Welcome to E-XFL.COM

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

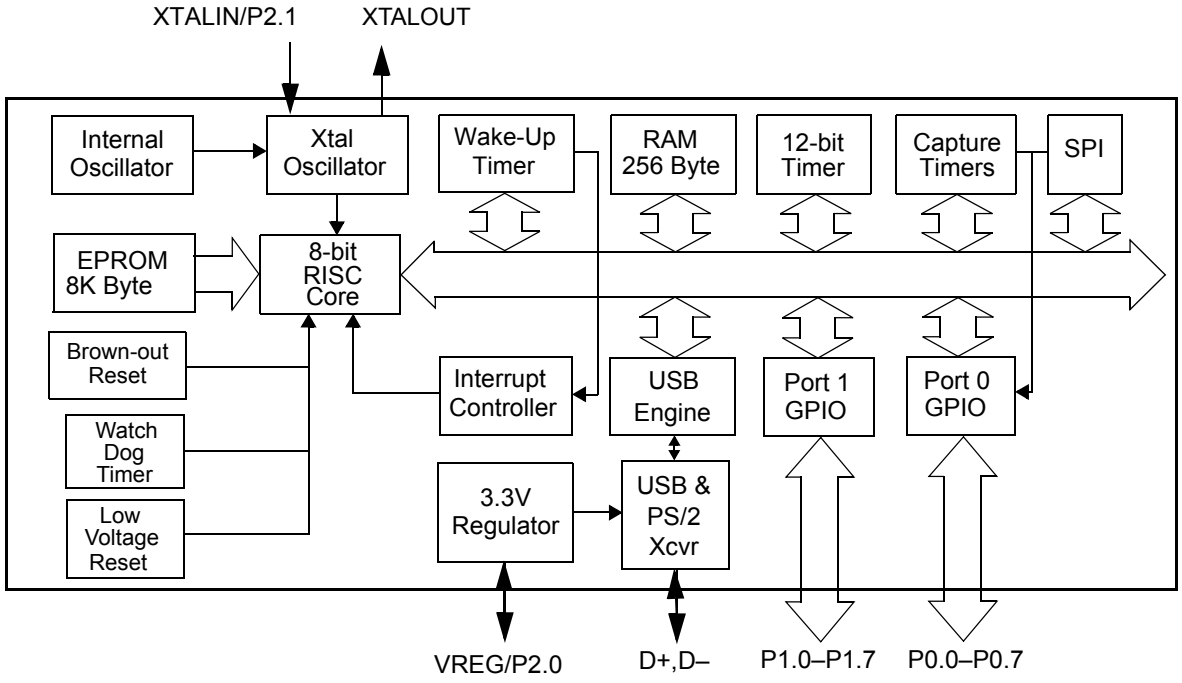
Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering 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-SSOP (0.154", 3.90mm Width)
Supplier Device Package	24-QSOP
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy7c63743c-qxc

Logic Block Diagram



Programming Model

Refer to the *CYASM Assembler User's Guide* for more details on firmware operation with the CY7C637xxC microcontrollers.

Program Counter (PC)

The 14-bit program counter (PC) allows access for up to 8 Kbytes of EPROM using the CY7C637xxC architecture. The program counter is cleared during reset, such that the first instruction executed after a reset is at address 0x0000. This instruction is typically a jump instruction to a reset handler that initializes the application.

The lower 8 bits of the program counter are incremented as instructions are loaded and executed. The upper six bits of the program counter are incremented by executing an XPAGE instruction. As a result, the last instruction executed within a 256-byte "page" of sequential code should be an XPAGE instruction. The assembler directive "XPAGEON" will cause the assembler to insert XPAGE instructions automatically. As instructions can be either one or two bytes long, the assembler may occasionally need to insert a NOP followed by an XPAGE for correct execution.

The program counter of the next instruction to be executed, carry flag, and zero flag are saved as two bytes on the program stack during an interrupt acknowledge or a CALL instruction. The program counter, carry flag, and zero flag are restored from the program stack only during a RETI instruction.

Please note the program counter cannot be accessed directly by the firmware. The program stack can be examined by reading SRAM from location 0x00 and up.

8-bit Accumulator (A)

The accumulator is the general-purpose, do everything register in the architecture where results are usually calculated.

8-bit Index Register (X)

The index register "X" is available to the firmware as an auxiliary accumulator. The X register also allows the processor to perform indexed operations by loading an index value into X.

8-bit Program Stack Pointer (PSP)

During a reset, the program stack pointer (PSP) is set to zero. This means the program "stack" starts at RAM address 0x00 and "grows" upward from there. Note that the program stack pointer is directly addressable under firmware control, using the MOV PSP,A instruction. The PSP supports interrupt service under hardware control and CALL, RET, and RETI instructions under firmware control.

During an interrupt acknowledge, interrupts are disabled and the program counter, carry flag, and zero flag are written as two bytes of data memory. The first byte is stored in the memory addressed by the program stack pointer, then the PSP is incremented. The second byte is stored in memory addressed by the program stack pointer and the PSP is incremented again. The net effect is to store the program counter and flags on the program "stack" and increment the program stack pointer by two.

The return from interrupt (RETI) instruction decrements the program stack pointer, then restores the second byte from memory addressed by the PSP. The program stack pointer is decremented again and the first byte is restored from memory addressed by the PSP. After the program counter and flags have been restored from stack, the interrupts are enabled. The effect is to restore the program counter and flags from the program stack, decrement the program stack pointer by two, and re-enable interrupts.

The call subroutine (CALL) instruction stores the program counter and flags on the program stack and increments the PSP by two.

The return from subroutine (RET) instruction restores the program counter, but not the flags, from program stack and decrements the PSP by two.

Note that there are restrictions in using the JMP, CALL, and INDEX instructions across the 4-KByte boundary of the program memory. Refer to the *CYASM Assembler User's Guide* for a detailed description.

8-bit Data Stack Pointer (DSP)

The data stack pointer (DSP) supports PUSH and POP instructions that use the data stack for temporary storage. A PUSH instruction will pre-decrement the DSP, then write data to the memory location addressed by the DSP. A POP instruction will read data from the memory location addressed by the DSP, then post-increment the DSP.

During a reset, the Data Stack Pointer will be set to zero. A PUSH instruction when DSP equals zero will write data at the top of the data RAM (address 0xFF). This would write data to the memory area reserved for a FIFO for USB endpoint 0. In non-USB applications, this works fine and is not a problem.

For USB applications, the firmware should set the DSP to an appropriate location to avoid a memory conflict with RAM dedicated to USB FIFOs. The memory requirements for the USB endpoints are shown in Data Memory Organization on page 10 . For example, assembly instructions to set the DSP to 20h (giving 32 bytes for program and data stack combined) are shown below.

```
MOV A,20h     ; Move 20 hex into Accumulator (must be D8h or less to avoid USB FIFOs)
SWAP A,DSP  ; swap accumulator value into DSP register
```

Address Modes

The CY7C637xxC microcontrollers support three addressing modes for instructions that require data operands: data, direct, and indexed.

