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

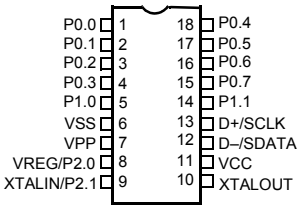
Details	
Product Status	Obsolete
Applications	USB Microcontroller
Core Processor	M8B
Program Memory Type	OTP (8kB)
Controller Series	CY7C637xx
RAM Size	256 x 8
Interface	PS/2, USB
Number of I/O	16
Voltage - Supply	3.5V ~ 5.5V
Operating Temperature	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	24-SOIC (0.295", 7.50mm Width)
Supplier Device Package	24-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy7c63743c-sxct

Contents

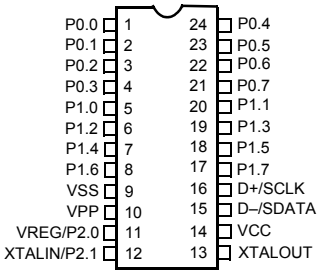
Pin Configurations

Top View

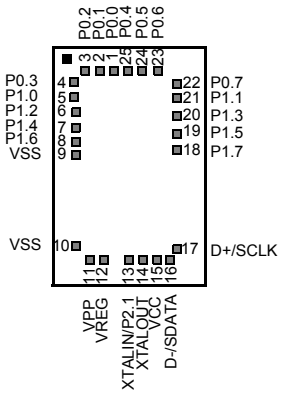
CY7C63723C
18-pin SOIC/PDIP



CY7C63743C
24-pin SOIC/PDIP/QSOP



CY7C63722C-XC
DIE



Pin Definitions

Name	I/O	CY7C63723C	CY7C63743C	CY7C63722C	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

■ DSPINIT: EQU 30h

■ MOV A,DSPINIT

Direct

“Direct” address mode is used when the data operand is a variable stored in SRAM. In that case, the one byte address of the variable is encoded in the instruction. As an example, consider an instruction that loads A with the contents of memory address location 0x10h:

■ MOV A, [10h]

In normal usage, variable names are assigned to variable addresses using “EQU” statements to improve the readability of the assembler source code. As an example, the following code is equivalent to the example shown above.

■ buttons: EQU 10h

■ MOV A, [buttons]

Indexed

“Indexed” address mode allows the firmware to manipulate arrays of data stored in SRAM. The address of the data operand is the sum of a constant encoded in the instruction and the contents of the “X” register. In normal usage, the constant will be the “base” address of an array of data and the X register will contain an index that indicates which element of the array is actually addressed.

■ array: EQU 10h

■ MOV X,3

■ MOV A, [x+array]

This would have the effect of loading A with the fourth element of the SRAM “array” that begins at address 0x10h. The fourth element would be at address 0x13h.

Instruction Set Summary

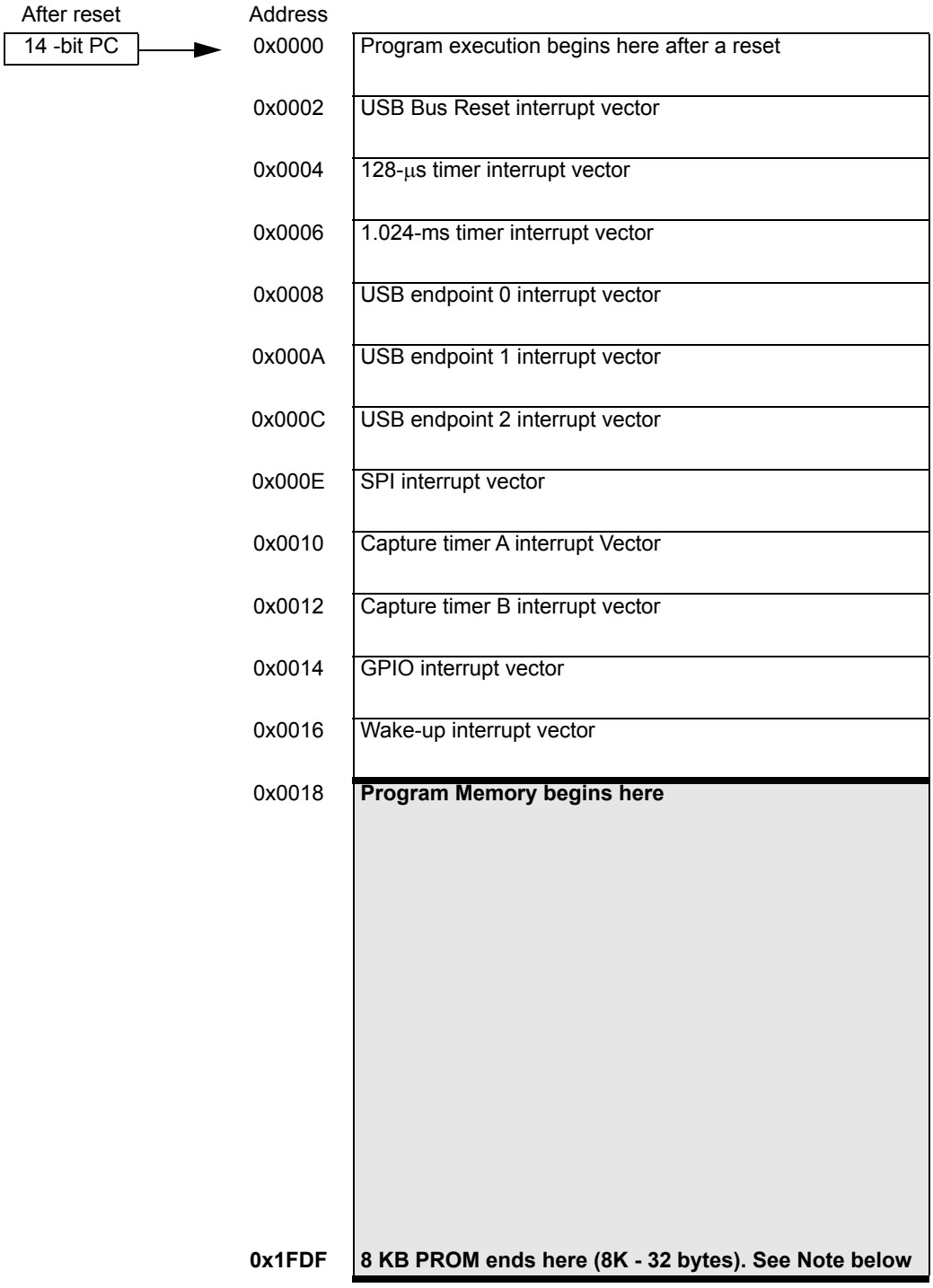
Refer to the *CYASM Assembler User’s Guide* for detailed information on these instructions. Note that conditional jump instructions (i.e., JC, JNC, JZ, JNZ) take five cycles if jump is taken, four cycles if no jump.

MNEMONIC	Operand	Opcode	Cycles		MNEMONIC	Operand	Opcode	Cycles
HALT		00	7		NOP		20	4
ADD A,expr	data	01	4		INC A	acc	21	4
ADD A,[expr]	direct	02	6		INC X	x	22	4
ADD A,[X+expr]	index	03	7		INC [expr]	direct	23	7
ADC A,expr	data	04	4		INC [X+expr]	index	24	8
ADC A,[expr]	direct	05	6		DEC A	acc	25	4
ADC A,[X+expr]	index	06	7		DEC X	x	26	4
SUB A,expr	data	07	4		DEC [expr]	direct	27	7
SUB A,[expr]	direct	08	6		DEC [X+expr]	index	28	8
SUB A,[X+expr]	index	09	7		IORD expr	address	29	5
SBB A,expr	data	0A	4		IOWR expr	address	2A	5
SBB A,[expr]	direct	0B	6		POP A		2B	4
SBB A,[X+expr]	index	0C	7		POP X		2C	4
OR A,expr	data	0D	4		PUSH A		2D	5
OR A,[expr]	direct	0E	6		PUSH X		2E	5
OR A,[X+expr]	index	0F	7		SWAP A,X		2F	5
AND A,expr	data	10	4		SWAP A,DSP		30	5
AND A,[expr]	direct	11	6		MOV [expr],A	direct	31	5
AND A,[X+expr]	index	12	7		MOV [X+expr],A	index	32	6
XOR A,expr	data	13	4		OR [expr],A	direct	33	7
XOR A,[expr]	direct	14	6		OR [X+expr],A	index	34	8
XOR A,[X+expr]	index	15	7		AND [expr],A	direct	35	7
CMP A,expr	data	16	5		AND [X+expr],A	index	36	8
CMP A,[expr]	direct	17	7		XOR [expr],A	direct	37	7
CMP A,[X+expr]	index	18	8		XOR [X+expr],A	index	38	8
MOV A,expr	data	19	4		IOWX [X+expr]	index	39	6

