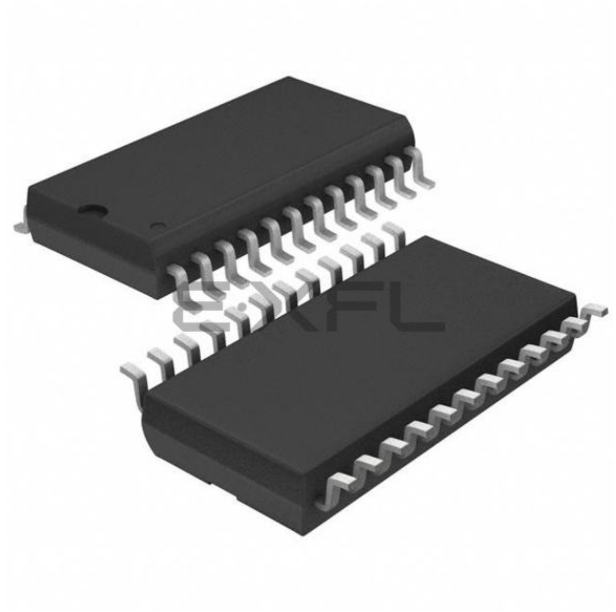
Infineon Technologies - CY7C63743-SC Datasheet



Welcome to E-XFL.COM

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

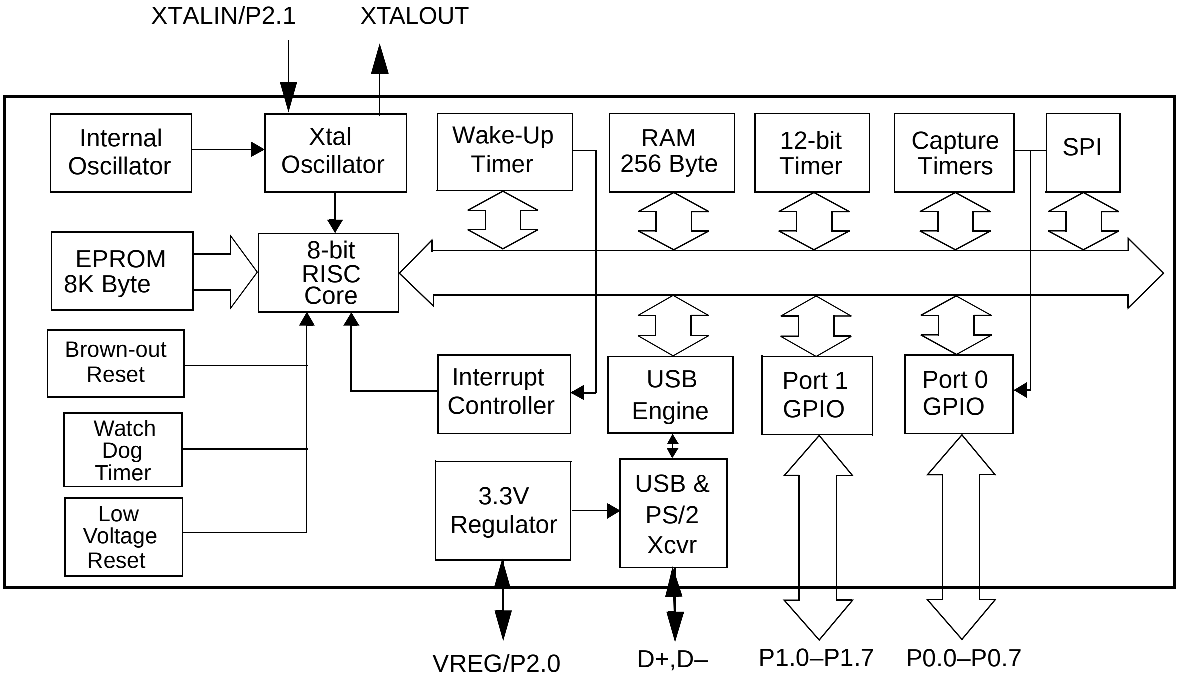
Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering 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	16
Voltage - Supply	4V ~ 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/cy7c63743-sc

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

Table 8-1. I/O Register Summary (continued)

Register Name	I/O Address	Read/Write	Function	Fig.
USB Device Address	0x10	R/W	USB Device Address register	14-1
EP0 Counter Register	0x11	R/W	USB Endpoint 0 counter register	14-4
EP0 Mode Register	0x12	R/W	USB Endpoint 0 configuration register	14-2
EP1 Counter Register	0x13	R/W	USB Endpoint 1 counter register	14-4
EP1 Mode Register	0x14	R/W	USB Endpoint 1 configuration register	14-3
EP2 Counter Register	0x15	R/W	USB Endpoint 2 counter register	14-4
EP2 Mode Register	0x16	R/W	USB Endpoint 2 configuration register	14-3
USB Status & Control	0x1F	R/W	USB status and control register	13-1
Global Interrupt Enable	0x20	R/W	Global interrupt enable register	21-1
Endpoint Interrupt Enable	0x21	R/W	USB endpoint interrupt enables	21-2
Timer (LSB)	0x24	R	Lower 8 bits of free-running timer (1 MHz)	18-1
Timer (MSB)	0x25	R	Upper 4 bits of free-running timer	18-2
WDR Clear	0x26	W	Watchdog Reset clear	-
Capture Timer A Rising	0x40	R	Rising edge Capture Timer A data register	19-2
Capture Timer A Falling	0x41	R	Falling edge Capture Timer A data register	19-3
Capture Timer B Rising	0x42	R	Rising edge Capture Timer B data register	19-4
Capture Timer B Falling	0x43	R	Falling edge Capture Timer B data register	19-5
Capture TImer Configuration	0x44	R/W	Capture Timer configuration register	19-7
Capture Timer Status	0x45	R	Capture Timer status register	19-6
SPI Data	0x60	R/W	SPI read and write data register	17-2
SPI Control	0x61	R/W	SPI status and control register	17-3
Clock Configuration	0xF8	R/W	Internal / External Clock configuration register	9-2
Processor Status & Control	0xFF	R/W	Processor status and control	20-1

9.0 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 9-1*. No additional capacitance is included on chip at the XTALIN/OUT pins. Operation is controlled by the Clock Configuration Register, *Figure 9-2*.