Data

The "Data" address mode refers to a data operand that is actually a constant encoded in the instruction. As an example, consider the instruction that loads A with the constant 0x30:

- MOV A, 30h

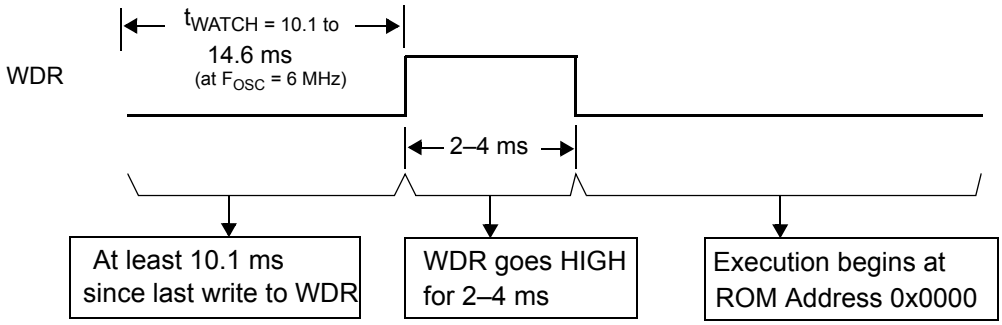
This instruction will require two bytes of code where the first byte identifies the "MOV A" instruction with a data operand as the second byte. The second byte of the instruction will be the constant "0xE8h". A constant may be referred to by name if a prior "EQU" statement assigns the constant value to the name. For example, the following code is equivalent to the example shown above.

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

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

Figure 5. Watchdog Reset (WDR, Address 0x26)



Suspend Mode

The CY7C637xxC parts support a versatile low-power suspend mode. In suspend mode, only an enabled interrupt or a LOW state on the D–/SDATA pin will wake the part. Two options are available. For lowest power, all internal circuits can be disabled, so only an external event will resume operation. Alternatively, a low-power internal wake-up timer can be used to trigger the wake-up interrupt. This timer is described in Wake-up Timer on page 16, and can be used to periodically poll the system to check for changes, such as looking for movement in a mouse, while maintaining a low average power.

The CY7C637xxC is placed into a low-power state by setting the Suspend bit of the Processor Status and Control Register (Figure 34). All logic blocks in the device are turned off except the GPIO interrupt logic, the D–/SDATA pin input receiver, and (optionally) the wake-up timer. The clock oscillators, as well as the free-running and Watchdog timers are shut down. Only the occurrence of an enabled GPIO interrupt, wake-up interrupt, SPI slave interrupt, or a LOW state on the D–/SDATA pin will wake the part from suspend (D– LOW indicates non-idle USB activity). Once one of these resuming conditions occurs, clocks will be restarted and the device returns to full operation after the oscillator is stable and the selected delay period expires. This delay period is determined by selection of internal versus external clock, and by the state of the Ext. Clock Resume Delay as explained in Clocking Mode on Wake-up from Suspend on page 15.

In suspend mode, any enabled and pending interrupt will wake the part up. The state of the Interrupt Enable Sense bit (Bit 2, Figure 34 on page 30) does not have any effect. As a result, any interrupts not intended for waking from suspend should be disabled through the Global Interrupt Enable Register and the USB End Point Interrupt Enable Register (Interrupts).

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

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

To achieve the lowest possible current during suspend mode, all I/O should be held at either V_{CC} or ground. In addition, the GPIO *bit* interrupts (Figure on page 34 and Figure on page 34) should be disabled for any pins that are not being used for a wake-up interrupt. This should be done even if the main GPIO Interrupt Enable (Figure 35 on page 32) is off.

Typical code for entering suspend is shown below:

```
...             ; All GPIO set to low-power state (no floating
                  pins, and bit interrupts disabled unless using
                  for wake-up)
...             ; Enable GPIO and/or wake-up timer
                  interrupts if desired for wake-up
...             ; Select clock mode for wake-up (see Clocking
                  Mode on Wake-up from Suspend on page
                  15)
mov a, 09h      ; Set suspend and run bits
iowr FFh        ; Write to Status and Control Register – Enter
                  suspend, wait for GPIO/wake-up interrupt or
                  USB activity
nop             ; This executes before any ISR
...             ; Remaining code for exiting suspend routine
```

Clocking Mode on Wake-up from Suspend

When exiting suspend on a wake-up event, the device can be configured to run in either Internal or External Clock mode. The mode is selected by the state of the External Oscillator Enable bit in the Clock Configuration Register (Figure 4 on page 12). Using the Internal Clock saves the external oscillator start-up time and keeps that oscillator off for additional power savings. The external oscillator mode can be activated when desired, similar to operation at power-up.

The sequence of events for these modes is as follows:

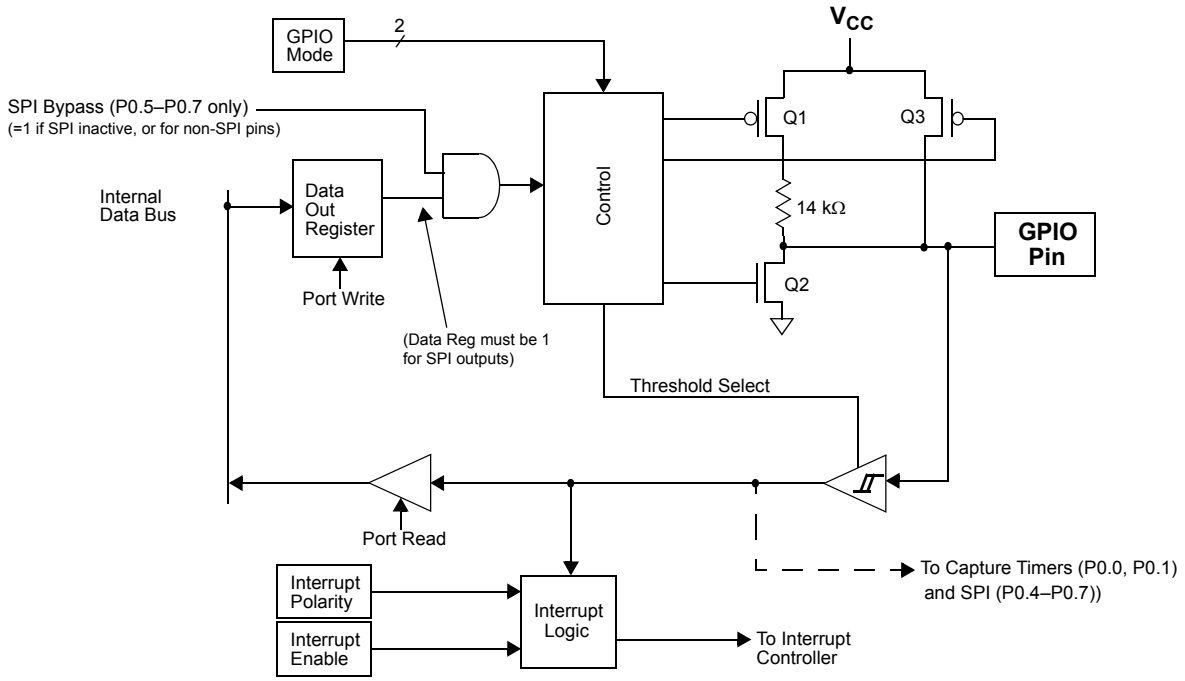
Wake in Internal Clock Mode:

