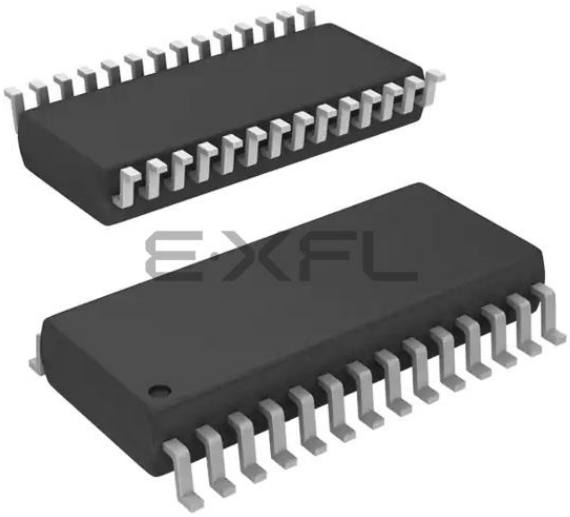


Welcome to E-XFL.COM

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	M8C
Program Memory Type	OTP (8kB)
Controller Series	CY7C640xx
RAM Size	256 x 8
Interface	I²C, USB, HAPI
Number of I/O	19
Voltage - Supply	4V ~ 5.25V
Operating Temperature	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy7c64013-sc

TABLE OF CONTENTS

TABLE OF CONTENTS

2.0 Functional Overview

The CY7C64013 and CY7C64113 are 8-bit One Time Programmable microcontrollers that are designed for full-speed USB applications. The instruction set has been optimized specifically for USB operations, although the microcontrollers can be used for a variety of non-USB embedded applications.

GPIO

The CY7C64013 features 19 GPIO pins to support USB and other applications. The I/O pins are grouped into three ports (P0[7:0], P1[2:0], P2[6:2], P3[2:0]) where each port can be configured as inputs with internal pull-ups, open drain outputs, or traditional CMOS outputs. There are 16 GPIO pins (Ports 0 and 1) which are rated at 7 mA typical sink current. Port 3 pins are rated at 12 mA typical sink current, a current sufficient to drive LEDs. Multiple GPIO pins can be connected together to drive a single output for more drive current capacity. Additionally, each GPIO can be used to generate a GPIO interrupt to the microcontroller. All of the GPIO interrupts share the same “GPIO” interrupt vector.

The CY7C64113 has 32 GPIO pins (P0[7:0], P1[7:0], P2[7:0], P3[7:0])

DAC

The 64113 has four programmable sink current I/O pins (DAC) pins (P4[7,2:0]). Every DAC pin includes an integrated 14-kΩ pull-up resistor. When a ‘1’ is written to a DAC I/O pin, the output current sink is disabled and the output pin is driven HIGH by the internal pull-up resistor. When a ‘0’ is written to a DAC I/O pin, the internal pull-up resistor is disabled and the output pin provides the programmed amount of sink current. A DAC I/O pin can be used as an input with an internal pull-up by writing a ‘1’ to the pin.

The sink current for each DAC I/O pin can be individually programmed to one of 16 values using dedicated Isink registers. DAC bits P4[1:0] can be used as high-current outputs with a programmable sink current range of 3.2 to 16 mA (typical). DAC bits P4[7,2] have a programmable current sink range of 0.2 to 1.0 mA (typical). Multiple DAC pins can be connected together to drive a single output that requires more sink current capacity. Each I/O pin can be used to generate a DAC interrupt to the microcontroller. Also, the interrupt polarity for each DAC I/O pin is individually programmable.

Clock

The microcontroller uses an external 6-MHz crystal and an internal oscillator to provide a reference to an internal PLL-based clock generator. This technology allows the customer application to use an inexpensive 6-MHz fundamental crystal that reduces the clock-related noise emissions (EMI). A PLL clock generator provides the 6-, 12-, and 48-MHz clock signals for distribution within the microcontroller.

Memory

The CY7C64013 and CY7C64113 have 8 KB of PROM.

Power on Reset, Watchdog and Free running Time

These parts include power-on reset logic, a Watchdog timer, and a 12-bit free-running timer. The power-on reset (POR) logic detects when power is applied to the device, resets the logic to a known state, and begins executing instructions at PROM address 0x0000. The Watchdog timer is used to ensure the microcontroller recovers after a period of inactivity. The firmware may become inactive for a variety of reasons, including errors in the code or a hardware failure such as waiting for an interrupt that never occurs.