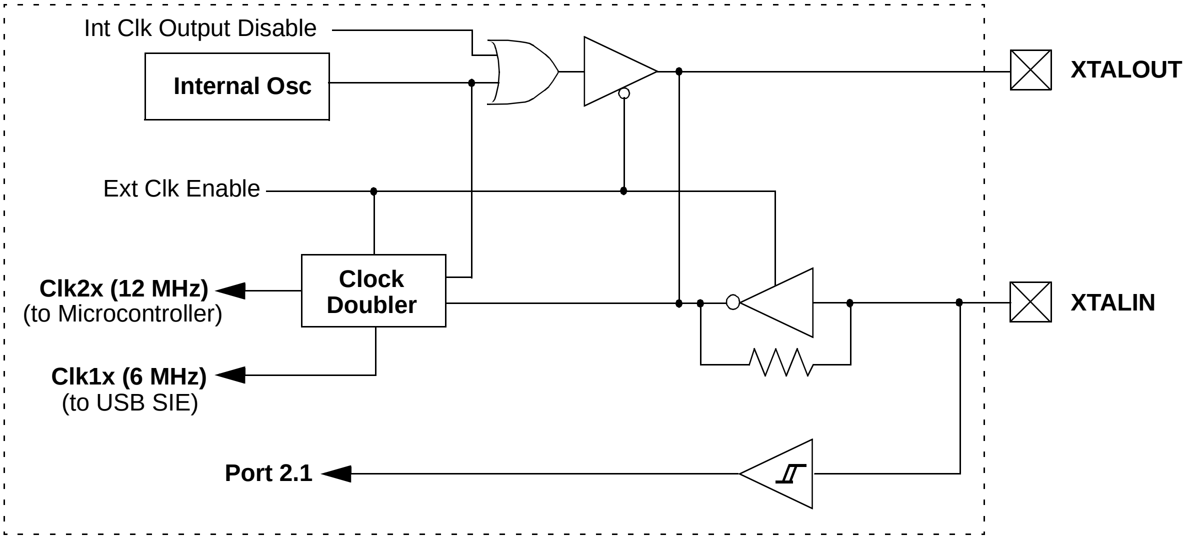


Figure 9-1. Clock Oscillator On-chip Circuit

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

Figure 9-2. Clock Configuration Register (Address 0xF8)

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 Section 11.2. 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 Section 25.0 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 Section 26.0 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

The microcontroller begins execution from ROM address 0x0000 after a LVR, BOR, or WDR reset. Although this looks like interrupt vector 0, there is an important difference. Reset processing does NOT push the program counter, carry flag, and zero flag onto program stack. Attempting to execute either a RET or RETI in the reset handler will cause unpredictable execution results.

The following events take place on reset. More details on the various resets are given in the following sections.

1. All registers are reset to their default states (all bits cleared, except in Processor Status and Control Register).
2. GPIO and USB pins are set to high-impedance state.
3. The VREG pin is set to high-impedance state.
4. Interrupts are disabled.
5. USB operation is disabled and must be enabled by firmware if desired, as explained in Section 14.1.
6. For a BOR or LVR, the external oscillator is disabled and Internal Clock mode is activated, followed by a time-out period t_{START} for V_{CC} to stabilize. A WDR does not change the clock mode, and there is no delay for V_{CC} stabilization on a WDR. Note that the External Oscillator Enable (Bit 0, *Figure 9-2*) will be cleared by a WDR, but it does not take effect until suspend mode is entered.
7. The Program Stack Pointer (PSP) and Data Stack Pointer (DSP) reset to address 0x00. Firmware should move the DSP for USB applications, as explained in Section 6.5.
8. Program execution begins at address 0x0000 after the appropriate time-out period.

10.1 Low-voltage Reset (LVR)

When V_{CC} is first applied to the chip, the internal oscillator is started and the Low-voltage Reset is initially enabled by default. At the point where V_{CC} has risen above V_{LVR} (see Section 25.0 for the value of V_{LVR}), an internal counter starts counting for a period of t_{START} (see Section 26.0 for the value of t_{START}). During this t_{START} time, the microcontroller enters a partial suspend state to wait for V_{CC} to stabilize before it begins executing code from address 0x0000.

As long as the LVR circuit is enabled, this reset sequence repeats whenever the V_{CC} pin voltage drops below V_{LVR}. The LVR can be disabled by firmware by setting the Low-voltage Reset Disable bit in the Clock Configuration Register (*Figure 9-2*). In addition, the LVR is automatically disabled in suspend mode to save power. If the LVR was enabled before entering suspend mode, it becomes active again once the suspend mode ends.

When LVR is disabled during normal operation (i.e., by writing ‘0’ to the Low-voltage Reset Disable bit in the Clock Configuration Register), the chip may enter an unknown state if V_{CC} drops below V_{LVR}. Therefore, LVR should be enabled at all times during normal operation. If LVR is disabled (i.e., by firmware or during suspend mode), a secondary low-voltage monitor, BOR, becomes active, as described in the next section. The LVR/BOR Reset bit of the Processor Status and Control Register (*Figure 20-1*), is set to ‘1’ if either a LVR or BOR has occurred.

10.2 Brown Out Reset (BOR)

The Brown Out Reset (BOR) circuit is always active and behaves like the POR. BOR is asserted whenever the V_{CC} voltage to the device is below an internally defined trip voltage of approximately 2.5V. The BOR re-enables LVR. That is, once V_{CC} drops and trips BOR, the part remains in reset until V_{CC} rises above V_{LVR}. At that point, the t_{START} delay occurs before normal operation resumes, and the microcontroller starts executing code from address 0x00 after the t_{START} delay.

In suspend mode, only the BOR detection is active, giving a reset if V_{CC} drops below approximately 2.5V. Since the device is suspended and code is not executing, this lower reset voltage is safe for retaining the state of all registers and memory. Note that in suspend mode, LVR is disabled as discussed in Section 10.1.

10.3 Watchdog Reset (WDR)

The Watchdog Timer Reset (WDR) occurs when the internal Watchdog timer rolls over. Writing any value to the write-only Watchdog Reset Register at address 0x26 will clear the timer. The timer will roll over and WDR will occur if it is not cleared within t_{WATCH} (see *Figure 10-1*) of the last clear. Bit 6 (Watchdog Reset bit) of the Processor Status and Control Register is set to record this event (see Section 20.0 for more details). A Watchdog Timer Reset typically lasts for 2–4 ms, after which the microcontroller begins execution at ROM address 0x0000.

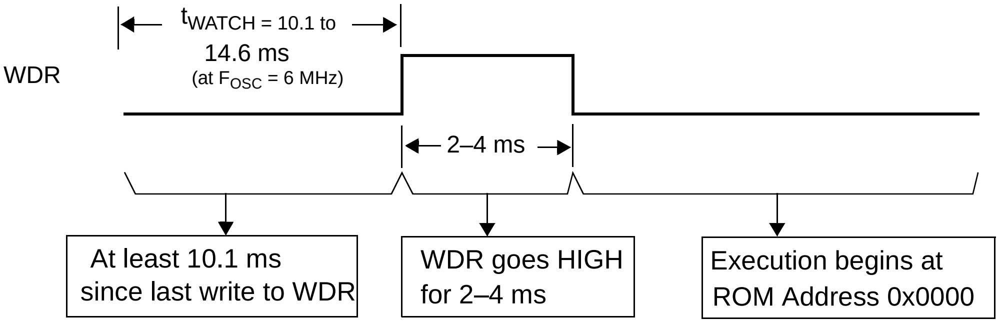


Figure 10-1. Watchdog Reset (WDR, Address 0x26)

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

Figure 12-2. Port 0 Data (Address 0x00)

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

1 = Port Pin is logic HIGH

0 = Port Pin is logic LOW

Bit #	7	6	5	4	3	2	1	0
Bit Name	P1							
Notes	Pins 7:2 only in CY7C63743						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

Figure 12-3. Port 1 Data (Address 0x01)

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

1 = Port Pin is logic HIGH

0 = Port Pin is logic LOW

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

Figure 12-4. GPIO Port 0 Mode0 Register (Address 0x0A)

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

1 = Port 0 Mode 0 is logic HIGH

0 = Port 0 Mode 0 is logic LOW

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

Figure 12-5. GPIO Port 0 Mode1 Register (Address 0x0B)

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

1 = Port Pin Mode 1 is logic HIGH

0 = Port Pin Mode 1 is logic LOW

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

Figure 12-6. GPIO Port 1 Mode0 Register (Address 0x0C)

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

1 = Port Pin Mode 0 is logic HIGH

0 = Port Pin Mode 0 is logic LOW

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

Figure 12-7. GPIO Port 1 Mode1 Register (Address 0x0D)

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 12-1* 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 12-1*) 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 Section 25.0 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.