1. Before entering suspend, clear bit 0 of the Clock Configuration Register. This selects Internal clock mode after suspend.
2. Enter suspend mode by setting the suspend bit of the Processor Status and Control Register.
3. After a wake-up event, the internal clock starts immediately (within 2 µs).
4. A time-out period of 8 µs passes, and then firmware execution begins.

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

USB Non-control Endpoints

The CY7C637xxC feature two non-control endpoints, endpoint 1 (EP1) and endpoint 2 (EP2). The EP1 and EP2 Mode Registers do not have the locking mechanism of the EP0 Mode Register. The EP1 and EP2 Mode Registers use the format shown in Figure 17. EP1 uses an 8-byte FIFO at SRAM locations 0xF0–0xF7, EP2 uses an 8-byte FIFO at SRAM locations 0xE8–0xEF as shown in Data Memory Organization on page 10.

Figure 17. USB Endpoint EP1, EP2 Mode Registers (Addresses 0x14 and 0x16)

Bit #	7	6	5	4	3	2	1	0
Bit Name	STALL	Reserved		ACKed Transaction	Mode Bit			
Read/ Write	R/W	-	-	R/C	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7: STALL

1 = The SIE will stall an OUT packet if the Mode Bits are set to ACK-OUT, and the SIE will stall an IN packet if the mode bits are set to ACK-IN. See USB Mode Tables on page 36 for the available modes.

0 = This bit must be set to LOW for all other modes.

Bit [6:5]: Reserved. Must be written to zero during register writes.

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 EP1 and EP2 Mode Bits operate in the same manner as the EP0 Mode Bits (see USB Mode Tables on page 36).

USB Endpoint Counter Registers

There are three Endpoint Counter registers, with identical formats for both control and non-control endpoints. These registers contain byte count information for USB transactions, as well as bits for data packet status. The format of these registers is shown in Figure 18.

Figure 18. Endpoint 0,1,2 Counter Registers (Addresses 0x11, 0x13 and 0x15)

Bit #	7	6	5	4	3	2	1	0
Bit Name	Data Toggle	Data Valid	Reserved		Byte Count			
Read/Write	R/W	R/W	-	-	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit 7: Data Toggle

This bit selects the DATA packet's toggle state. For IN transactions, firmware must set this bit to the select the transmitted Data Toggle. For OUT or SETUP transactions, the hardware sets this bit to the state of the received Data Toggle bit.

1 = DATA1

0 = DATA0

Bit 6: Data Valid

This bit is used for OUT and SETUP tokens only. This bit is cleared to '0' if CRC, bitstuff, or PID errors have occurred. This bit does not update for some endpoint mode settings. Refer to *Table 10* for more details.

1 = Data is valid.

0 = Data is invalid. If enabled, the endpoint interrupt will occur even if invalid data is received.

Bit [5:4]: Reserved

Bit [3:0]: Byte Count Bit [3:0]

Byte Count Bits indicate the number of data bytes in a transaction: For IN transactions, firmware loads the count with the number of bytes to be transmitted to the host from the endpoint FIFO. Valid values are 0 to 8 inclusive. For OUT or SETUP transactions, the count is updated by hardware to the number of data bytes received, plus 2 for the CRC bytes. Valid values are 2 to 10 inclusive.

For Endpoint 0 Count Register, whenever the count updates from a SETUP or OUT transaction, the count register locks and cannot be written by the CPU. Reading the register unlocks it. This prevents firmware from overwriting a status update on incoming SETUP or OUT transactions before firmware has a chance to read the data.

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 DC Characteristics on page 42).

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 14*). 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 13 on page 19 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 14*), 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 15) 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 CY7C637xxC 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.

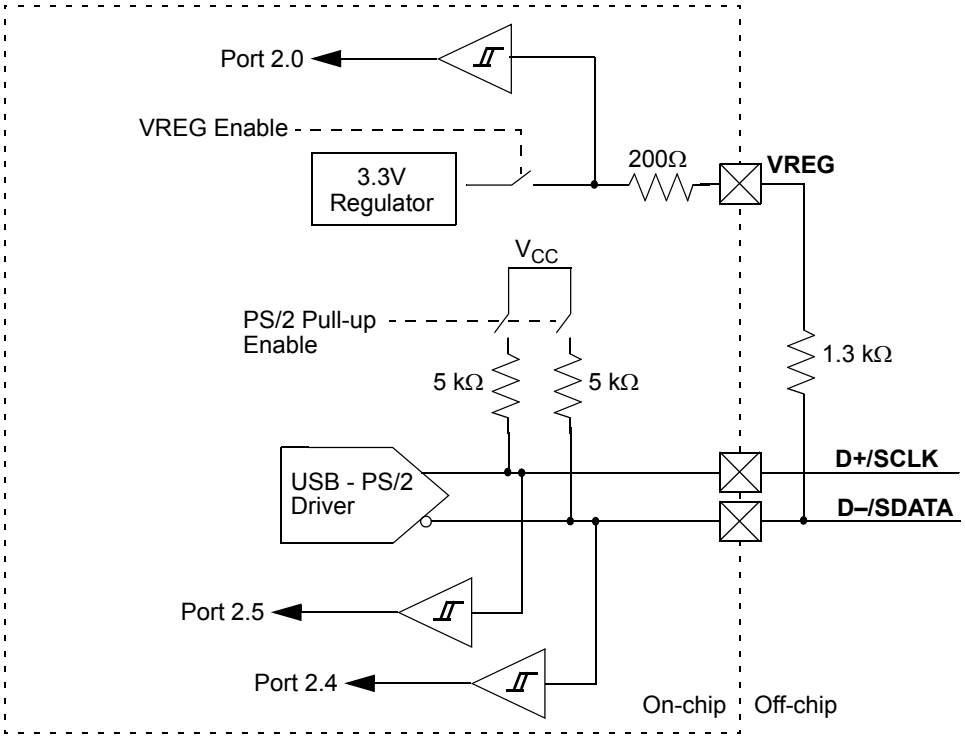
PS/2 Operation

The CY7C637xxC 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 15).
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 USB Port Status and Control on page 19 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 14*).
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 19.