Memory Organization

Program Memory Organization[2]

Figure 1. Program Memory Space with Interrupt Vector Table

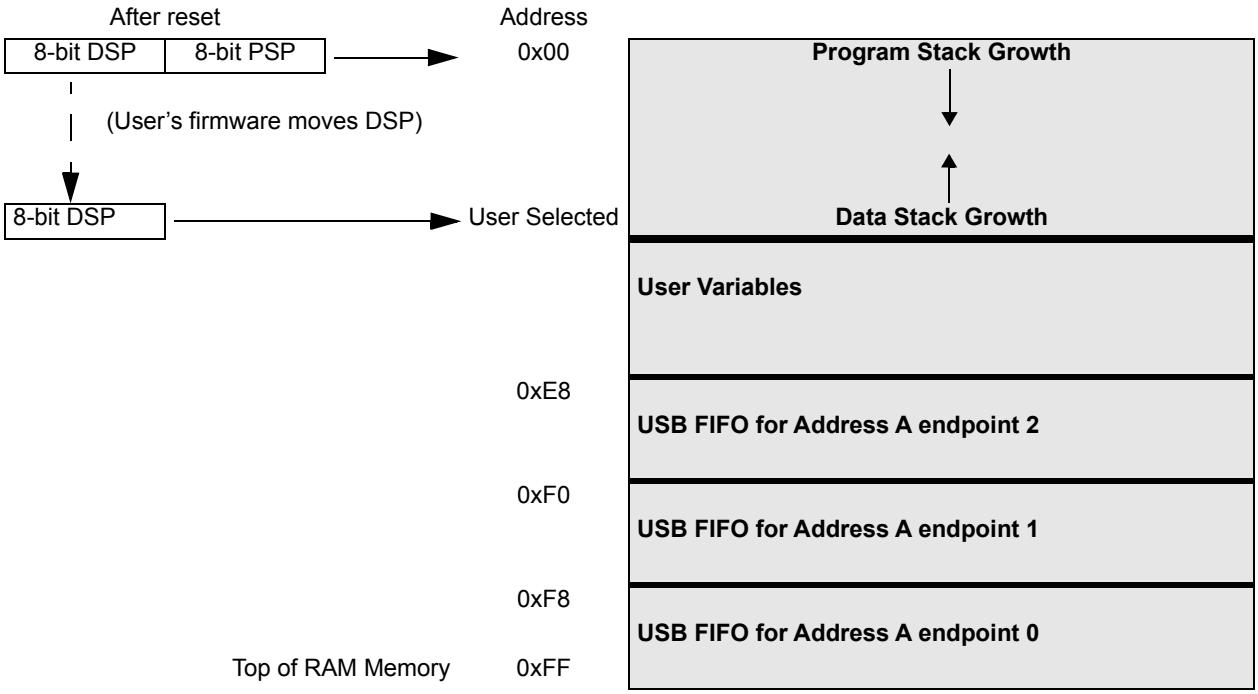


Note
2. The upper 32 bytes of the 8K PROM are reserved. Therefore, the user's program must not overwrite this space.

Data Memory Organization

The CY7C637xxC microcontrollers provide 256 bytes of data RAM. In normal usage, the SRAM is partitioned into four areas: program stack, data stack, user variables and USB endpoint FIFOs as shown below.

Figure 2. Data Memory Organization



I/O Register Summary

I/O registers are accessed via the I/O Read (IORD) and I/O Write (IOWR, IOWX) instructions. IORD reads the selected port into the accumulator. IOWR writes data from the accumulator to the selected port. Indexed I/O Write (IOWX) adds the contents of X to the address in the instruction to form the port address and writes data from the accumulator to the specified port. Note that specifying address 0 with IOWX (e.g., IOWX 0h) means the I/O port is selected solely by the contents of X.

Note: All bits of all registers are cleared to all zeros on reset, except the Processor Status and Control Register (Figure 34). *All registers not listed are reserved, and should never be written by firmware. All bits marked as reserved should always be written as 0 and be treated as undefined by reads.*

Table 1. I/O Register Summary

Register Name	I/O Address	Read/Write	Function	Fig
Port 0 Data	0x00	R/W	GPIO Port 0	7
Port 1 Data	0x01	R/W	GPIO Port 1	8
Port 2 Data	0x02	R	Auxiliary input register for D+, D–, VREG, XTALIN	
Port 0 Interrupt Enable	0x04	W	Interrupt enable for pins in Port 0	
Port 1 Interrupt Enable	0x05	W	Interrupt enable for pins in Port 1	
Port 0 Interrupt Polarity	0x06	W	Interrupt polarity for pins in Port 0	
Port 1 Interrupt Polarity	0x07	W	Interrupt polarity for pins in Port 1	
Port 0 Mode0	0x0A	W	Controls output configuration for Port 0	9
Port 0 Mode1	0x0B	W		
Port 1 Mode0	0x0C	W	Controls output configuration for Port 1	11
Port 1 Mode1	0x0D	W		12

Table 1. I/O Register Summary (continued)

Register Name	I/O Address	Read/Write	Function	Fig
USB Device Address	0x10	R/W	USB Device Address register	15
EP0 Counter Register	0x11	R/W	USB Endpoint 0 counter register	18
EP0 Mode Register	0x12	R/W	USB Endpoint 0 configuration register	
EP1 Counter Register	0x13	R/W	USB Endpoint 1 counter register	18
EP1 Mode Register	0x14	R/W	USB Endpoint 1 configuration register	
EP2 Counter Register	0x15	R/W	USB Endpoint 2 counter register	18
EP2 Mode Register	0x16	R/W	USB Endpoint 2 configuration register	
USB Status & Control	0x1F	R/W	USB status and control register	14
Global Interrupt Enable	0x20	R/W	Global interrupt enable register	
Endpoint Interrupt Enable	0x21	R/W	USB endpoint interrupt enables	
Timer (LSB)	0x24	R	Lower 8 bits of free-running timer (1 MHz)	24
Timer (MSB)	0x25	R	Upper 4 bits of free-running timer	25
WDR Clear	0x26	W	Watchdog Reset clear	-
Capture Timer A Rising	0x40	R	Rising edge Capture Timer A data register	28
Capture Timer A Falling	0x41	R	Falling edge Capture Timer A data register	
Capture Timer B Rising	0x42	R	Rising edge Capture Timer B data register	30
Capture Timer B Falling	0x43	R	Falling edge Capture Timer B data register	31
Capture TImer Configuration	0x44	R/W	Capture Timer configuration register	33
Capture Timer Status	0x45	R	Capture Timer status register	32
SPI Data	0x60	R/W	SPI read and write data register	
SPI Control	0x61	R/W	SPI status and control register	22
Clock Configuration	0xF8	R/W	Internal / External Clock configuration register	
Processor Status & Control	0xFF	R/W	Processor status and control	34

Clocking

The chip can be clocked from either the internal on-chip clock, or from an oscillator based on an external resonator/crystal, as shown in Figure 3. No additional capacitance is included on chip at the XTALIN/OUT pins. Operation is controlled by the Clock Configuration Register, Figure 4.