15.0 USB Regulator Output

The VREG pin provides a regulated output for connecting the pull-up resistor required for USB operation. For USB, a 1.5-kΩ resistor is connected between the D– pin and the V_{REG} voltage, to indicate low-speed USB operation. Since the VREG output has an internal series resistance of approximately 200Ω, the external pull-up resistor required is R_{PU} (see Section 25.0).

The regulator output is placed in a high-impedance state at reset, and must be enabled by firmware by setting the VREG Enable bit in the USB Status and Control Register (*Figure 13-1*). This simplifies the design of a combination PS/2-USB device, since the USB pull-up resistor can be left in place during PS/2 operation without loading the PS/2 line. In this mode, the V_{REG} pin can be used as an input and its state can be read at port P2.0. Refer to *Figure 12-8* for the Port 2 data register. This input has a TTL threshold.

In suspend mode, the regulator is automatically disabled. If VREG Enable bit is set (*Figure 13-1*), the VREG pin is pulled up to V_{CC} with an internal 6.2-kΩ resistor. This holds the proper V_{OH} state in suspend mode

Note that enabling the device for USB (by setting the Device Address Enable bit, *Figure 14-1*) activates the internal regulator, even if the VREG Enable bit is cleared to 0. This insures proper USB signaling in the case where the VREG pin is used as an input, and an external regulator is provided for the USB pull-up resistor. This also limits the swing on the D– and D+ pins to about 1V above the internal regulator voltage, so the Device Address Enable bit normally should only be set for USB operating modes.

The regulator output is only designed to provide current for the USB pull-up resistor. In addition, the output voltage at the VREG pin is effectively disconnected when the CY7C637xx device transmits USB from the internal SIE. This means that the VREG pin does not provide a stable voltage during transmits, although this does not affect USB signaling.

16.0 PS/2 Operation

The CY7C637xx parts are optimized for combination USB or PS/2 devices, through the following features:

1. USB D+ and D– lines can also be used for PS/2 SCLK and SDATA pins, respectively. With USB disabled, these lines can be placed in a high-impedance state that will pull up to V_{CC}. (Disable USB by clearing the Address Enable bit of the USB Device Address Register, *Figure 14-1*).
2. An interrupt is provided to indicate a long LOW state on the SDATA pin. This eliminates the need to poll this pin to check for PS/2 activity. Refer to Section 21.3 for more details.
3. Internal PS/2 pull-up resistors can be enabled on the SCLK and SDATA lines, so no GPIO pins are required for this task (bit 7, USB Status and Control Register, *Figure 13-1*).
4. The controlled slew rate outputs from these pins apply to both USB and PS/2 modes to minimize EMI.
5. The state of the SCLK and SDATA pins can be read, and can be individually driven LOW in an open drain mode. The pins are read at bits [5:4] of Port 2, and are driven with the Control Bits [2:0] of the USB Status and Control Register.
6. The V_{REG} pin can be placed into a high-impedance state, so that a USB pull-up resistor on the D–/SDATA pin will not interfere with PS/2 operation (bit 6, USB Status and Control Register).

The PS/2 on-chip support circuitry is illustrated in *Figure 16-1*.

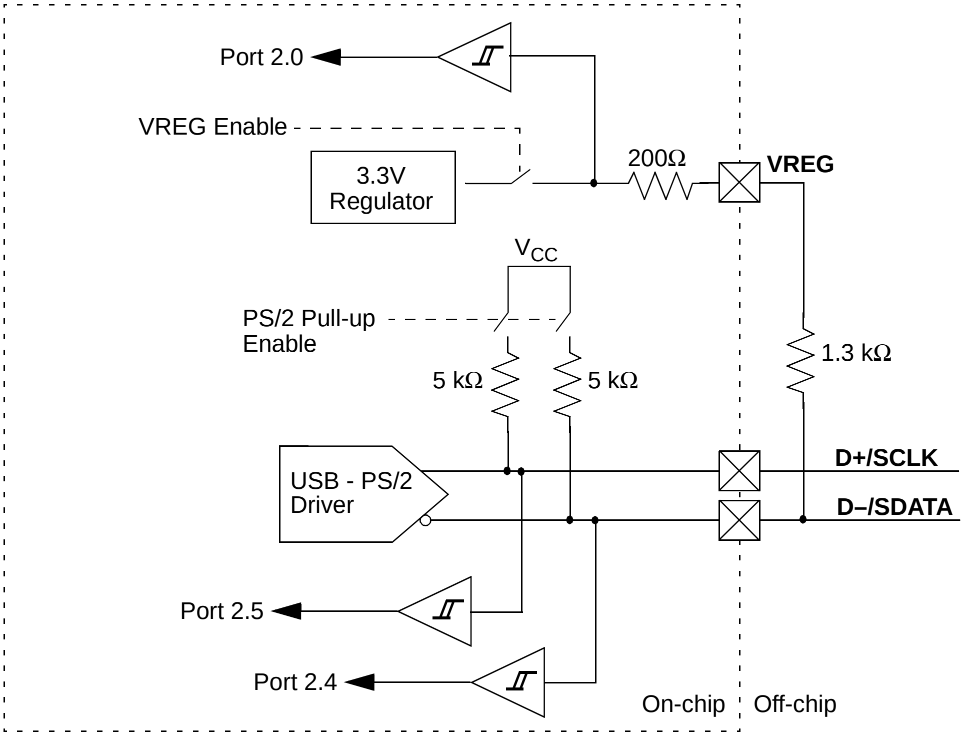


Figure 16-1. Diagram of USB-PS/2 System Connections

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.

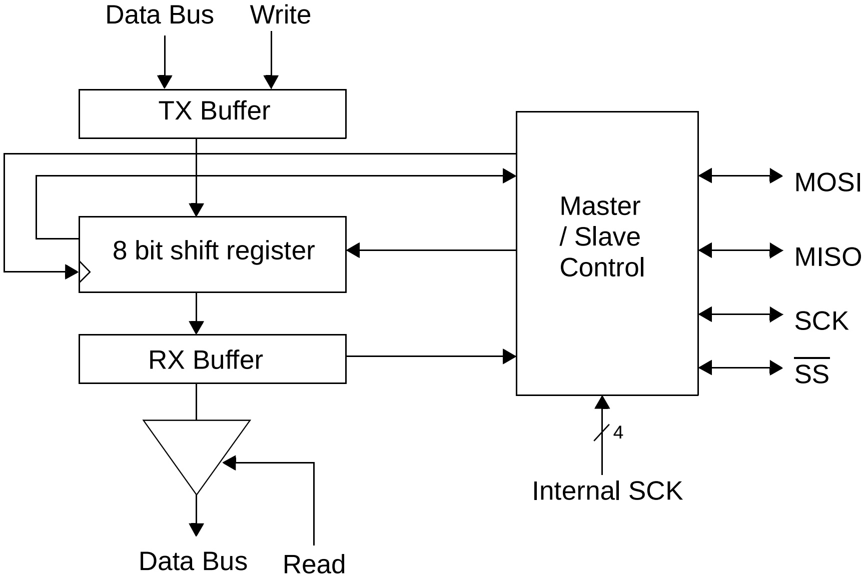


Figure 17-1. SPI Block Diagram

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 (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