Figure 19. Diagram of USB-PS/2 System Connections

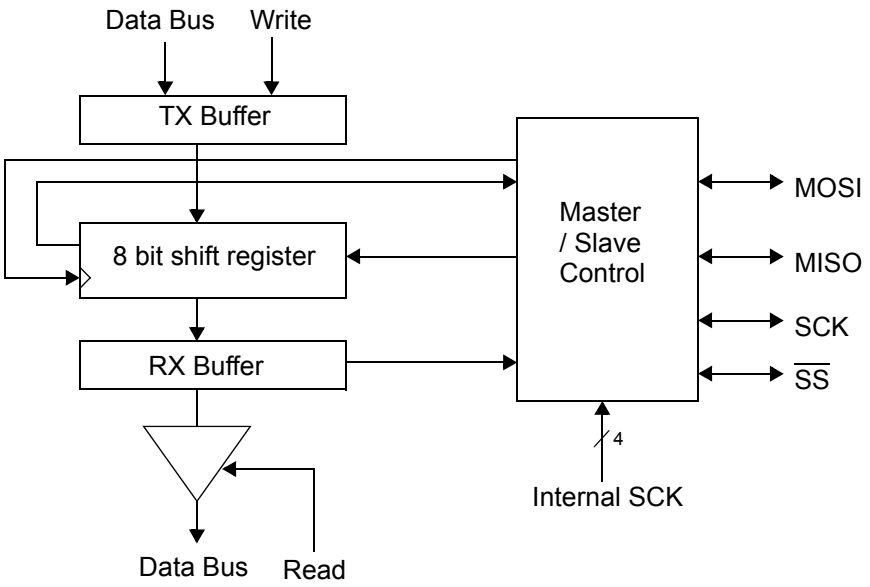


Serial Peripheral Interface (SPI)

SPI is a four-wire, full-duplex serial communication interface between a master device and one or more slave devices. The CY7C637xxC SPI circuit supports byte serial transfers in either Master or Slave modes. The block diagram of the SPI circuit is shown in Figure 20. The block contains buffers for both transmit and receive data for maximum flexibility and throughput. The CY7C637xxC 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 20. SPI Block Diagram



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

Figure 21. SPI Data Register (Address 0x60)

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

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

Operation as an SPI Master

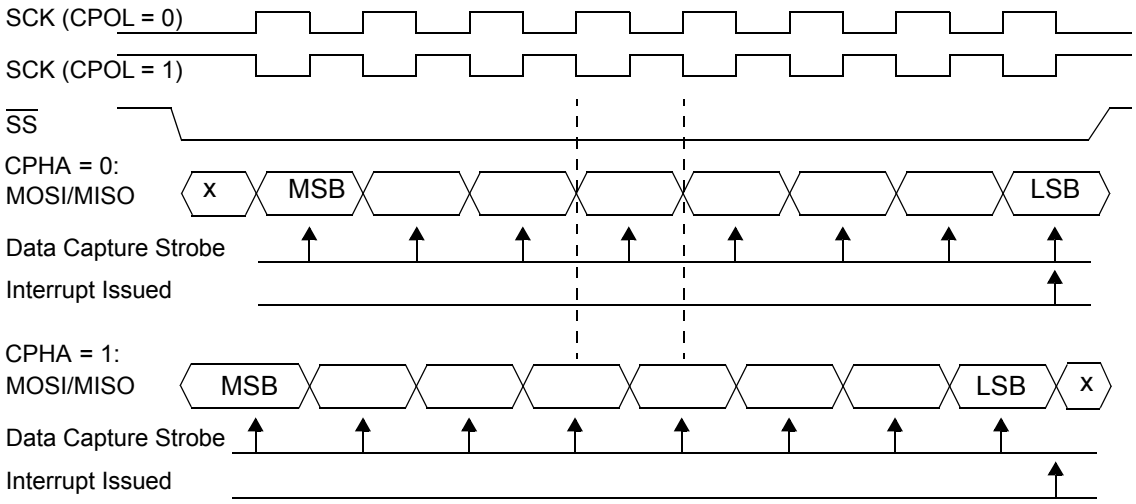
Only an SPI Master can initiate a byte/data transfer. This is done by the Master writing to the SPI Data Register. The Master shifts out 8 bits of data (MSB first) along with the serial clock SCK for the Slave. The Master's outgoing byte is replaced with an incoming one from a Slave device. When the 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 General Purpose I/O Ports on page 17 for GPIO configuration details.

Figure 23. SPI Data Timing



SPI Interrupt

For SPI, an interrupt request is generated after a byte is received or transmitted. See Interrupt Sources on page 32 for details on the SPI interrupt.

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 6 on page 17). 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 6 on page 17 is always 1, for normal GPIO operation.