Figure 3. Clock Oscillator On-chip Circuit

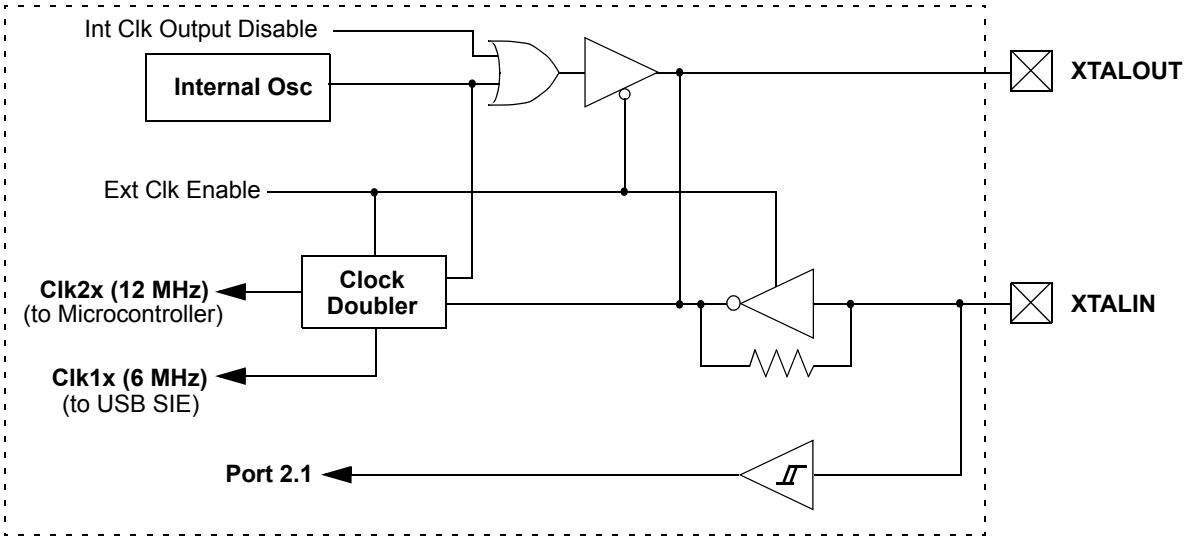


Figure 4. Clock Configuration Register (Address 0xF8)

Bit #	7	6	5	4	3	2	1	0
Bit Name	Ext. Clock Resume Delay	Wake-up Timer Adjust Bit [2:0]			Low-voltage Reset Disable	Precision USB Clocking Enable	Internal Clock Output Disable	External Oscillator 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

Bit 7: Ext. Clock Resume Delay

External Clock Resume Delay bit selects the delay time when switching to the external oscillator from the internal oscillator mode, or when waking from suspend mode with the external oscillator enabled.

1 = 4 ms delay.

0 = 128 μs delay.

The delay gives the oscillator time to start up. The shorter time is adequate for operation with ceramic resonators, while the longer time is preferred for start-up with a crystal. (These times **do not include** an initial oscillator start-up time which depends on the resonating element. This time is typically 50–100 μs for ceramic resonators and 1–10 ms for crystals). Note that this bit only selects the delay time for the external clock mode. When waking from suspend mode with the internal oscillator (Bit 0 is LOW), the delay time is only 8 μs in addition to a delay of approximately 1 μs for the oscillator to start.

Bit [6:4]: Wake-up Timer Adjust Bit [2:0]

The Wake-up Timer Adjust Bits are used to adjust the Wake-up timer period.

If the Wake-up interrupt is enabled in the Global Interrupt Enable Register, the microcontroller will generate wake-up interrupts periodically. The frequency of these periodical wake-up interrupts is adjusted by setting the Wake-up Timer Adjust Bit [2:0], as described in Wake-up Timer on page 16. One common use of the wake-up interrupts is to generate periodical wake-up events during suspend mode to check for changes, such as looking for movement in a mouse, while maintaining a low average power.

Bit 3: Low-voltage Reset Disable

When V_{CC} drops below V_{LVR} (see DC Characteristics on page 42 for the value of V_{LVR}) and the Low-voltage Reset circuit is enabled, the microcontroller enters a partial suspend state for a period of t_{START} (see Switching Characteristics on

page 44 for the value of t_{START}). Program execution begins from address 0x0000 after this t_{START} delay period. This provides time for V_{CC} to stabilize before the part executes code. See Low-voltage Reset (LVR) on page 14 for more details.

1 = Disables the LVR circuit.

0 = Enables the LVR circuit.

Bit 2: Precision USB Clocking Enable

The Precision USB Clocking Enable only affects operation in internal oscillator mode. **In that mode, this bit must be set to 1 to cause the internal clock to automatically precisely tune to USB timing requirements (6 MHz ±1.5%)**. The frequency may have a looser initial tolerance at power-up, but all USB transmissions from the chip will meet the USB specification.

1 = Enabled. The internal clock accuracy is **6 MHz ±1.5%** after USB traffic is received**.**

0 = Disabled. The internal clock accuracy is 6 MHz ±5%.

Bit 1: Internal Clock Output Disable

The Internal Clock Output Disable is used to keep the internal clock from driving out to the XTALOUT pin. This bit has no effect in the external oscillator mode.

1 = Disable internal clock output. XTALOUT pin will drive HIGH.

0 = Enable the internal clock output. The internal clock is driven out to the XTALOUT pin.

Bit 0: External Oscillator Enable

At power-up, the chip operates from the internal clock by default. Setting the External Oscillator Enable bit HIGH disables the internal clock, and halts the part while the external resonator/crystal oscillator is started. Clearing this bit has no immediate effect, although the state of this bit is used when waking out of suspend mode to select between internal and external clock. In internal clock mode, XTALIN pin will be configured as an input with a weak pull-down and can be used as a GPIO input (P2.1).

1 = Enable the external oscillator. The clock is switched to external clock mode, as described in Internal/External Oscillator Operation on page 13.

0 = Enable the internal oscillator.

Internal/External Oscillator Operation

The internal oscillator provides an operating clock, factory set to a nominal frequency of 6 MHz. This clock requires no external components. At power-up, the chip operates from the internal clock. In this mode, the internal clock is buffered and driven to the XTALOUT pin by default, and the state of the XTALIN pin can be read at Port 2.1. While the internal clock is enabled, its output can be disabled at the XTALOUT pin by setting the Internal Clock Output Disable bit of the Clock Configuration Register.

Setting the External Oscillator Enable bit of the Clock Configuration Register HIGH disables the internal clock, and halts the part while the external resonator/crystal oscillator is started. The steps involved in switching from Internal to External Clock mode are as follows:

1. At reset, chip begins operation using the internal clock.
2. Firmware sets Bit 0 of the Clock Configuration Register. For example,

```
mov A, 1h      ; Set Bit 0 HIGH (External Oscillator
                 Enable bit). Bit 7 cleared gives
                 faster start-up
iowr F8h       ; Write to Clock Configuration
                 Register
```

3. Internal clocking is halted, the internal oscillator is disabled, and the external clock oscillator is enabled.
4. After the external clock becomes stable, chip clocks are re-enabled using the external clock signal. (Note that the time for the external clock to become stable depends on the external resonating device; see next section.)
5. After an additional delay the CPU is released to run. This delay depends on the state of the Ext. Clock Resume Delay bit of the Clock Configuration Register. The time is 128 µs if the bit is 0, or 4 ms if the bit is 1.
6. Once the chip has been set to external oscillator, it can only return to internal clock when waking from suspend mode. Clearing bit 0 of the Clock Configuration Register will not re-enable internal clock mode until suspend mode is entered. See Suspend Mode on page 15 for more details on suspend mode operation.

If the Internal Clock is enabled, the XTALIN pin can serve as a general purpose input, and its state can be read at Port 2, Bit 1 (P2.1). Refer to Figure 13 on page 19 for the Port 2 Data Register. In this mode, there is a weak pull-down at the XTALIN pin. This input cannot provide an interrupt source to the CPU.

External Oscillator

The user can connect a low-cost ceramic resonator or an external oscillator to the XTALIN/XTALOUT pins to provide a precise reference frequency for the chip clock, as shown in Figure 3 on page 12. The external components required are a ceramic resonator or crystal and any associated capacitors. To run from the external resonator, the External Oscillator Enable bit of the Clock Configuration Register must be set to 1, as explained in the previous section.