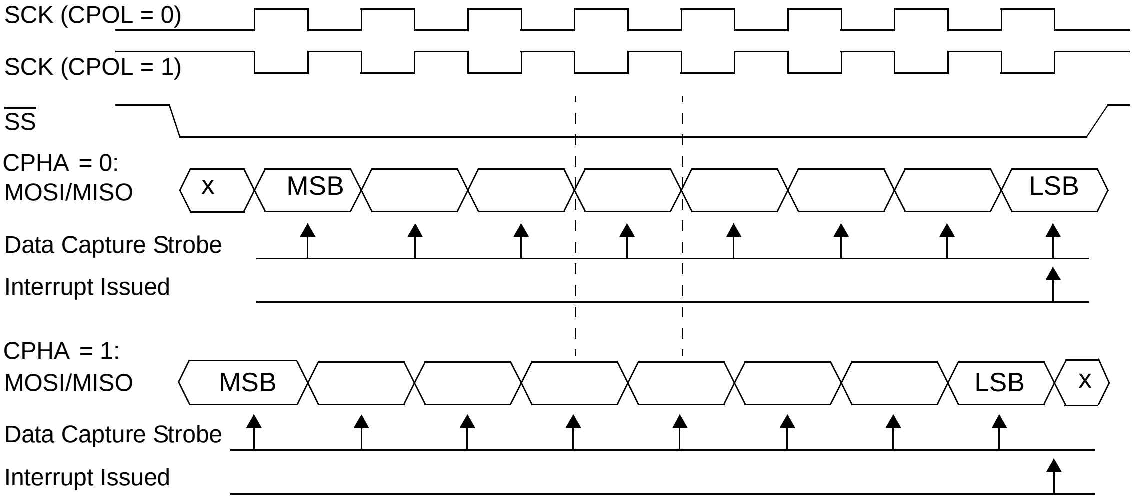


Figure 17-4. SPI Data Timing

17.5 SPI Interrupt

For SPI, an interrupt request is generated after a byte is received or transmitted. See Section 21.3 for details on the SPI interrupt.

17.6 SPI Modes for GPIO Pins

The GPIO pins used for SPI outputs (P0.5–P0.7) contain a bypass mode, as shown in the GPIO block diagram (*Figure 12-1*). Whenever the SPI block is inactive (Mode[5:4] = 00), the bypass value is 1, which enables normal GPIO operation. When SPI master or slave modes are activated, the appropriate bypass signals are driven by the hardware for outputs, and are held at 1 for inputs. **Note that the corresponding data bits in the Port 0 Data Register must be set to 1 for each pin being used for an SPI output.** In addition, the GPIO modes are not affected by operation of the SPI block, so each pin must be programmed by firmware to the desired drive strength mode.

For GPIO pins that are not used for SPI outputs, the SPI bypass value in *Figure 12-1* is always 1, for normal GPIO operation.

Table 17-1. SPI Pin Assignments

SPI Function	GPIO Pin	Comment
Slave Select ($\overline{SS}$)	P0.4	For Master Mode, Firmware sets $\overline{SS}$, may use any GPIO pin. For Slave Mode, $\overline{SS}$ is an active LOW input.
Master Out, Slave In (MOSI)	P0.5	Data output for master, data input for slave.
Master In, Slave Out (MISO)	P0.6	Data input for master, data output for slave.
SCK	P0.7	SPI Clock: Output for master, input for slave.

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

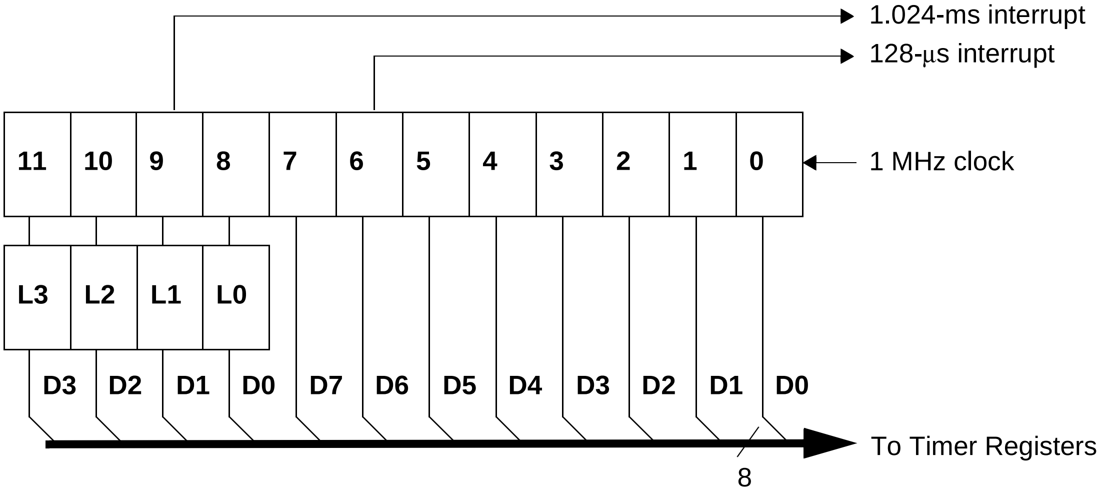


Figure 18-3. Timer Block Diagram

20.0 Processor Status and Control Register

Bit #	7	6	5	4	3	2	1	0
Bit Name	IRQ Pending	Watchdog Reset	Bus Interrupt Event	LVR/BOR Reset	Suspend	Interrupt Enable Sense	Reserved	Run
Read/Write	R	R/W	R/W	R/W	R/W	R	-	R/W
Reset	0	1	0	1	0	0	0	1

Figure 20-1. Processor Status and Control Register (Address 0xFF)

Bit 7: IRQ Pending

When an interrupt is generated, it is registered as a pending interrupt. The interrupt will remain pending until its interrupt enable bit is set (*Figure 21-1* and *Figure 21-2*) and interrupts are globally enabled (Bit 2, Processor Status and Control Register). At that point the internal interrupt handling sequence will clear the IRQ Pending bit until another interrupt is detected as pending. This bit is only valid if the Global Interrupt Enable bit is disabled.

1 = There are pending interrupts.

0 = No pending interrupts.

Bit 6: Watchdog Reset

The Watchdog Timer Reset (WDR) occurs when the internal Watchdog timer rolls over. The timer will roll over and WDR will occur if it is not cleared within t_{WATCH} (see Section 26.0 for the value of t_{WATCH}). This bit is cleared by an LVR/BOR. Note that a Watchdog reset can occur with a POR/LVR/BOR event, as discussed at the end of this section.

1 = A Watchdog reset occurs.

0 = No Watchdog reset

Bit 5: Bus Interrupt Event

The Bus Reset Status is set whenever the event for the USB Bus Reset or PS/2 Activity interrupt occurs. The event type (USB or PS/2) is selected by the state of the USB-PS/2 Interrupt Mode bit in the USB Status and Control Register (see *Figure 13-1*). The details on the event conditions that set this bit are given in Section 21.3. In either mode, this bit is set as soon as the event has lasted for 128–256 µs, and the bit will be set even if the interrupt is not enabled. The bit is only cleared by firmware or LVR/WDR.