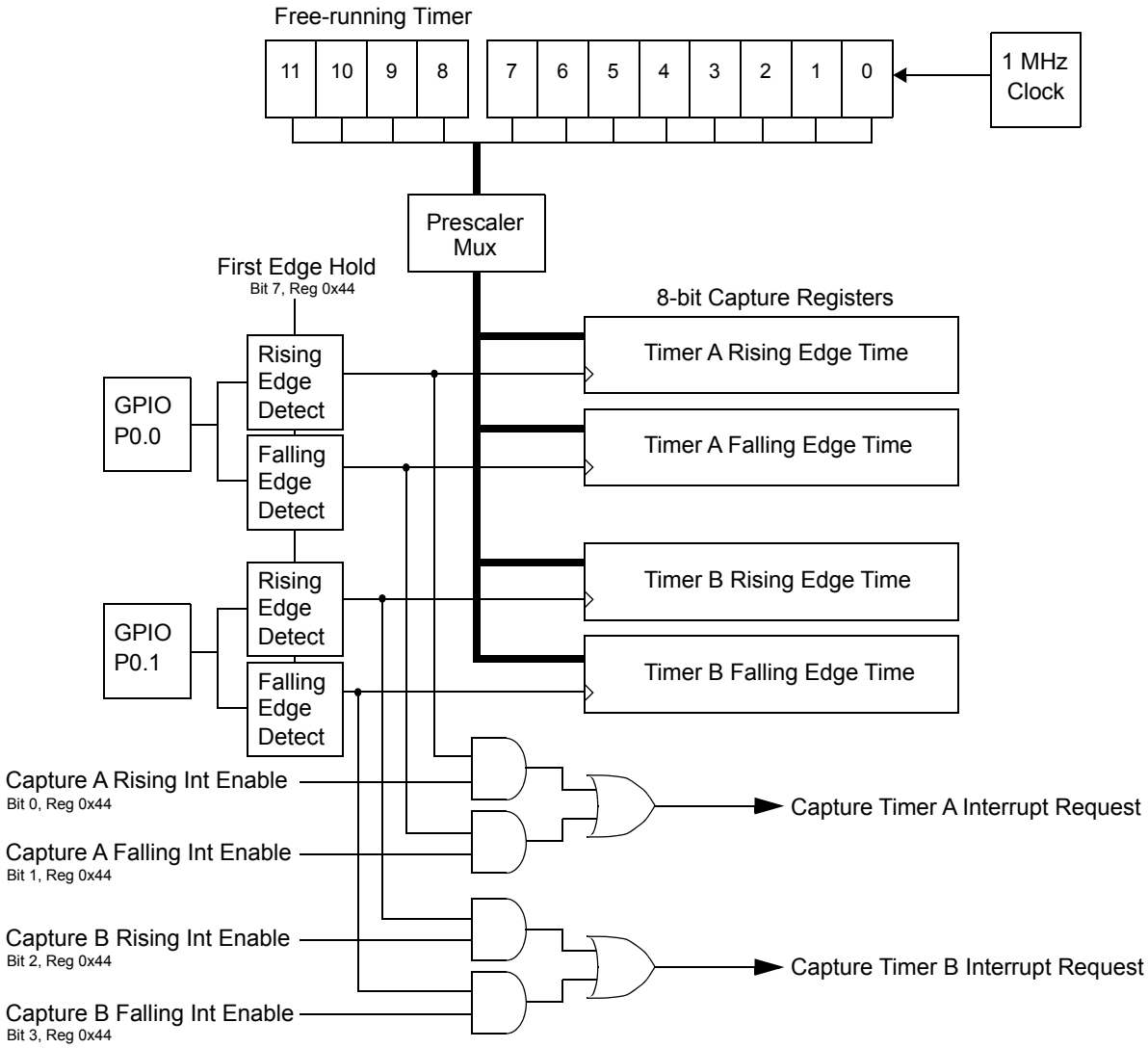
Table 5. SPI Pin Assignments

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.

Timer Capture Registers

Four 8-bit capture timer registers provide both rising- and falling-edge event timing capture on two pins. Capture Timer A is connected to Pin 0.0, and Capture Timer B is connected to Pin 0.1. These can be used to mark the time at which a rising or falling event occurs at the two GPIO pins. Each timer will capture eight bits of the free-running timer into its Capture Timer Data Register if a rising or falling edge event that matches the specified rising or falling edge condition at the pin. A prescaler allows selection of the capture timer tick size. Interrupts can be individually enabled for the four capture registers. A block diagram is shown in Figure 27.

Figure 27. Capture Timers Block Diagram



0 = Disable interrupt

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

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

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

Processor Status and Control Register

Figure 34. Processor Status and Control Register (Address 0xFF)

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

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 35 and Figure 36) 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 Switching Characteristics on page 44 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 14). The details on the event conditions that set this bit are given in Interrupt Sources on page 32. 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 Suspend Mode on page 15 for more details on suspend mode operation.

Figure 40. Port 1 Interrupt Polarity Register (Address 0x07)

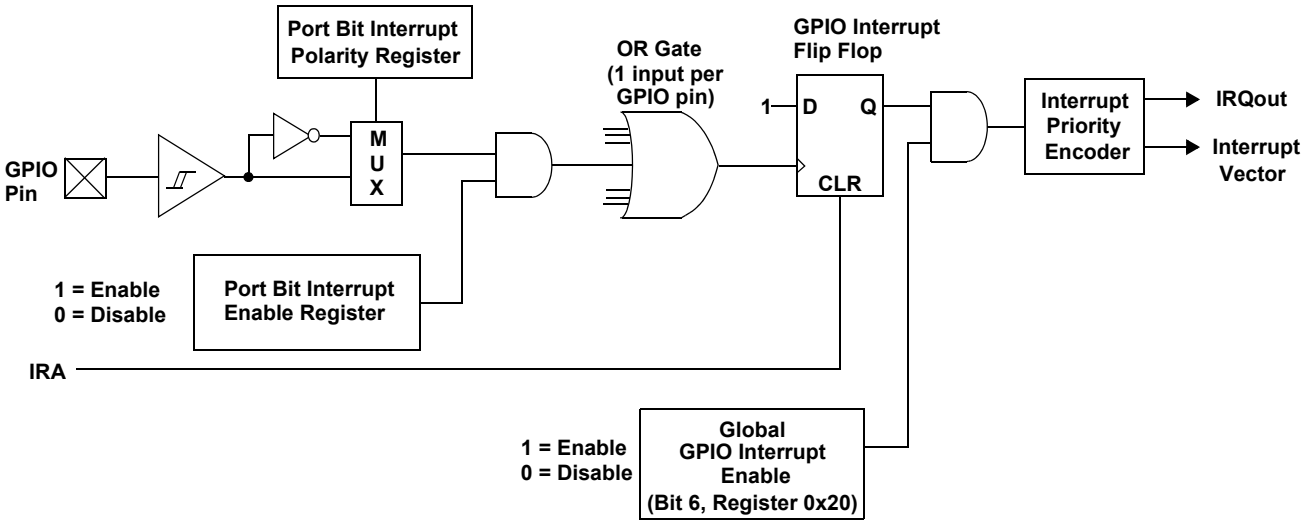
Bit #	7	6	5	4	3	2	1	0
Bit Name	P1 Interrupt Polarity							
Read/Write	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

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

1 = Rising GPIO edge

0 = Falling GPIO edge

Figure 41. GPIO Interrupt Diagram



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 8. 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[4]=0) ACK OUT(STALL[4]=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[4]=0) ACK IN(STALL[4]=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
4. STALL bit is the bit 7 of the USB Non-Control Device Endpoint Mode registers. Refer to USB Non-control Endpoints on page 22 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 16 and Figure 17). 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 16) are set to '0001', which is NAK IN/OUT mode as shown in Table 8 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 USB Serial Interface Engine (SIE) on page 19.