I2C and HAPI Interface

The microcontroller can communicate with external electronics through the GPIO pins. An I^2C-compatible interface accommodates a 100-kHz serial link with an external device. There is also a Hardware Assisted Parallel Interface (HAPI) which can be used to transfer data to an external device.

Timer

The free-running 12-bit timer clocked at 1 MHz provides two interrupt sources, 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 of the event and after the event is complete. The difference between the two readings indicates the duration of the event in microseconds. The upper four bits of the timer are latched into an internal register when the firmware reads the lower eight bits. A read from the upper four bits actually reads data from the internal register, instead of the timer. This feature eliminates the need for firmware to try to compensate if the upper four bits increment immediately after the lower eight bits are read.

Interrupts

The microcontroller supports 11 maskable interrupts in the vectored interrupt controller. Interrupt sources include the USB Bus Reset interrupt, the 128-µs (bit 6) and 1.024-ms (bit 9) outputs from the free-running timer, five USB endpoints, the DAC port, the GPIO ports, and the I^2C-compatible master mode interface. The timer bits cause an interrupt (if enabled) when the bit toggles from LOW ‘0’ to HIGH ‘1.’ The USB endpoints interrupt after the USB host has written data to the endpoint FIFO or after the USB controller sends a packet to the USB host. The DAC ports have an additional level of masking that allows the user to select which DAC inputs can cause a DAC interrupt. The GPIO ports also 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 pin of the DAC port. Input transition polarity can be programmed for each GPIO port as part of the port configuration. The interrupt polarity can be rising edge (‘0’ to ‘1’) or falling edge (‘1’ to ‘0’).

Table 4-2. I/O Register Summary (continued)

Register Name	I/O Address	Read/Write	Function	Page
GPIO Configuration	0x08	R/W	GPIO Port Configurations	20
HAPI and I^2C Configuration	0x09	R/W	HAPI Width and I^2C Position Configuration	24
USB Device Address A	0x10	R/W	USB Device Address A	34
EP A0 Counter Register	0x11	R/W	USB Address A, Endpoint 0 Counter	35
EP A0 Mode Register	0x12	R/W	USB Address A, Endpoint 0 Configuration	34
EP A1 Counter Register	0x13	R/W	USB Address A, Endpoint 1 Counter	35
EP A1 Mode Register	0x14	R/W	USB Address A, Endpoint 1 Configuration	35
EP A2 Counter Register	0x15	R/W	USB Address A, Endpoint 2 Counter	35
EP A2 Mode Register	0x16	R/W	USB Address A, Endpoint 2 Configuration	35
USB Status & Control	0x1F	R/W	USB Upstream Port Traffic Status and Control	34
Global Interrupt Enable	0x20	R/W	Global Interrupt Enable	29
Endpoint Interrupt Enable	0x21	R/W	USB Endpoint Interrupt Enables	29
Timer (LSB)	0x24	R	Lower 8 Bits of Free-running Timer (1 MHz)	23
Timer (MSB)	0x25	R	Upper 4 Bits of Free-running Timer	24
WDT Clear	0x26	W	Watchdog Timer Clear	18
I^2C Control & Status	0x28	R/W	I^2C Status and Control	25
I^2C Data	0x29	R/W	I^2C Data	25
DAC Data	0x30	R/W	DAC Data	22
DAC Interrupt Enable	0x31	W	Interrupt Enable for each DAC Pin	23
DAC Interrupt Polarity	0x32	W	Interrupt Polarity for each DAC Pin	23
DAC Isink	0x38-0x3F	W	Input Sink Current Control for each DAC Pin	22
Reserved	0x40		Reserved	
EP A3 Counter Register	0x41	R/W	USB Address A, Endpoint 3 Counter	35
EP A3 Mode Register	0x42	R/W	USB Address A, Endpoint 3 Configuration	34
EP A4 Counter Register	0x43	R/W	USB Address A, Endpoint 4 Counter	35
EP A4 Mode Register	0x44	R/W	USB Address A, Endpoint 4 Configuration	35
Reserved	0x48		Reserved	
Reserved	0x49		Reserved	
Reserved	0x4A		Reserved	
Reserved	0x4B		Reserved	
Reserved	0x4C		Reserved	
Reserved	0x4D		Reserved	
Reserved	0x4E		Reserved	
Reserved	0x4F		Reserved	
Reserved	0x50		Reserved	
Reserved	0x51		Reserved	
Processor Status & Control	0xFF	R/W	Microprocessor Status and Control Register	26