1 = A USB reset occurred or PS/2 Activity is detected, depending on USB-PS/2 Interrupt Select bit.

0 = No event detected since last cleared by firmware or LVR/WDR.

Bit 4: LVR/BOR Reset

The Low-voltage or Brown-out Reset is set to ‘1’ during a power-on reset. Firmware can check bits 4 and 6 in the reset handler to determine whether a reset was caused by a LVR/BOR condition or a Watchdog timeout. This bit is not affected by WDR. Note that a LVR/BOR event may be followed by a Watchdog reset before firmware begins executing, as explained at the end of this section.

1 = A POR or LVR has occurred.

0 = No POR nor LVR since this bit last cleared.

Bit 3: Suspend

Writing a '1' to the Suspend bit will halt the processor and cause the microcontroller to enter the suspend mode that significantly reduces power consumption. An interrupt or USB bus activity will cause the device to come out of suspend. After coming out of suspend, the device will resume firmware execution at the instruction following the IOWR which put the part into suspend. When writing the suspend bit with a resume condition present (such as non-idle USB activity), the suspend state will still be entered, followed immediately by the wake-up process (with appropriate delays for the clock start-up). See Section 11.0 for more details on suspend mode operation.

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 21-1*) and USB Endpoint Interrupt Enable Register (*Figure 21-2*). 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 Section 10.1), 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.*

A USB bus reset is indicated by a single ended zero (SE0) on the USB D+ and D– pins. The USB Bus Reset interrupt occurs when the SE0 condition ends. PS/2 activity is indicated by a continuous LOW on the SDATA pin. The PS/2 interrupt occurs as soon as the long LOW state is detected.

During the entire interval of a USB Bus Reset or PS/2 interrupt event, the USB Device Address register is cleared.

The Bus Reset/PS/2 interrupt may occur 128 µs after the bus condition is removed.

1 = Enable

0 = Disable

Bit #	7	6	5	4	3	2	1	0
Bit Name	Reserved					EP2 Interrupt Enable	EP1 Interrupt Enable	EP0 Interrupt Enable
Read/Write	-	-	-	-	-	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Figure 21-2. Endpoint Interrupt Enable Register (Address 0x21)

Bit [7:3]: Reserved.

Bit [2:1]: EP2,1 Interrupt Enable

There are two non-control endpoint (EP2 and EP1) interrupts. If enabled, a non-control endpoint interrupt is generated when:

- The USB host writes valid data to an endpoint FIFO. However, if the endpoint is in ACK OUT modes, an interrupt is generated regardless of data packet validity (i.e., good CRC). Firmware must check for data validity.
- The device SIE sends a NAK or STALL handshake packet to the USB host during the host attempts to read data from the endpoint (INs).
- The device receives an ACK handshake after a successful read transaction (IN) from the host.
- The device SIE sends a NAK or STALL handshake packet to the USB host during the host attempts to write data (OUTs) to the endpoint FIFO.

1 = Enable

0 = Disable

Refer to *Table 22-1* for more information.

Bit 0: EP0 Interrupt Enable

If enabled, a control endpoint interrupt is generated when:

- The endpoint 0 mode is set to accept a SETUP token.
- After the SIE sends a 0-byte packet in the status stage of a control transfer.
- The USB host writes valid data to an endpoint FIFO. However, if the endpoint is in ACK OUT modes, an interrupt is generated regardless of what data is received. Firmware must check for data validity.
- The device SIE sends a NAK or STALL handshake packet to the USB host during the host attempts to read data from the endpoint (INs).
- The device SIE sends a NAK or STALL handshake packet to the USB host during the host attempts to write data (OUTs) to the endpoint FIFO.

1 = Enable EP0 interrupt

0 = Disable EP0 interrupt

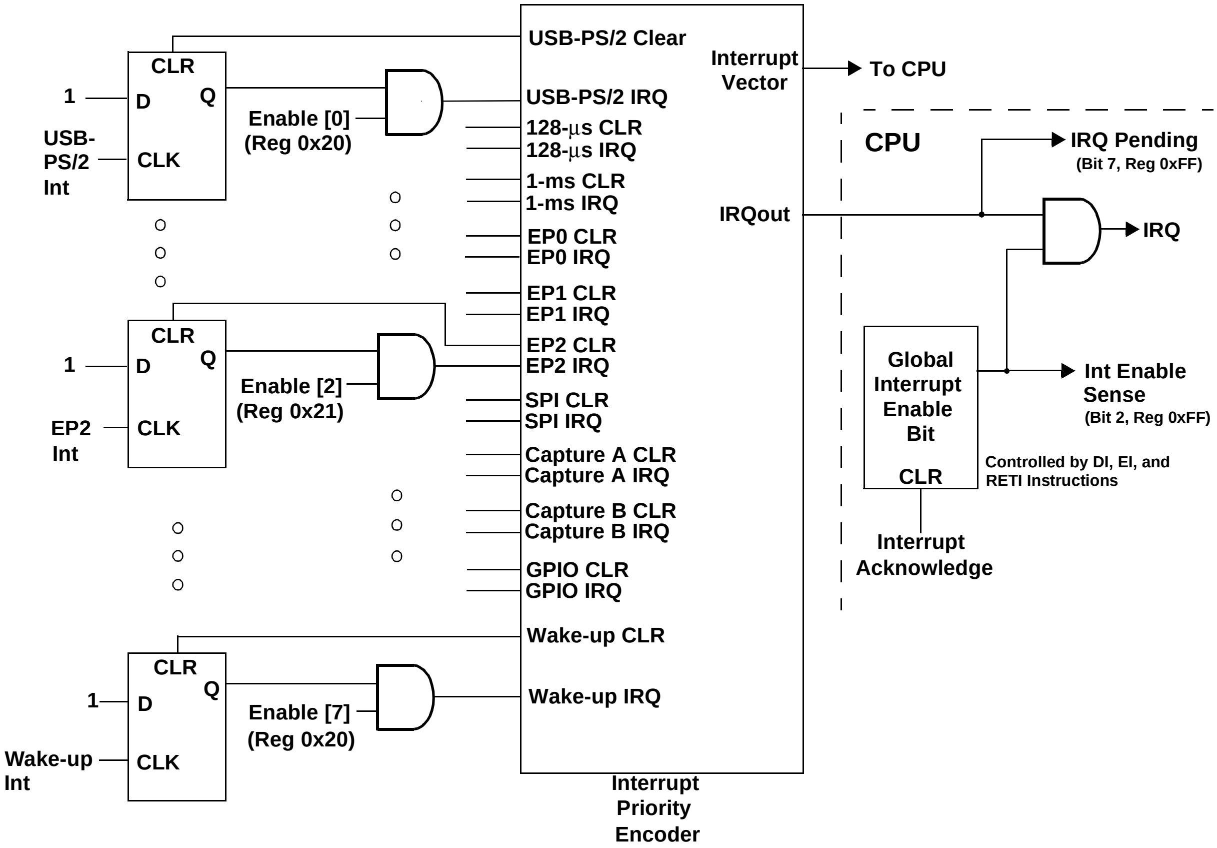


Figure 21-3. Interrupt Controller Logic Block Diagram

Bit #	7	6	5	4	3	2	1	0
Bit Name	P0 Interrupt Enable							
Read/Write	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Figure 21-4. Port 0 Interrupt Enable Register (Address 0x04)

Bit [7:0]: P0 [7:0] Interrupt Enable

1 = Enables GPIO interrupts from the corresponding input pin.