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 10 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 18) 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

Table 10. Details of Modes for Differing Traffic Conditions (continued)

1	0	1	1	IN	x	UC	x	UC	UC	UC	UC	1	UC	UC	No-Change	NAK	yes
NAK OUT/Status IN																	
1	0	1	0	OUT	<= 10	UC	valid	UC	UC	UC	UC	UC	1	UC	No-Change	NAK	yes
1	0	1	0	OUT	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	No-Change	Ignore	no
1	0	1	0	OUT	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	No-Change	Ignore	no
1	0	1	0	IN	x	UC	x	UC	UC	UC	UC	1	UC	1	No-Change	TX 0 Byte	yes
Status IN Only																	
0	1	1	0	OUT	<= 10	UC	valid	UC	UC	UC	UC	UC	1	UC	0	0	1	1	STALL	yes
0	1	1	0	OUT	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	No-Change	Ignore	no
0	1	1	0	OUT	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	No-Change	Ignore	no
0	1	1	0	IN	x	UC	x	UC	UC	UC	UC	1	UC	1	No-Change	TX 0 Byte	yes
Control Read																	
ACK IN/Status OUT																	
1	1	1	1	OUT	2	UC	valid	1	1	updates	UC	UC	1	1	No-Change	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	No-Change	Ignore	no
1	1	1	1	OUT	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	No-Change	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	No-Change	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	No-Change	Ignore	no
1	1	1	0	OUT	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	No-Change	Ignore	no
1	1	1	0	IN	x	UC	x	UC	UC	UC	UC	1	UC	UC	No-Change	NAK	yes
Status OUT Only																	
0	0	1	0	OUT	2	UC	valid	1	1	updates	UC	UC	1	1	No-Change	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	No-Change	Ignore	no
0	0	1	0	OUT	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	No-Change	Ignore	no
0	0	1	0	IN	x	UC	x	UC	UC	UC	UC	1	UC	UC	0	0	1	1	STALL	yes

Absolute Maximum Ratings

Storage Temperature ..... –65 °C to +150 °C
Ambient Temperature with Power Applied .... –0 °C to +70 °C
Supply Voltage on V_{CC} Relative to V_{SS} .....–0.5 V to +7.0 V
DC Input Voltage ..... –0.5 V to +V_{CC}+0.5 V
DC Voltage Applied to Outputs in High Z State..... –0.5 V to + V_{CC}+0.5 V
Maximum Total Sink Output Current into Port 0 and 1 and Pins ..... 70 mA
Maximum Total Source Output Current into Port 0 and 1 and Pins ..... 30 mA
Maximum On-chip Power Dissipation on any GPIO Pin ..... 50 mW
Power Dissipation..... 300 mW
Static Discharge Voltage ..... > 2000 V
Latch-up Current ..... > 200 mA

DC Characteristics

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

	Parameter	Conditions	Min.	Max.	Unit
	General				
V_{CC1}	Operating Voltage	Note 5	V_{LVR}	5.5	V
V_{CC2}	Operating Voltage	Note 5	4.35	5.25	V
I_{CC1}	V_{CC} Operating Supply Current – Internal Oscillator Mode Typical I_{CC1} = 16 mA[6]	V_{CC} = 5.5 V, no GPIO loading V_{CC} = 5.0 V. T = Room Temperature		20	mA
I_{CC2}	V_{CC} Operating Supply Current – External Oscillator Mode Typical I_{CC2}= 13 mA[6]	V_{CC} = 5.5 V, no GPIO loading V_{CC} = 5.0 V. T = Room Temperature		17	mA
I_{SB1}	Standby Current – No Wake-up Osc	Oscillator off, D– > 2.7 V		25	µA
I_{SB2}	Standby Current – With Wake-up Osc	Oscillator off, D– > 2.7 V		75	µA
V_{PP}	Programming Voltage (disabled)		–0.4	0.4	V
T_{RSNTR}	Resonator Start-up Interval	V_{CC} = 5.0 V, ceramic resonator		256	µs
I_{IL}	Input Leakage Current	Any I/O pin		1	µA
I_{SNK}	Max I_{SS} GPIO Sink Current	Cumulative across all ports[7]		70	mA
I_{SRC}	Max I_{CC} GPIO Source Current	Cumulative across all ports[7]		30	mA
	Low-Voltage and Power-on Reset				
V_{LVR}	Low-Voltage Reset Trip Voltage	V_{CC} below V_{LVR} for >100 ns[8]	3.5	4.0	V
t_{VCCS}	V_{CC} Power-on Slew Time	linear ramp: 0 to 4 V[9]		100	ms
	USB Interface				
V_{REG}	VREG Regulator Output Voltage	Load = R_{PU} +R_{PD}[10, 11]	3.0	3.6	V
C_{REG}	Capacitance on VREG Pin	External cap not required		300	pF
V_{OHU}	Static Output High, driven	R_{PD} to Gnd[5]	2.8	3.6	V

Notes
5. Full functionality is guaranteed in V_{CC1} range, except USB transmitter specifications and GPIO output currents are guaranteed for V_{CC2} range.
6. Bench measurements taken under nominal operating conditions. Spec cannot be guaranteed at final test.
7. Total current cumulative across all Port pins, limited to minimize Power and Ground-Drop noise effects.
8. LVR is automatically disabled during suspend mode.
9. LVR will re-occur whenever V_{CC} drops below V_{LVR}. In suspend or with LVR disabled, BOR occurs whenever V_{CC} drops below approximately 2.5V.
10. V_{REG} specified for regulator enabled, idle conditions (i.e., no USB traffic), with load resistors listed. During USB transmits from the internal SIE, the VREG output is not regulated, and should not be used as a general source of regulated voltage in that case. During receive of USB data, the VREG output drops when D– is LOW due to internal series resistance of approximately 200Ω at the VREG pin.
11. In suspend mode, V_{REG} is only valid if R_{PU} is connected from D– to VREG pin, and R_{PD} is connected from D– to ground.

Switching Characteristics (continued)