5.1.1 Program Memory Organization

after reset	Address	
14-bit PC →	0x0000	Program execution begins here after a reset
	0x0002	USB Bus Reset interrupt vector
	0x0004	128-µs timer interrupt vector
	0x0006	1.024-ms timer interrupt vector
	0x0008	USB address A endpoint 0 interrupt vector
	0x000A	USB address A endpoint 1 interrupt vector
	0x000C	USB address A endpoint 2 interrupt vector
	0x000E	USB address A endpoint 3 interrupt vector
	0x0010	USB address A endpoint 4 interrupt vector
	0x0012	Reserved
	0x0014	DAC interrupt vector
	0x0016	GPIO interrupt vector
	0x0018	I^2C interrupt vector
	0x001A	**Program Memory begins here**
	0x1FDF	**8 KB (-32) PROM ends here (CY7C64013, CY7C64113)**

6.0 Clocking

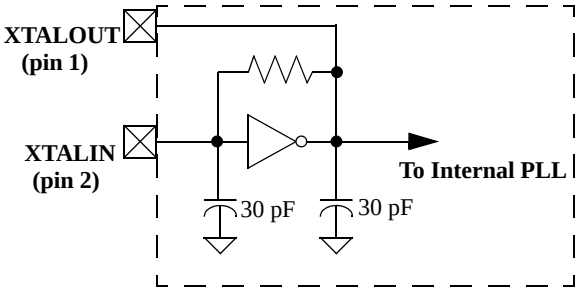


Figure 6-1. Clock Oscillator On-Chip Circuit

The XTALIN and XTALOUT are the clock pins to the microcontroller. The user can connect an external oscillator or a crystal to these pins. When using an external crystal, keep PCB traces between the chip leads and crystal as short as possible (less than 2 cm). A 6-MHz fundamental frequency parallel resonant crystal can be connected to these pins to provide a reference frequency for the internal PLL. The two internal 30-pF load caps appear in series to the external crystal and would be equivalent to a 15-pF load. Therefore, the crystal must have a required load capacitance of about 15–18 pF. A ceramic resonator does not allow the microcontroller to meet the timing specifications of full speed USB and therefore a ceramic resonator is not recommended with these parts.

An external 6-MHz clock can be applied to the XTALIN pin if the XTALOUT pin is left open. Grounding the XTALOUT pin when driving XTALIN with an oscillator does not work because the internal clock is effectively shorted to ground.

7.0 Reset

The CY7C64x13 supports two resets: Power-On Reset (POR) and a Watchdog Reset (WDR). Each of these resets causes:

- all registers to be restored to their default states,
- the USB Device Address to be set to 0,
- all interrupts to be disabled,
- the PSP and Data Stack Pointer (DSP) to be set to memory address 0x00.

The occurrence of a reset is recorded in the Processor Status and Control Register, as described in Section 15.0. Bits 4 and 6 are used to record the occurrence of POR and WDR, respectively. Firmware can interrogate these bits to determine the cause of a reset.

Program execution starts at ROM address 0x0000 after a 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. The firmware reset handler should configure the hardware before the “main” loop of code. Attempting to execute a RET or RETI in the firmware reset handler causes unpredictable execution results.

7.1 Power-On Reset (POR)

When V_{CC} is first applied to the chip, the Power-On Reset (POR) signal is asserted and the CY7C64x13 enters a “semi-suspend” state. During the semi-suspend state, which is different from the suspend state defined in the USB specification, the oscillator and all other blocks of the part are functional, except for the CPU. This semi-suspend time ensures that both a valid V_{CC} level is reached and that the internal PLL has time to stabilize before full operation begins. When the V_{CC} has risen above approximately 2.5V, and the oscillator is stable, the POR is deasserted and the on-chip timer starts counting. The first 1 ms of suspend time is not interruptible, and the semi-suspend state continues for an additional 95 ms unless the count is bypassed by a USB Bus Reset on the upstream port. The 95 ms provides time for V_{CC} to stabilize at a valid operating voltage before the chip executes code.