Start-up times for the external oscillator depend on the resonating device. Ceramic resonator based oscillators typically start in less than 100 µs, while crystal based oscillators take longer, typically 1 to 10 ms. Board capacitance should be minimized on the XTALIN and XTALOUT pins by keeping the traces as short as possible.

An external 6-MHz clock can be applied to the XTALIN pin if the XTALOUT pin is left open.

Reset

The USB Controller supports three types of resets. The effects of the reset are listed below. The reset types are:

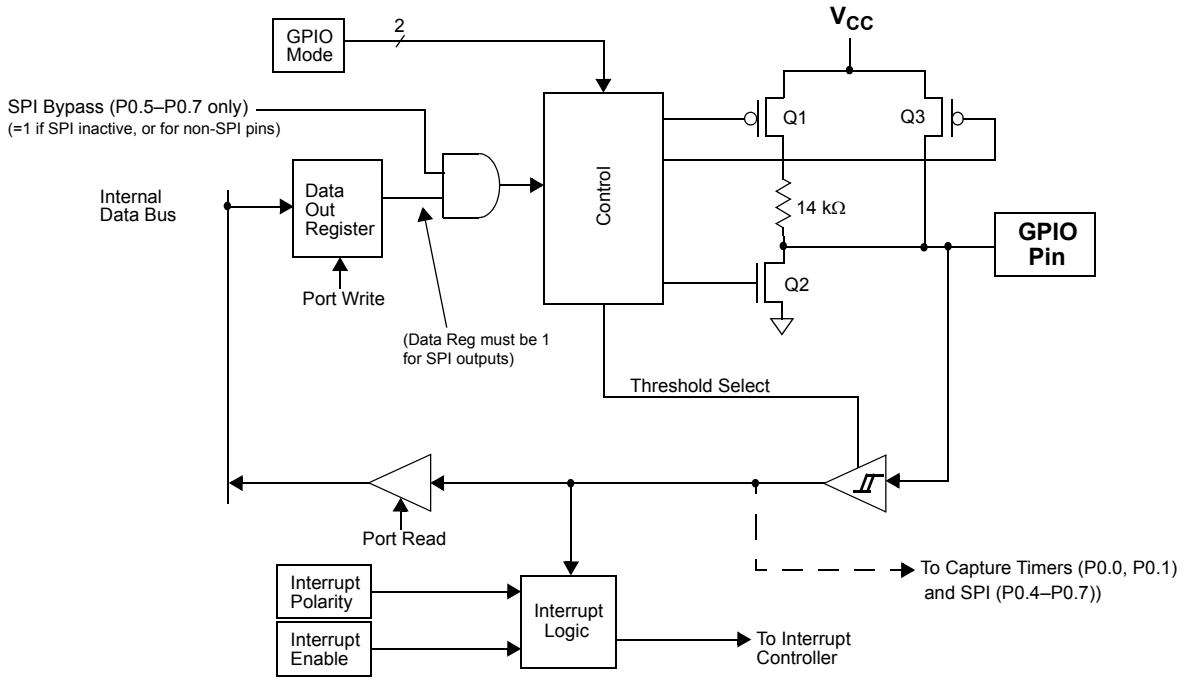
1. Low-voltage Reset (LVR)
2. Brown Out Reset (BOR)
3. Watchdog Reset (WDR)

The occurrence of a reset is recorded in the Processor Status and Control Register (Figure 34 on page 30). Bits 4 (Low-voltage or Brown-out Reset bit) and 6 (Watchdog Reset bit) are used to

General Purpose I/O Ports

Ports 0 and 1 provide up to 16 versatile GPIO pins that can be read or written (the number of pins depends on package type). Figure 6 shows a diagram of a GPIO port pin.

Figure 6. Block Diagram of GPIO Port (one pin shown)



Port 0 is an 8-bit port; Port 1 contains either 2 bits, P1.1–P1.0 in the CY7C63723C, or all 8 bits, P1.7–P1.0 in the CY7C63743C parts. Each bit can also be selected as an interrupt source for the microcontroller, as explained in Interrupt Sources on page 32.

The data for each GPIO pin is accessible through the Port Data register. Writes to the Port Data register store outgoing data state for the port pins, while reads from the Port Data register return the actual logic value on the port pins, not the Port Data register contents.

Each GPIO pin is configured independently. The driving state of each GPIO pin is determined by the value written to the pin's Data Register and by two associated pin's Mode0 and Mode1 bits.

The Port 0 Data Register is shown in Figure 7 on page 17, and the Port 1 Data Register is shown in Figure 8 on page 17. The Mode0 and Mode1 bits for the two GPIO ports are given in Figure 9 on page 17 through Figure 12 on page 18.

Figure 7. Port 0 Data (Address 0x00)

Bit #	7	6	5	4	3	2	1	0
Bit Name	P0							
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

Bit [7:0]: P0[7:0]

1 = Port Pin is logic HIGH

0 = Port Pin is logic LOW

Figure 8. Port 1 Data (Address 0x01)

Bit #	7	6	5	4	3	2	1	0
Bit Name	P1							
Notes	Pins 7:2 only in CY7C63743C						Pins 1:0 in all parts	
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

Bit [7:0]: P1[7:0]

1 = Port Pin is logic HIGH

0 = Port Pin is logic LOW

Figure 9. GPIO Port 0 Mode0 Register (Address 0x0A)

Bit #	7	6	5	4	3	2	1	0
Bit Name	P0[7:0] Mode0							
Read/Write	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit [7:0]: P0[7:0] Mode 0

1 = Port 0 Mode 0 is logic HIGH

0 = Port 0 Mode 0 is logic LOW

Figure 10. GPIO Port 0 Mode1 Register (Address 0x0B)

Bit #	7	6	5	4	3	2	1	0
Bit Name	P0[7:0] Mode1							
Read/Write	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit [7:0]: P0[7:0] Mode 1

1 = Port Pin Mode 1 is logic HIGH

0 = Port Pin Mode 1 is logic LOW

Figure 11. GPIO Port 1 Mode0 Register (Address 0x0C)

Bit #	7	6	5	4	3	2	1	0
Bit Name	P1[7:0] Mode0							
Read/Write	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit [7:0]: P1[7:0] Mode 0

1 = Port Pin Mode 0 is logic HIGH

0 = Port Pin Mode 0 is logic LOW

Figure 12. GPIO Port 1 Mode1 Register (Address 0x0D)

Bit #	7	6	5	4	3	2	1	0
Bit Name	P1[7:0] Mode1							
Read/Write	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit [7:0]: P1[7:0] Mode 1

1 = Port Pin Mode 1 is logic HIGH

0 = Port Pin Mode 1 is logic LOW

Each pin can be independently configured as high-impedance inputs, inputs with internal pull-ups, open drain outputs, or traditional CMOS outputs with selectable drive strengths.

The driving state of each GPIO pin is determined by the value written to the pin's Data Register and by its associated Mode0 and Mode1 bits. *Table 3* lists the configuration states based on these bits. The GPIO ports default on reset to all Data and Mode Registers cleared, so the pins are all in a high-impedance state. The available GPIO output drive strength are:

- **Hi-Z Mode** (Mode1 = 0 and Mode0 = 0)

  Q1, Q2, and Q3 (Figure 6 on page 17) are OFF. The GPIO pin is not driven internally. Performing a read from the Port Data Register return the actual logic value on the port pins.

- **Low Sink Mode** (Mode1 = 1, Mode0 = 0, and the pin's Data Register = 0)

  Q1 and Q3 are OFF. Q2 is ON. The GPIO pin is capable of sinking 2 mA of current.

- **Medium Sink Mode** (Mode1 = 0, Mode0 = 1, and the pin's Data Register = 0)

  Q1 and Q3 are OFF. Q2 is ON. The GPIO pin is capable of sinking 8 mA of current.

- **High Sink Mode** (Mode1 = 1, Mode0 = 1, and the pin's Data Register = 0)

  Q1 and Q3 are OFF. Q2 is ON. The GPIO pin is capable of sinking 50 mA of current.