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

Figure 42. Clock Timing

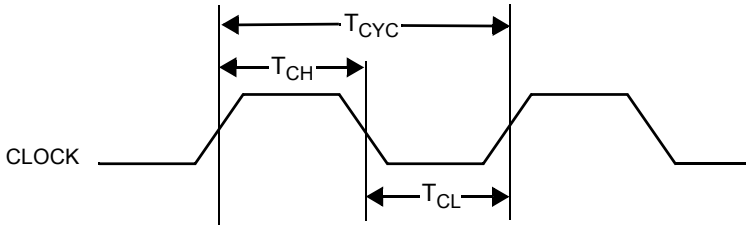


Figure 43. USB Data Signal Timing

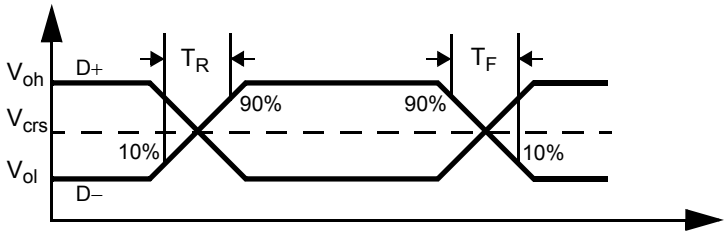


Figure 47. SPI Master Timing, CPHA = 0

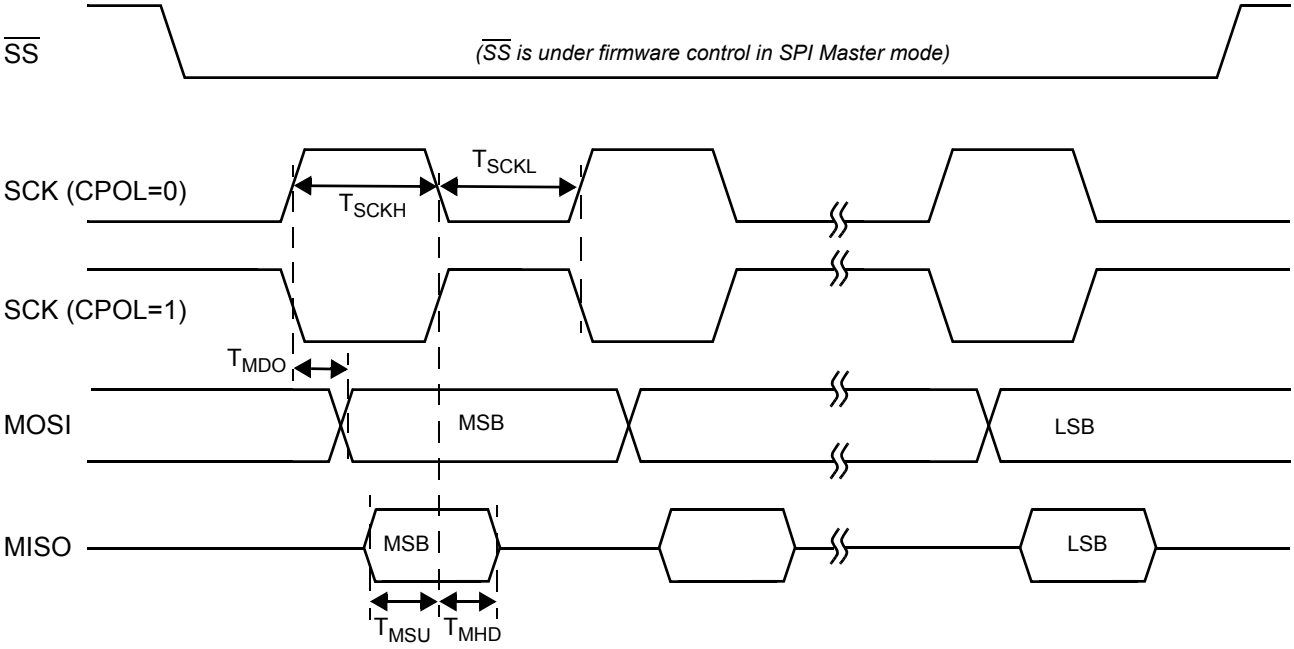


Figure 48. SPI Slave Timing, CPHA = 0

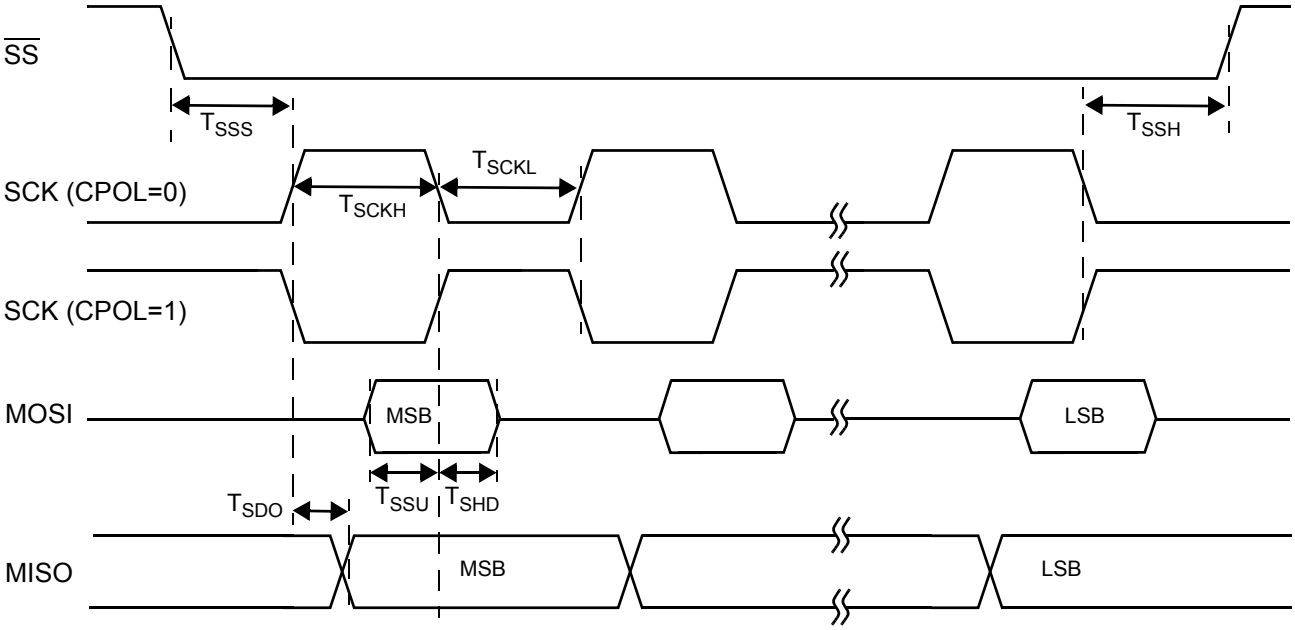


Figure 52. 18-pin SOIC (0.463 × 0.300 × 0.0932 Inches) Package Outline, 51-85023

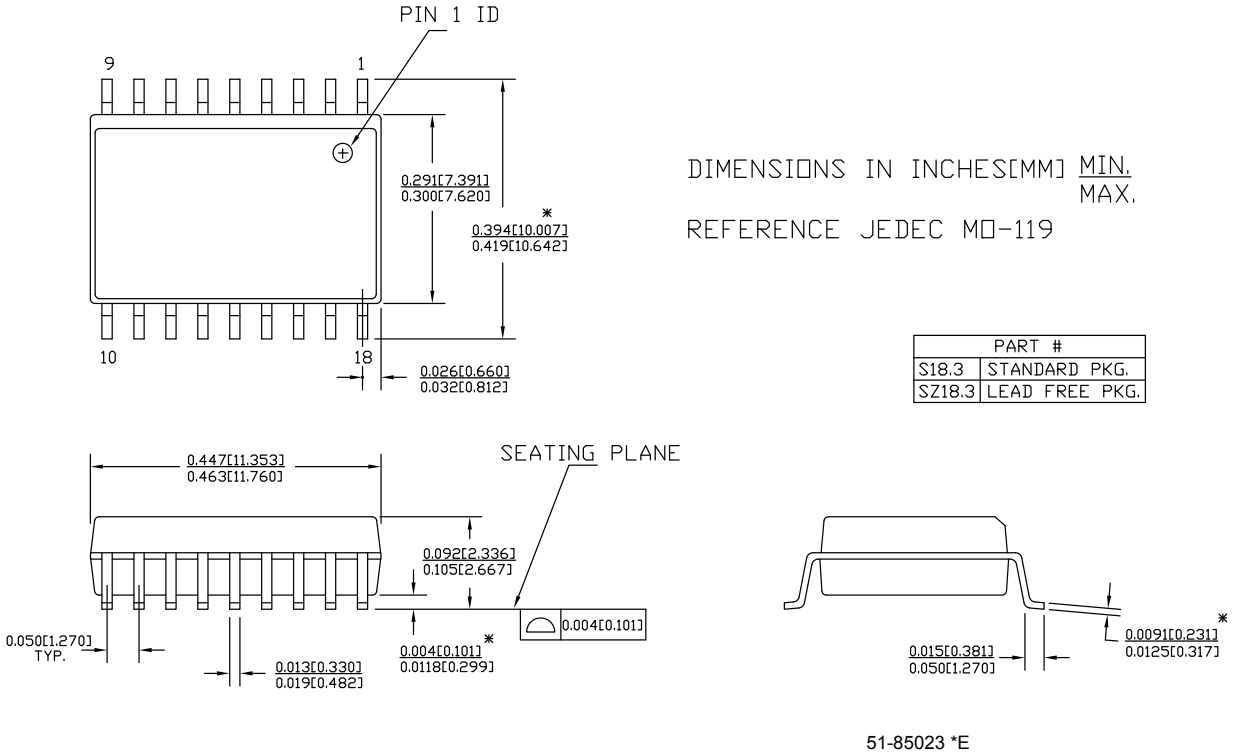


Figure 53. 24-pin SOIC (0.615 × 0.300 × 0.0932 Inches)

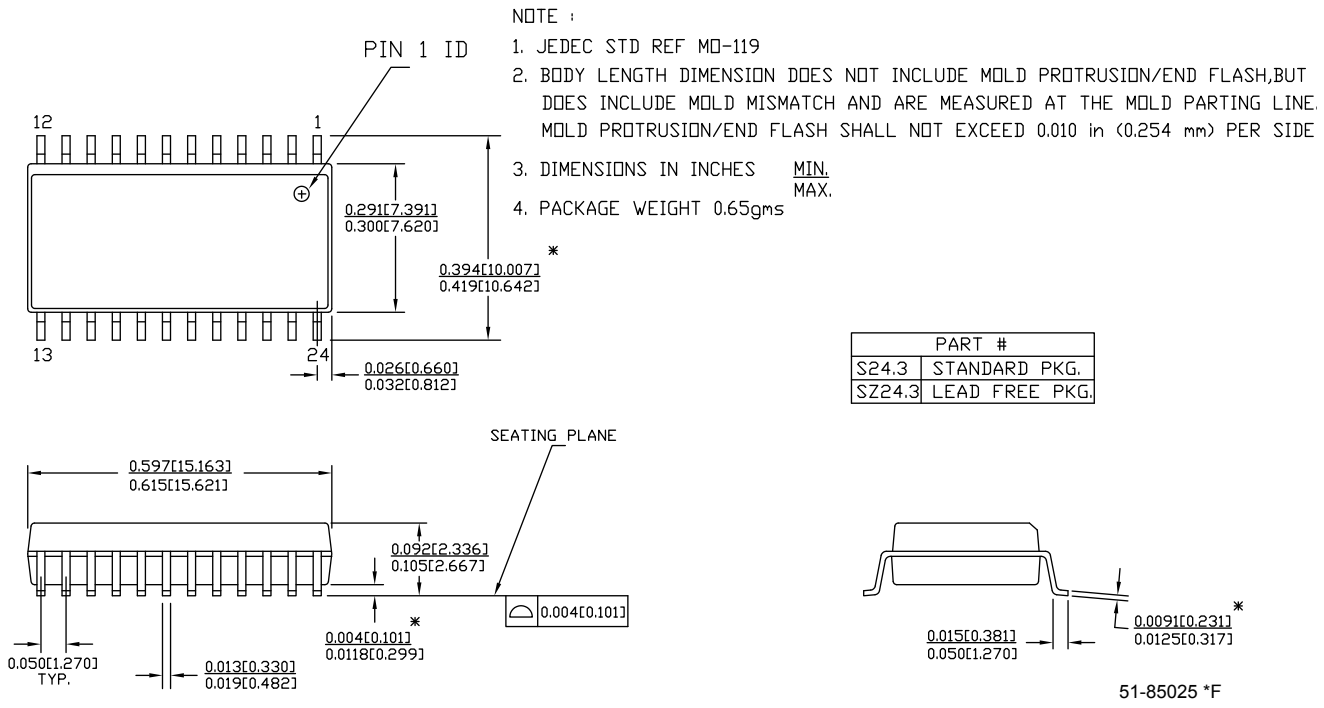


Figure 56. Die Form

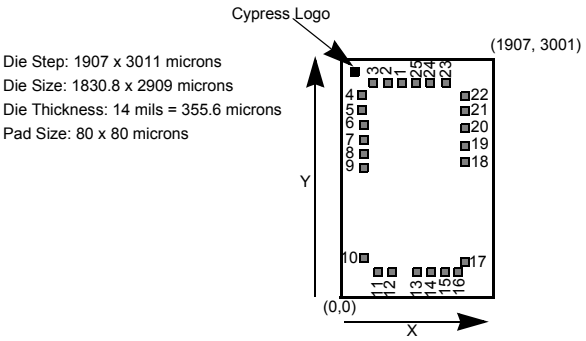


Table 11 below shows the die pad coordinates for the CY7C63722C-XC. The center location of each bond pad is relative to the bottom left corner of the die which has coordinate (0,0).

Table 11. CY7C63722C-XC Probe Pad Coordinates in microns ((0,0) to bond pad centers)

Pad Number	Pin Name	X (microns)	Y (microns)
1	P0.0	788.95	2843.15
2	P0.1	597.45	2843.15
3	P0.2	406.00	2843.15
4	P0.3	154.95	2687.95
5	P1.0	154.95	2496.45
6	P1.2	154.95	2305.05
7	P1.4	154.95	2113.60
8	P1.6	154.95	1922.05
9	Vss	154.95	1730.90
10	Vss	154.95	312.50
11	Vpp	363.90	184.85
12	VREG	531.70	184.85
13	XTALIN	1066.55	184.85
14	XTALOUT	1210.75	184.85
15	Vcc	1449.75	184.85
16	D–	1662.35	184.85
17	D+	1735.35	289.85
18	P1.7	1752.05	1832.75
19	P1.5	1752.05	2024.30
20	P1.3	1752.05	2215.75
21	P1.1	1752.05	2407.15
22	P0.7	1752.05	2598.65
23	P0.6	1393.25	2843.15
24	P0.5	1171.80	2843.15
25	P0.4	980.35	2843.15