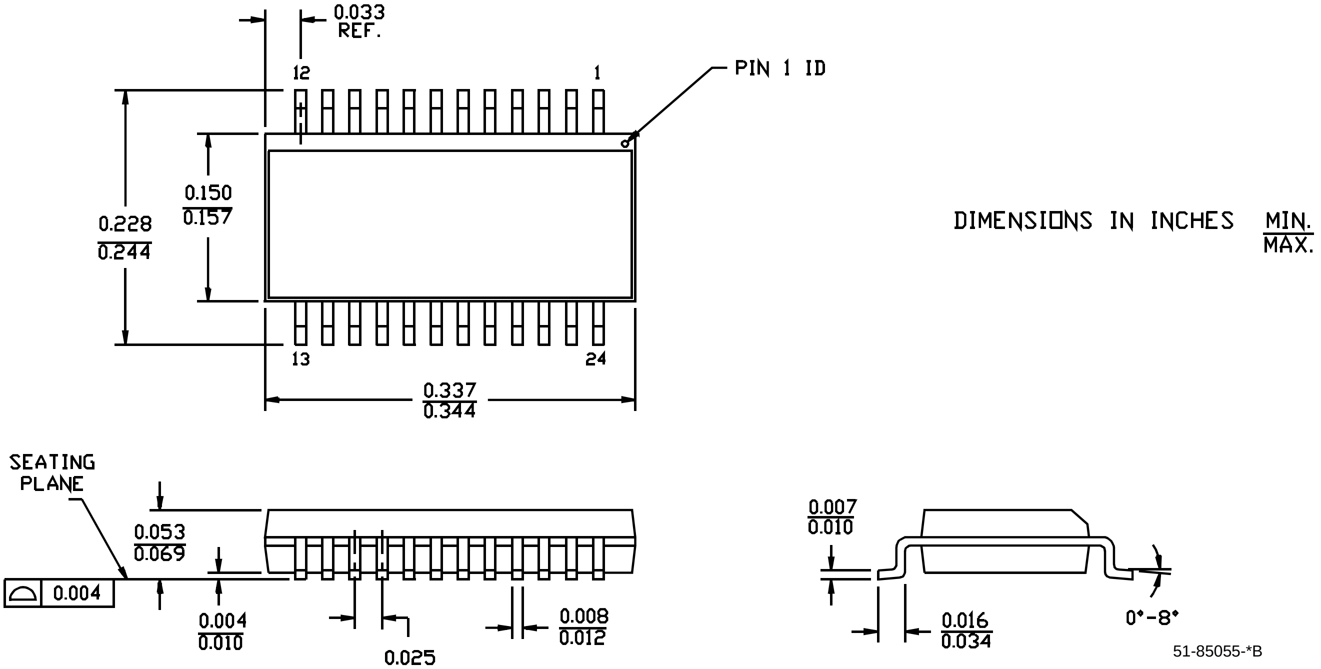
28.0 Package Diagrams

18-Lead (300-Mil) Molded DIP P3

51-85010-*A

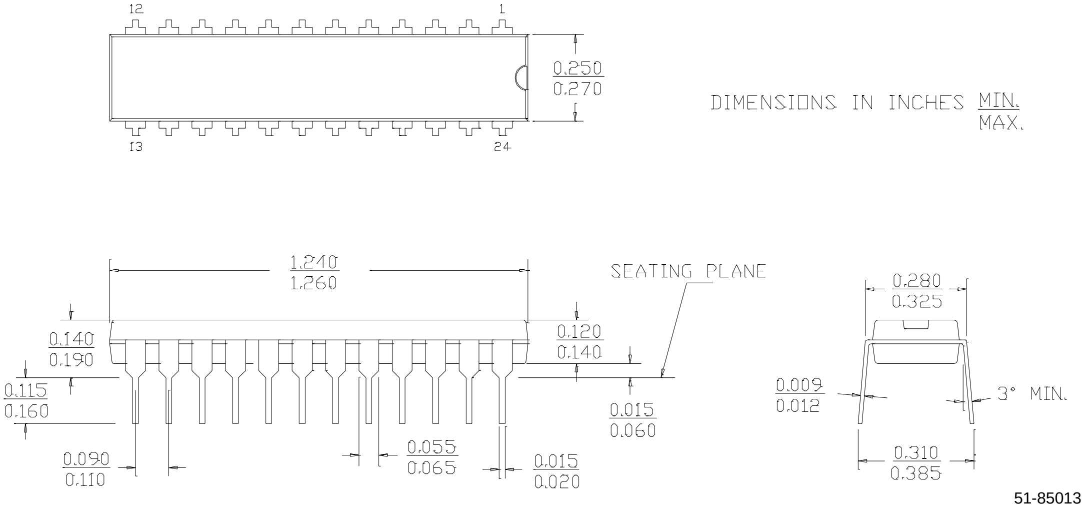
28.0 Package Diagrams (continued)

24-Lead Quarter Size Outline Q13



51-85055-*B

24-Lead (300-Mil) PDIP P13



51-85013-*B

Document History Page

Document Title: CY7C63722, CY7C63723, CY7C63743 enCoRe™ USB Combination Low-Speed USB and PS/2 Peripheral Controller
Document Number: 38-08022

REV.	ECN NO.	Issue Date	Orig. of Change	Description of Change
**	118643	10/22/02	BON	Converted from Spec 38-00944 to Spec 38-08022. Added notes 17, 18 to section 26 Removed obsolete parts (63722-PC and 63742) Added die sale Added section 23 (Register Summary)
*A	243308	SEE ECN	KKU	Added 24 QSOP package Added Lead-free packages to section 27 Reformatted to update format
*B	267229	See ECN	ARI	Corrected part number in the Ordering Information section