- **High Drive Mode** (Mode1 = 0 or 1, Mode0 = 1, and the pin's Data Register = 1)

  Q1 and Q2 are OFF. Q3 is ON. The GPIO pin is capable of sourcing 2 mA of current.

- **Resistive Mode** (Mode1 = 1, Mode0 = 0, and the pin's Data Register = 1)

  Q2 and Q3 are OFF. Q1 is ON. The GPIO pin is pulled up with an internal 14-kΩ resistor.

Note that open drain mode can be achieved by fixing the Data and Mode1 Registers LOW, and switching the Mode0 register.

Input thresholds are CMOS, or TTL as shown in the table (See DC Characteristics on page 42 for the input threshold voltage in TTL or CMOS modes). Both input modes include hysteresis to minimize noise sensitivity. In suspend mode, if a pin is used for a wake-up interrupt using an external R-C circuit, CMOS mode is preferred for lowest power.

Table 3. 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

Auxiliary Input Port

Port 2 serves as an auxiliary input port as shown in Figure 13. The Port 2 inputs all have TTL input thresholds.

Figure 14. USB Status and Control Register (Address 0x1F)

Bit #	7	6	5	4	3	2:0		
Bit Name	PS/2 Pull-up Enable	VREG Enable	USB Reset-PS/2 Activity Interrupt Mode	Reserved	USB Bus Activity	D+/D– Forcing Bit		
Read/ Write	R/W	R/W	R/W	-	R/W	R/W		
Reset	0	0	0	0	0	0	0	0

Bit 7: PS/2 Pull-up Enable

This bit is used to enable the internal PS/2 pull-up resistors on the SDATA and SCLK pins. Normally the output high level on these pins is V_{CC}, but note that the output will be clamped to approximately 1 Volt above V_{REG} if the VREG Enable bit is set, or if the Device Address is enabled (bit 7 of the USB Device Address Register, Figure 15 on page 21).

1 = Enable PS/2 Pull-up resistors. The SDATA and SCLK pins are pulled up internally to V_{CC} with two resistors of approximately 5 kΩ (see DC Characteristics on page 42 for the value of R_{PS2}).

0 = Disable PS/2 Pull-up resistors.

Bit 6: V_{REG} Enable

A 3.3 V voltage regulator is integrated on chip to provide a voltage source for a 1.5-kΩ pull-up resistor connected to the D– pin as required by the USB Specification. Note that the VREG output has an internal series resistance of approximately 200Ω, the external pull-up resistor required is approximately 1.3-kΩ (see Figure 19 on page 23).

1 = Enable the 3.3 V output voltage on the VREG pin.

0 = Disable. The VREG pin can be configured as an input.

Bit 5: USB-PS/2 Interrupt Select

This bit allows the user to select whether an USB bus reset interrupt or a PS/2 activity interrupt will be generated when the interrupt conditions are detected.

1 = PS/2 interrupt mode. A PS/2 activity interrupt will occur if the SDATA pin is continuously LOW for 128 to 256 µs.

0 = USB interrupt mode (default state). In this mode, a USB bus reset interrupt will occur if the single ended zero (SE0, D– and D+ are LOW) exists for 128 to 256 µs.

See Interrupt Sources on page 32 for more details.

Bit 4: Reserved. Must be written as a ‘0’.

Bit 3: USB Bus Activity

The Bus Activity bit is a “sticky” bit that detects any non-idle USB event has occurred on the USB bus. Once set to HIGH by the SIE to indicate the bus activity, this bit retains its logical HIGH value until firmware clears it. Writing a ‘0’ to this bit clears it; writing a ‘1’ preserves its value. The user firmware should check and clear this bit periodically to detect any loss of bus activity. Firmware can clear the Bus Activity bit, but only the SIE can set it. The 1.024-ms timer interrupt service routine is normally used to check and clear the Bus Activity bit.

1 = There has been bus activity since the last time this bit was cleared. This bit is set by the SIE.

0 = No bus activity since last time this bit was cleared (by firmware).

Bit [2:0]: D+/D– Forcing Bit [2:0]

Forcing bits allow firmware to directly drive the D+ and D– pins, as shown in Table 4. Outputs are driven with controlled edge rates in these modes for low EMI. For forcing the D+ and D– pins in USB mode, D+/D– Forcing Bit 2 should be 0. Setting D+/D– Forcing Bit 2 to ‘1’ puts both pins in an open-drain mode, preferred for applications such as PS/2 or LED driving.

Table 4. Control Modes to Force D+/D– Outputs

D+/D– Forcing Bit [2:0]	Control Action	Application
000	Not forcing (SIE controls driver)	Any Mode
001	Force K (D+ HIGH, D– LOW)	USB Mode
010	Force J (D+ LOW, D– HIGH)	
011	Force SE0 (D– LOW, D+ LOW)	
100	Force D– LOW, D+ LOW	PS/2 Mode[3]
101	Force D– LOW, D+ HiZ	
110	Force D– HiZ, D+ LOW	
111	Force D– HiZ, D+ HiZ	

USB Device

The CY7C637xxC supports one USB Device Address with three endpoints: EP0, EP1, and EP2.

USB Address Register

The USB Device Address Register contains a 7-bit USB address and one bit to enable USB communication. This register is cleared during a reset, setting the USB device address to zero and marking this address as disabled. Figure 15 shows the format of the USB Address Register.

Note
3. For PS/2 operation, the D+/D– Forcing Bit [2:0] = 111b mode must be set initially (one time only) before using the other PS/2 force modes

Figure 15. USB Device Address Register (Address 0x10)

Bit #	7	6	5	4	3	2	1	0
Bit Name	Device Address Enable	Device Address						
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

In either USB or PS/2 mode, this register is cleared by both hardware resets and the USB bus reset. See Interrupt Sources on page 32 for more information on the USB Bus Reset – PS/2 interrupt.

Bit 7: Device Address Enable

This bit must be enabled by firmware before the serial interface engine (SIE) will respond to USB traffic at the address specified in Bit [6:0].

1 = Enable USB device address.

0 = Disable USB device address.

Bit [6:0]: Device Address Bit [6:0]

These bits must be set by firmware during the USB enumeration process (i.e., SetAddress) to the non-zero address assigned by the USB host.

USB Control Endpoint

All USB devices are required to have an endpoint number 0 (EP0) that is used to initialize and control the USB device. EP0 provides access to the device configuration information and allows generic USB status and control accesses. EP0 is bidirectional as the device can both receive and transmit data. EP0 uses an 8-byte FIFO at SRAM locations 0xF8-0xFF, as shown in Data Memory Organization on page 10.

The EP0 endpoint mode register uses the format shown in Figure .

Figure 16. Endpoint 0 Mode Register (Address 0x12)

Bit #	7	6	5	4	3:0
Bit Name	SETUP Received	IN Received	OUT Received	ACKed Transaction	Mode Bit
Read/ Write	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0 0 0 0

The SIE provides a locking feature to prevent firmware from overwriting bits in the USB Endpoint 0 Mode Register. Writes to the register have no effect from the point that Bit[6:0] of the register are updated (by the SIE) until the firmware reads this register. The CPU can unlock this register by reading it.

Because of these hardware-locking features, firmware should perform an read after a write to the USB Endpoint 0 Mode Register and USB Endpoint 0 Count Register (Figure 18) to verify that the contents have changed as desired, and that the SIE has not updated these values.

Bit [7:4] of this register are cleared by any non-locked write to this register, regardless of the value written.

Bit 7: SETUP Received

1 = A valid SETUP packet has been received. This bit is forced HIGH from the start of the data packet phase of the SETUP transaction until the start of the ACK packet returned by the SIE. The CPU is prevented from clearing this bit during this interval. While this bit is set to ‘1’, the CPU cannot write to the EP0 FIFO. This prevents firmware from overwriting an incoming SETUP transaction before firmware has a chance to read the SETUP data.

0 = No SETUP received. This bit is cleared by any non-locked writes to the register.

Bit 6: IN Received

1 = A valid IN packet has been received. This bit is updated to ‘1’ after the last received packet in an IN transaction. This bit is cleared by any non-locked writes to the register.