0 = Disables GPIO interrupts from the corresponding input pin.

Bit #	7	6	5	4	3	2	1	0
Bit Name	P1 Interrupt Enable							
Read/Write	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Figure 21-5. Port 1 Interrupt Enable Register (Address 0x05)

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

1 = Enables GPIO interrupts from the corresponding input pin.

0 = Disables GPIO interrupts from the corresponding input pin.

The polarity that triggers an interrupt is controlled independently for each GPIO pin by the GPIO Interrupt Polarity Registers. *Figure 21-6* and *Figure 21-7* control the interrupt polarity of each GPIO pin.

Bit #	7	6	5	4	3	2	1	0
Bit Name	P0 Interrupt Polarity							
Read/Write	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Figure 21-6. Port 0 Interrupt Polarity Register (Address 0x06)

Bit [7:0]: P0[7:0] Interrupt Polarity

1 = Rising GPIO edge

0 = Falling GPIO edge

Bit #	7	6	5	4	3	2	1	0
Bit Name	P1 Interrupt Polarity							
Read/Write	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Figure 21-7. Port 1 Interrupt Polarity Register (Address 0x07)

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

1 = Rising GPIO edge

0 = Falling GPIO edge

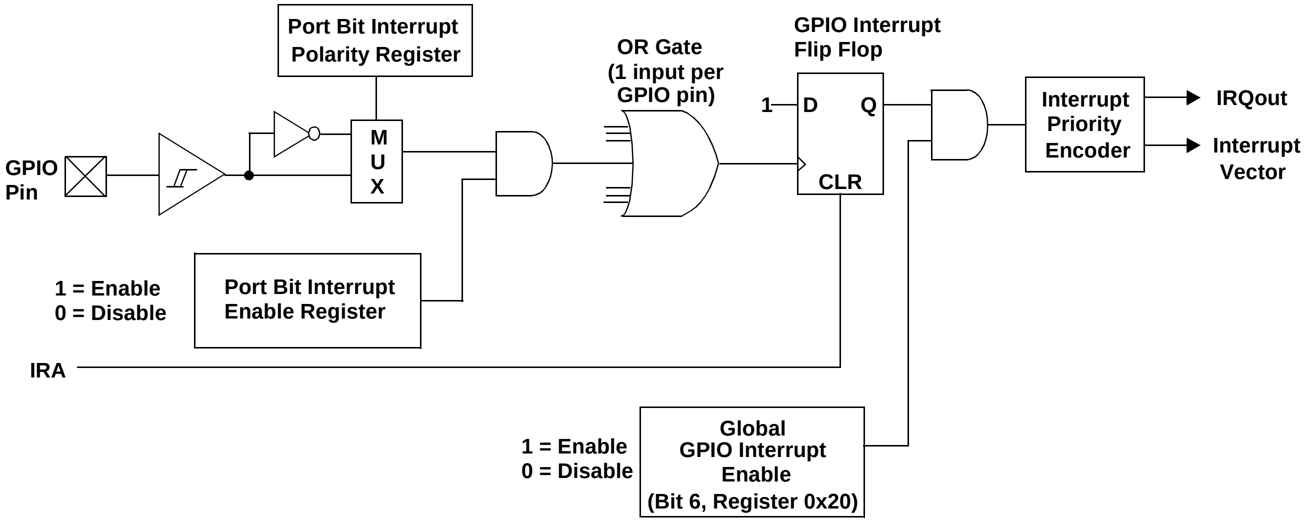

Port Bit Interrupt Polarity Register
GPIO Pin
M U X
OR Gate (1 input per GPIO pin)
GPIO Interrupt Flip Flop
1
D
Q
CLR
Interrupt Priority Encoder
IRQout
Interrupt Vector
1 = Enable
0 = Disable
Port Bit Interrupt Enable Register
IRA
1 = Enable
0 = Disable
Global GPIO Interrupt Enable (Bit 6, Register 0x20)

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”

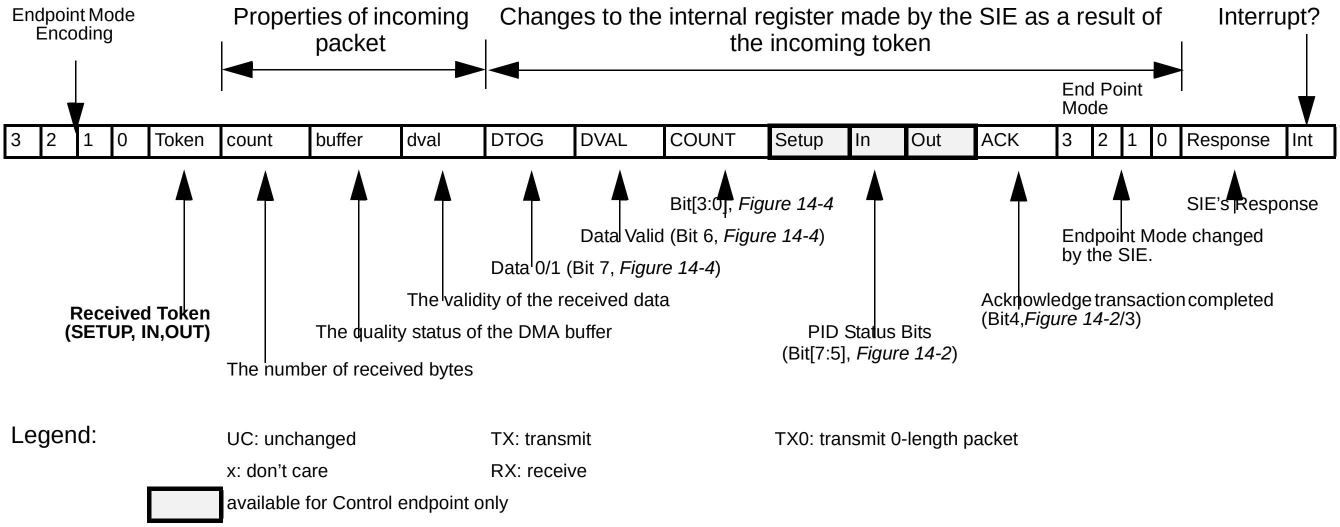

Endpoint Mode Encoding
Properties of incoming packet
Changes to the internal register made by the SIE as a result of the incoming token
Interrupt?
End Point Mode
3
2
1
0
Token
count
buffer
dval
DTOG
DVAL
COUNT
Setup
In
Out
ACK
3
2
1
0
Response
Int
Bit[3:0], Figure 14-4
SIE’s Response
Data Valid (Bit 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																				

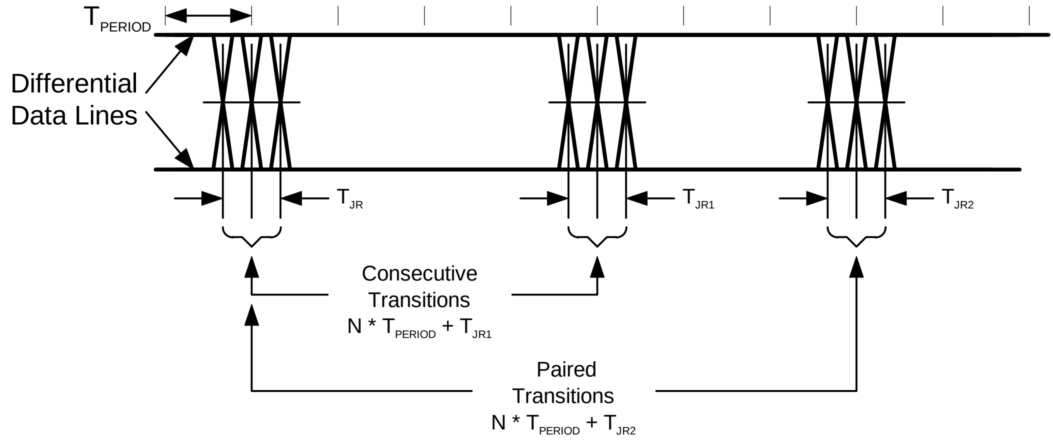


Figure 26-3. Receiver Jitter Tolerance

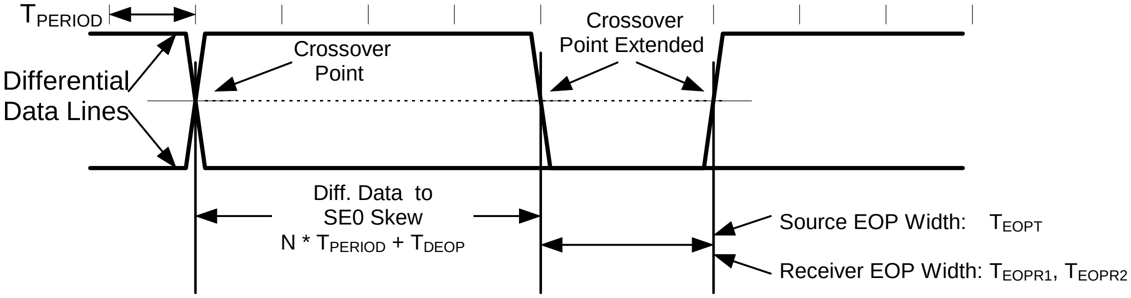

T_{PERIOD}
Differential
Data Lines
Crossover
Point
Crossover
Point Extended
Diff. Data to
SE0 Skew
N * T_{PERIOD} + T_{DEOP}
Source EOP Width: T_{EOPT}
Receiver EOP Width: T_{EOPR1}, T_{EOPR2}