0 = No IN received. This bit is cleared by any non-locked writes to the register.

Bit 5: OUT Received

1 = A valid OUT packet has been received. This bit is updated to ‘1’ after the last received packet in an OUT transaction. This bit is cleared by any non-locked writes to the register.

0 = No OUT received. This bit is cleared by any non-locked writes to the register.

Bit 4: ACKed Transaction

The ACKed Transaction bit is set whenever the SIE engages in a transaction to the register's endpoint that completes with an ACK packet.

1 = The transaction completes with an ACK.

0 = The transaction does not complete with an ACK.

Bit [3:0]: Mode Bit[3:0]

The endpoint modes determine how the SIE responds to USB traffic that the host sends to the endpoint. For example, if the endpoint Mode Bits [3:0] are set to 0001 which is NAK IN/OUT mode as shown in *Table 8*, the SIE will send NAK handshakes in response to any IN or OUT token sent to this endpoint. In this NAK IN/OUT mode, the SIE will send an ACK handshake when the host sends a SETUP token to this endpoint. The mode encoding is shown in *Table 8*. Additional information on the mode bits can be found in *Table 9* and *Table 10*. These modes give the firmware total control on how to respond to different tokens sent to the endpoints from the host.

In addition, the Mode Bits are automatically changed by the SIE in response to many USB transactions. For example, if the Mode Bit [3:0] are set to 1011 which is ACK OUT-NAK IN mode as shown in *Table 8*, the SIE will change the endpoint Mode Bit [3:0] to NAK IN/OUT (0001) mode after issuing an ACK handshake in response to an OUT token. Firmware needs to update the mode for the SIE to respond appropriately.

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.

Figure 24. Timer LSB Register (Address 0x24)

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

Bit [7:0]: Timer lower eight bits

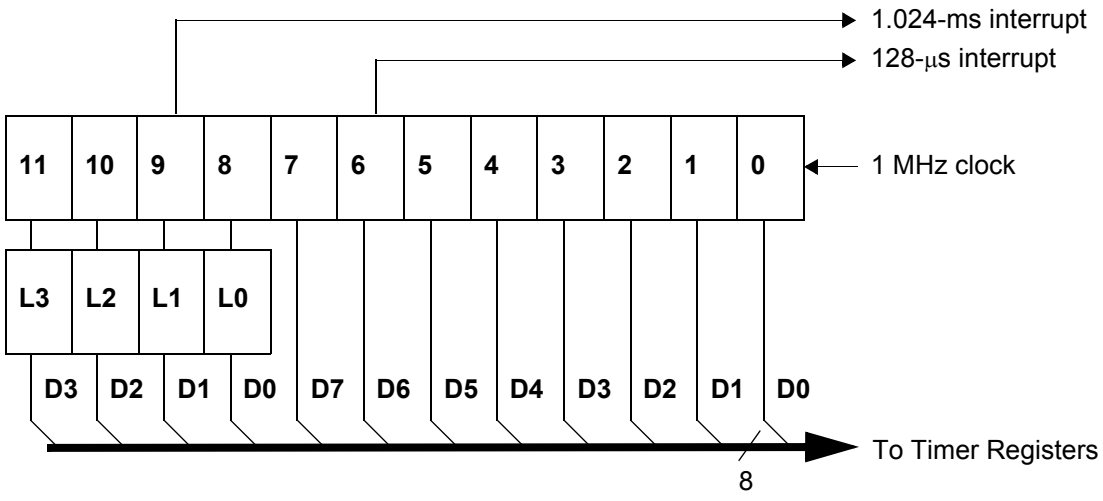
Figure 25. Timer MSB Register (Address 0x25)

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

Bit [7:4]: Reserved

Bit [3:0]: Timer upper four bits

Figure 26. Timer Block Diagram



1 = Suspend the processor.

0 = Not in suspend mode. Cleared by the hardware when resuming from suspend.

Bit 2: Interrupt Enable Sense

This bit shows whether interrupts are enabled or disabled. Firmware has no direct control over this bit as writing a zero or one to this bit position will have no effect on interrupts. This bit is further gated with the bit settings of the Global Interrupt Enable Register (Figure 35) and USB Endpoint Interrupt Enable Register (Figure 36). Instructions DI, EI, and RETI manipulate the state of this bit.

1 = Interrupts are enabled.

0 = Interrupts are masked off.

Bit 1: Reserved. Must be written as a 0.

Bit 0: Run

This bit is manipulated by the HALT instruction. When Halt is executed, the processor clears the run bit and halts at the end of the current instruction. The processor remains halted until a reset occurs (low-voltage, brown-out, or Watchdog). This bit should normally be written as a '1'.

During power-up, or during a low-voltage reset, the Processor Status and Control Register is set to 00010001, which indicates a LVR/BOR (bit 4 set) has occurred and no interrupts are pending (bit 7 clear). Note that during the t_{START} ms partial suspend at start-up (explained in Reset on page 13), a Watchdog Reset will also occur. When a WDR occurs during the power-up suspend interval, firmware would read 01010001 from the Status and Control Register after power-up. Normally the LVR/BOR bit should be cleared so that a subsequent WDR can be clearly identified. *Note that if a USB bus reset (long SE0) is received before firmware examines this register, the Bus Interrupt Event bit would also be set.*

During a Watchdog Reset, the Processor Status and Control Register is set to 01XX0001, which indicates a Watchdog Reset (bit 4 set) has occurred and no interrupts are pending (bit 7 clear).

Interrupts

Interrupts can be generated by the GPIO lines, the internal free-running timer, the SPI block, the capture timers, on various USB events, PS/2 activity, or by the wake-up timer. All interrupts are maskable by the Global Interrupt Enable Register and the USB End Point Interrupt Enable Register. Writing a '1' to a bit position enables the interrupt associated with that bit position. During a reset, the contents of the interrupt enable registers are cleared, along with the Global Interrupt enable bit of the CPU, effectively disabling all interrupts.

The interrupt controller contains a separate flip-flop for each interrupt. See Figure 37 for the logic block diagram of the interrupt controller. When an interrupt is generated it is first registered as a pending interrupt. It will stay pending until it is serviced or a reset occurs. A pending interrupt will only generate an interrupt request if it is enabled by the corresponding bit in the interrupt enable registers. The highest priority interrupt request will be serviced following the completion of the currently executing instruction.

When servicing an interrupt, the hardware will first disable all interrupts by clearing the Global Interrupt Enable bit in the CPU (the state of this bit can be read at Bit 2 of the Processor Status and Control Register). Next, the flip-flop of the current interrupt is cleared. This is followed by an automatic CALL instruction to the ROM address associated with the interrupt being serviced (i.e., the Interrupt Vector, see Interrupt Vectors on page 31). 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 executing an EI instruction. Interrupts can be nested to a level limited only by the available stack space.

The Program Counter value as well as the Carry and Zero flags (CF, ZF) are stored onto the Program Stack by the automatic CALL instruction generated 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. The PUSH A instruction should typically be used as the first command in the ISR to save the accumulator value and the POP A instruction should be used just before the RETI instruction to restore the accumulator value. The program counter, CF and ZF are restored and interrupts are enabled when the RETI instruction is executed.

The DI and EI instructions can be used to disable and enable interrupts, respectively. These instructions affect only the Global Interrupt Enable bit of the CPU. If desired, EI can be used to re-enable interrupts while inside an ISR, instead of waiting for the RETI that exits the ISR. While the global interrupt enable bit is cleared, the presence of a pending interrupt can be detected by examining the IRQ Sense bit (Bit 7 in the Processor Status and Control Register).

Interrupt Vectors

The Interrupt Vectors supported by the device are listed in Table 7. The highest priority interrupt is #1 (USB Bus Reset / PS/2 activity), and the lowest priority interrupt is #11 (Wake-up Timer). Although Reset is not an interrupt, the first instruction executed after a reset is at ROM address 0x0000, which corresponds to the first entry in the Interrupt Vector Table. Interrupt vectors occupy two bytes to allow for a two-byte JMP instruction to the appropriate Interrupt Service Routine (ISR).

Table 7. Interrupt Vector Assignments

Interrupt Vector No.	ROM Address	Function
not applicable	0x0000	Execution after Reset begins here
1	0x0002	USB Bus Reset or PS/2 Activity interrupt
2	0x0004	128-µs timer interrupt
3	0x0006	1.024-ms timer interrupt
4	0x0008	USB Endpoint 0 interrupt
5	0x000A	USB Endpoint 1 interrupt
6	0x000C	USB Endpoint 2 interrupt
7	0x000E	SPI Interrupt
8	0x0010	Capture Timer A interrupt

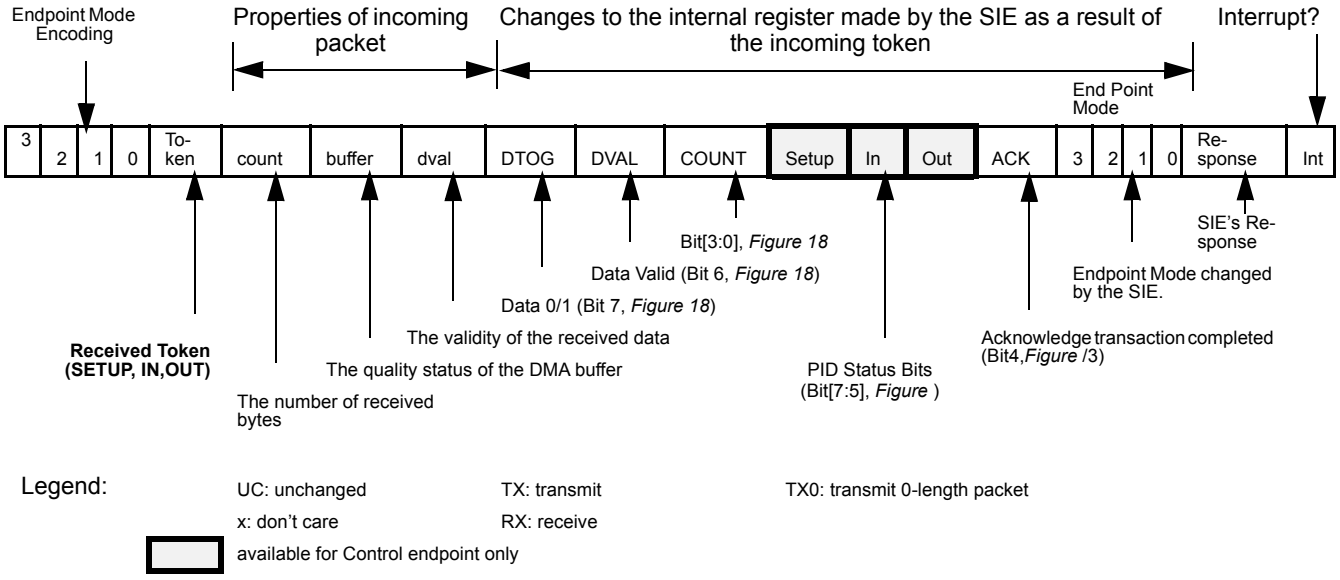
Bits [3:0] are set to '1111' which is ACK IN-Status OUT mode as shown in Table 8, the SIE will change the endpoint Mode Bits [3:0] to NAK IN-Status OUT mode (1110) after ACKing a valid status stage OUT token. The firmware needs to update the mode for the SIE to respond appropriately. See Table 8 for 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 9. Decode table for Table 10: "Details of Modes for Differing Traffic Conditions"



Register Summary

	Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Read/Write/ Both/	Default/ Reset
GPIO CONFIGURATION PORTS 0, 1 AND 2	0x00	Port 0 Data	P0								BBBBBBBB	00000000
	0x01	Port 1 Data	P1								BBBBBBBB	00000000
	0x02	Port 2 Data	Reserved		D+(SCLK) State	D- (SDATA) State	Reserved		P2.1 (Int Clk Mode Only	VREG Pin State	--RR--RR	00000000
	0x0A	GPIO Port 0 Mode 0	P0[7:0] Mode0								WWWWWWWW	00000000
	0x0B	GPIO Port 0 Mode 1	P0[7:0] Mode1								WWWWWWWW	00000000
	0x0C	GPIO Port 1 Mode 0	P1[7:0] Mode0								WWWWWWWW	00000000
	0x0D	GPIO Port 1 Mode 1	P1[7:0] Mode1								WWWWWWWW	00000000
	0x04	Port 0 Interrupt Enable	P0[7:0] Interrupt Enable								WWWWWWWW	00000000
	0x05	Port 1 Interrupt Enable	P1[7:0] Interrupt Enable								WWWWWWWW	00000000
	0x06	Port 0 Interrupt Polarity	P0[7:0] Interrupt Polarity								WWWWWWWW	00000000
	0x07	Port 1 Interrupt Polarity	P1[7:0] Interrupt Polarity								WWWWWWWW	00000000
Clock Config.	0xF8	Clock Configuration	Ext. Clock Resume Delay	Wake-up Timer Adjust Bit [2:0]			Low-voltage Reset Disable	Precision USB Clocking Enable	Internal Clock Output Disable	External Oscillator Enable	BBBBBBBB	00000000
ENDPOINT 0, I AND 2 CONFIGURATION	0x10	USB Device Address	Device Address Enable	Device Address							BBBBBBBB	00000000
	0x12	EP0 Mode	SETUP Received	IN Received	OUT Received	ACKed Transaction	Mode Bit				BBBBBBBB	00000000
	0x14, 0x16	EP1, EP2 Mode Register	STALL	Reserved		ACKed Transaction	Mode Bit				B--BBBBB	00000000
	0x11, 0x13, and 0x15	EP0,1, and 2 Counter	Data 0/1 Toggle	Data Valid	Reserved		Byte Count				BB--BBBB	00000000
USB-SC	0x1F	USB Status and Control	PS/2 Pull-up Enable	VREG Enable	USB Reset-PS/2 Activity Interrupt Mode	Reserved	USB Bus Activity	D+/D- Forcing Bit			BBB-BBBB	00000000
INTERRUPT	0x20	Global Interrupt Enable	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	BBBBBBBB	00000000
	0x21	Endpoint Interrupt Enable	Reserved					EP2 Interrupt Enable	EP1 Interrupt Enable	EP0 Interrupt Enable	-----BBB	00000000
TIMER	0x24	Timer LSB	Timer Bit [7:0]								RRRRRRRR	00000000
	0x25	Timer (MSB)	Reserved				Timer Bit [11:8]				----RRRR	00000000
SPI	0x60	SPI Data	Data I/O								BBBBBBBB	00000000
	0x61	SPI Control	TCMP	TBF	Comm Mode [1:0]		CPOL	CPHA	SCK Select		BBBBBBBB	00000000
CAPTURE TIMER	0x40	Capture Timer A-Rising, Data Register	Capture A Rising Data								RRRRRRRR	00000000
	0x41	Capture Timer A-Falling, Data Register	Capture A Falling Data								RRRRRRRR	00000000
	0x42	Capture Timer B-Rising, Data Register	Capture B Rising Data								RRRRRRRR	00000000
	0x43	Capture Timer B-Falling, Data Register	Capture B Falling Data								RRRRRRRR	00000000
	0x44	Capture Timer Configuration	First Edge Hold	Prescale Bit [2:0]			Capture B Falling Intr Enable	Capture B Rising Intr Enable	Capture A Falling Intr Enable	Capture A Rising Intr Enable	BBBBBBBB	00000000
	0x45	Capture Timer Status	Reserved				Capture B Falling Event	Capture B Rising Event	Capture A Falling Event	Capture A Rising Event	----BBBB	00000000
PROC SC.	0xFF	Process Status & Control	IRQ Pending	Watch Dog Reset	Bus Interrupt Event	LVR/BOR Reset	Suspend	Interrupt Enable Sense	Reserved	Run	RBBBBR-B	See Processor Status and Control Register