Figure 26-4. Differential to EOP Transition Skew and EOP Width

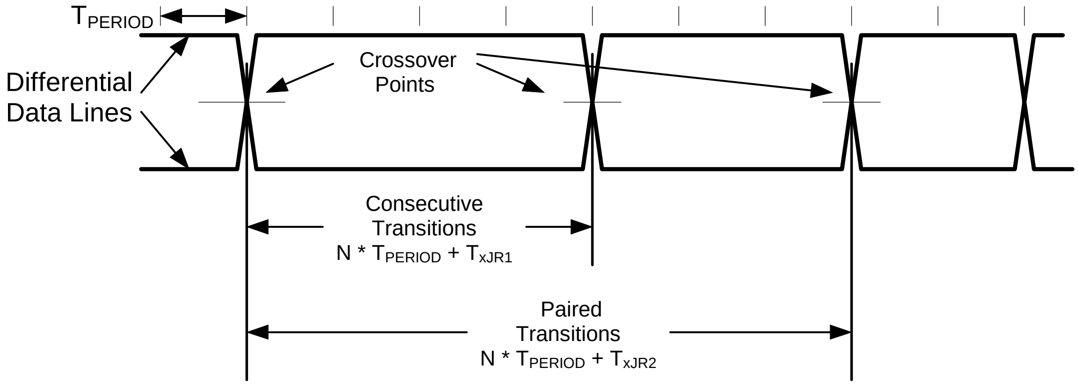


Figure 26-5. Differential Data Jitter

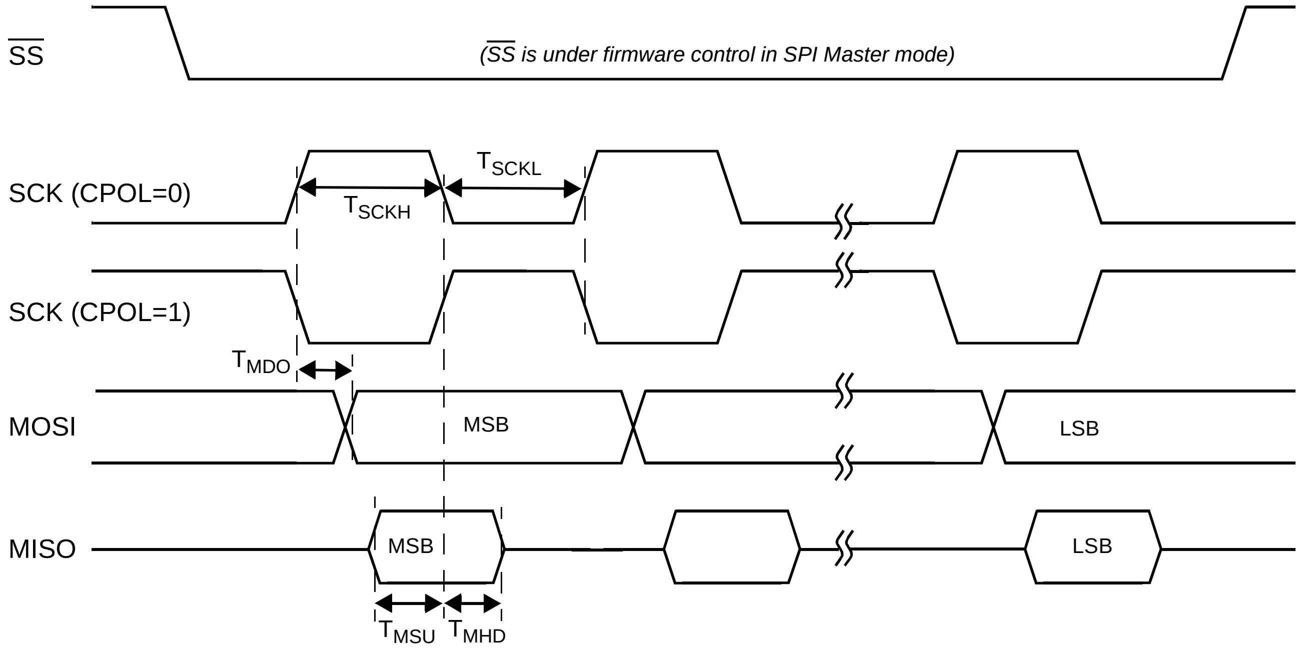


Figure 26-6. SPI Master Timing, CPHA = 0

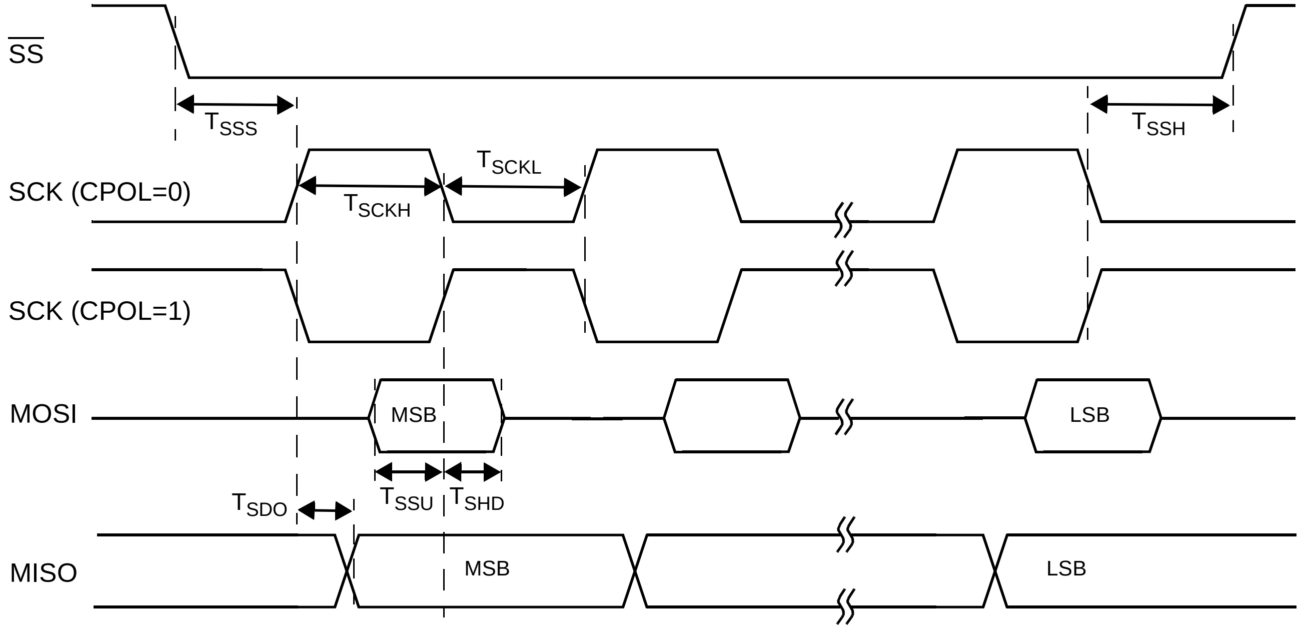


Figure 26-7. SPI Slave Timing, CPHA = 0

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

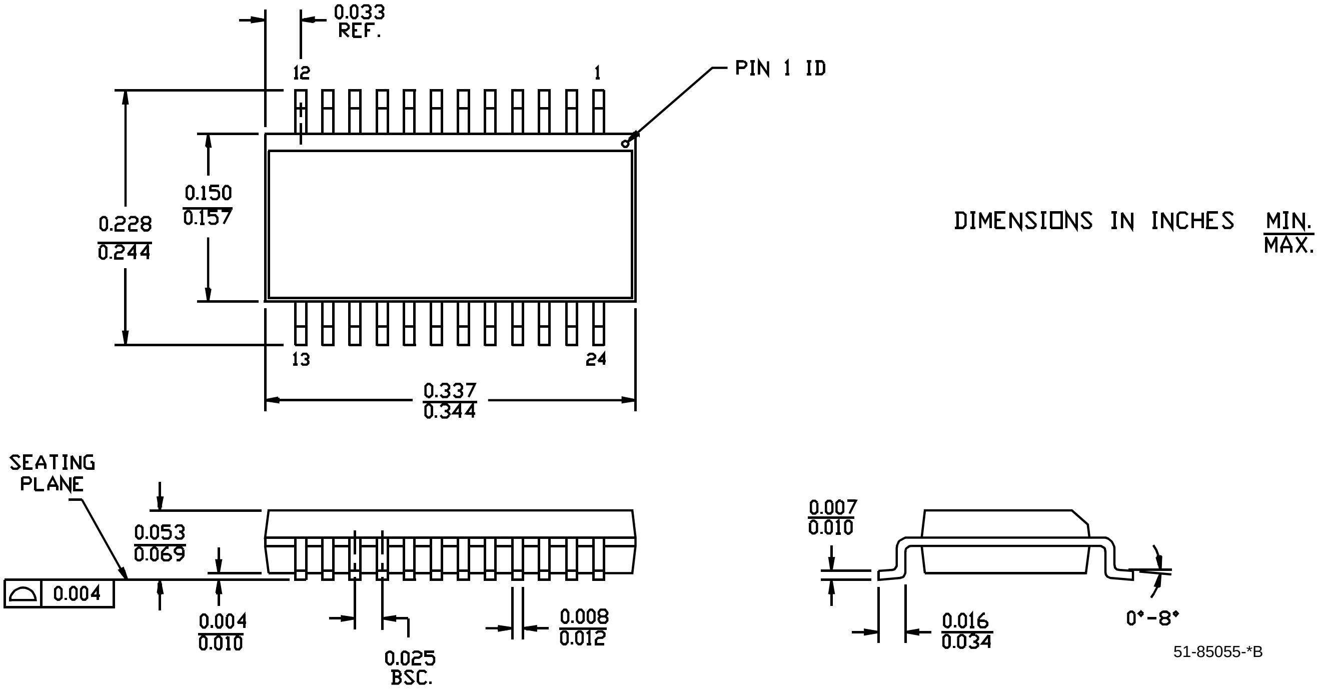
28.0 Package Diagrams

18-Lead (300-Mil) Molded DIP P3

51-85010-*A

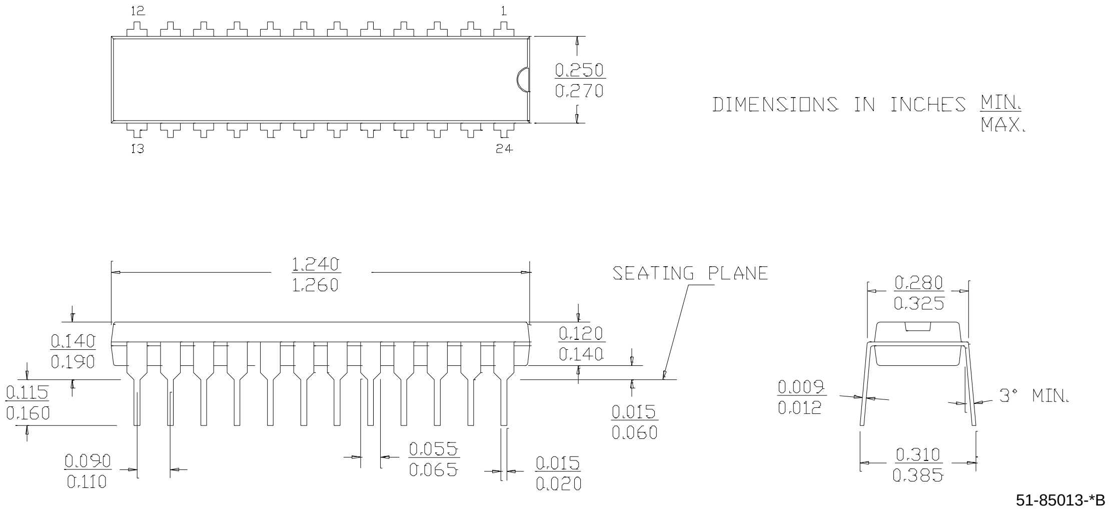
28.0 Package Diagrams (continued)

24-Lead Quarter Size Outline Q13



51-85055-*B

24-Lead (300-Mil) PDIP P13



51-85013-*B