Figure 44. Receiver Jitter Tolerance

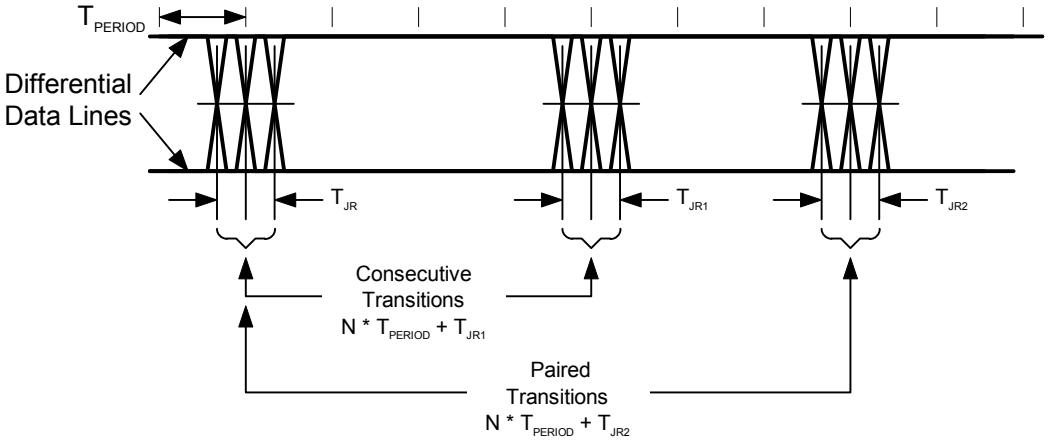


Figure 45. Differential to EOP Transition Skew and EOP Width

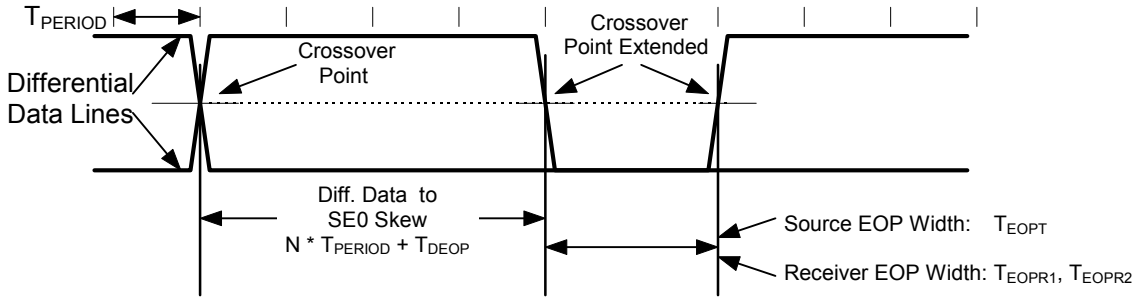
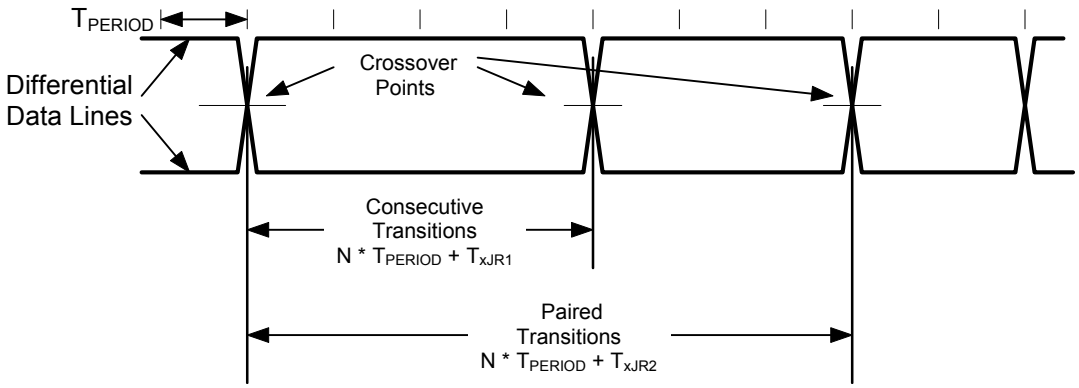


Figure 46. Differential Data Jitter



Errata

This section describes the errata for the enCoRe™ USB Combination Low-speed USB & PS/2 Periheral Controller / CY7C637xx. The details include errata trigger conditions, available workaround, and silicon revision applicability.

Please contact your local Cypress Sales Representative if you have further questions.

Part Numbers Affected

Part Number	Device Characteristics
CY7C63722	All packages
CY7C63723	All packages
CY7C63743	All packages

enCoRe™ USB Combination Low-speed USB & PS/2 Peripheral Controller Qualification Status

Product status: In Production - Qual report: 001406

enCoRe™ USB Combination Low-speed USB & PS/2 Peripheral Controller Errata Summary

The following table defines the errata applicability to available enCoRe™ USB Combination Low-speed USB & PS/2 Peripheral Controller family devices. An "X" indicates that the errata pertains to the selected device.

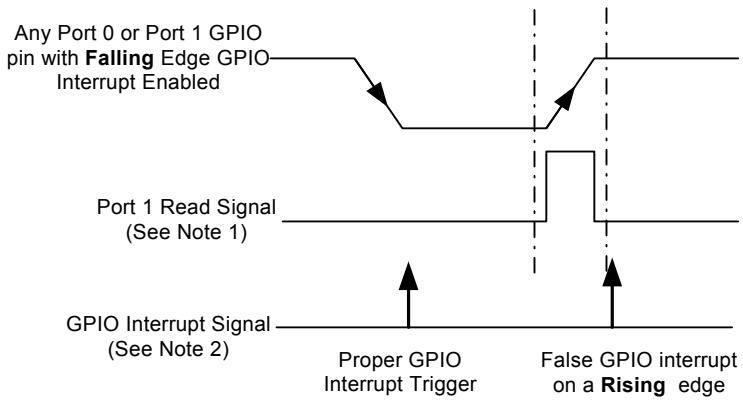
Note: Errata titles are hyperlinked. Click on table entry to jump to description.

Items	CY7C637xx	Rev Letter	Fix Status
1. Faulty GPIO Interrupt	X	A	No silicon fix planned.

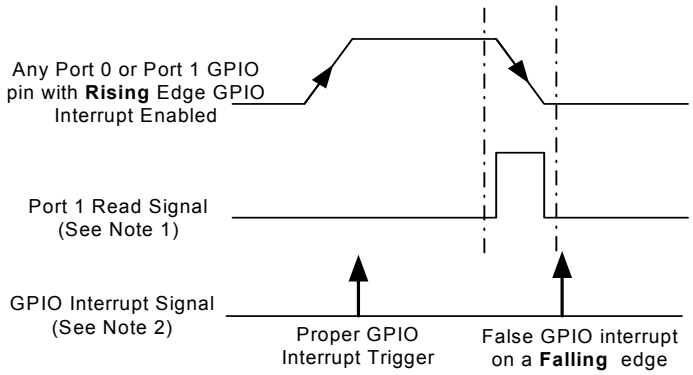
1. Faulty GPIO Interrupt

■ **Problem Definition**

When a falling edge interrupt is enabled for a GPIO pin, reading the GPIO Port 1 coincident to a rising edge of that GPIO signal may generate a false GPIO interrupt.



When a rising edge interrupt is enabled for a GPIO pin, reading the GPIO Port 1 coincident to a falling edge of that GPIO signal may generated a false GPIO intterupt.



Note 1: Port 1 Read is an internal signal that is asserted when Port 1 is read with an "IORD 01h" instruction.

Note 2: The GPIO Interrupt signal is an internal signal. The arrow indicates that a GPIO interrupt is triggered.

■ **Parameters Affected**

Interrupts

■ **Trigger Condition(S)**

Reading the GPIO Port 1 when either rising or falling edge interrupts are enabled for a GPIO pin.

■ **Scope of Impact**

The chip enters the GPIO Interrupt Service Routine (ISR) in error.

■ **Workaround**

Workarounds will need to be tailored to individual applications based on the flexibility of changing the GPIO usage, the timing of the GPIO interrupt sources and firmware interrupt latencies.

■ **Fix Status**

No silicon fix is planned.