If a USB Bus Reset occurs on the upstream port during the 95-ms semi-suspend time, the semi-suspend state is aborted and program execution begins immediately from address 0x0000. In this case, the Bus Reset interrupt is pending but not serviced until firmware sets the USB Bus Reset Interrupt Enable bit (bit 0 of register 0x20) and enables interrupts with the EI command.

The POR signal is asserted whenever V_{CC} drops below approximately 2.5V, and remains asserted until V_{CC} rises above this level again. Behavior is the same as described above.

7.2 Watchdog Reset (WDR)

The Watchdog Timer Reset (WDR) occurs when the internal Watchdog timer rolls over. Writing any value to the write-only Watchdog Restart Register at address 0x26 clears the timer. The timer rolls over and WDR occurs if it is not cleared within t_{WATCH} (8 ms minimum) of the last clear. Bit 6 of the Processor Status and Control Register is set to record this event (the register contents are set to 010X0001 by the WDR). A Watchdog Timer Reset lasts for 2 ms, after which the microcontroller begins execution at ROM address 0x0000.

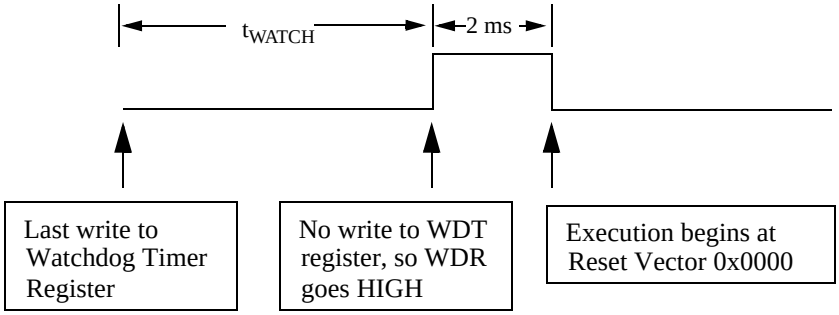


Figure 7-1. Watchdog Reset (WDR)

The USB transmitter is disabled by a Watchdog Reset because the USB Device Address Register is cleared (see Section 18.1). Otherwise, the USB Controller would respond to all address 0 transactions.

It is possible for the WDR bit of the Processor Status and Control Register (0xFF) to be set following a POR event. The WDR bit should be ignored If the firmware interrogates the Processor Status and Control Register for a Set condition on the WDR bit and if the POR (bit 3 of register 0xFF) bit is set.

8.0 Suspend Mode

The CY7C64x13 can be placed into a low-power state by setting the Suspend bit of the Processor Status and Control register. All logic blocks in the device are turned off except the GPIO interrupt logic and the USB receiver. The clock oscillator and PLL, as well as the free-running and Watchdog timers, are shut down. Only the occurrence of an enabled GPIO interrupt or non-idle bus activity at a USB upstream or downstream port wakes the part out of suspend. The Run bit in the Processor Status and Control Register must be set to resume a part out of suspend.

The clock oscillator restarts immediately after exiting suspend mode. The microcontroller returns to a fully functional state 1 ms after the oscillator is stable. The microcontroller executes the instruction following the I/O write that placed the device into suspend mode before servicing any interrupt requests.

The GPIO interrupt allows the controller to wake-up periodically and poll system components while maintaining a very low average power consumption. To achieve the lowest possible current during suspend mode, all I/O should be held at V_{CC} or Gnd. This also applies to internal port pins that may not be bonded in a particular package.

Typical code for entering suspend is shown below:

```
...          ; All GPIO set to low-power state (no floating pins)
...          ; Enable GPIO interrupts if desired for wake-up
mov a, 09h   ; Set suspend and run bits
iowr FFh     ; Write to Status and Control Register - Enter suspend, wait for USB activity (or GPIO Interrupt)
nop          ; This executes before any ISR
...          ; Remaining code for exiting suspend routine
```

Timer MSB **ADDRESS 0x25**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Reserved	Reserved	Reserved	Reserved	Timer Bit 11	Timer Bit 10	Timer Bit 9	Timer Bit 8
Read/Write	-	-	-	-	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Figure 11-2. Timer MSB Register

Bit [3:0]: Timer higher nibble

Bit [7:4]: Reserved

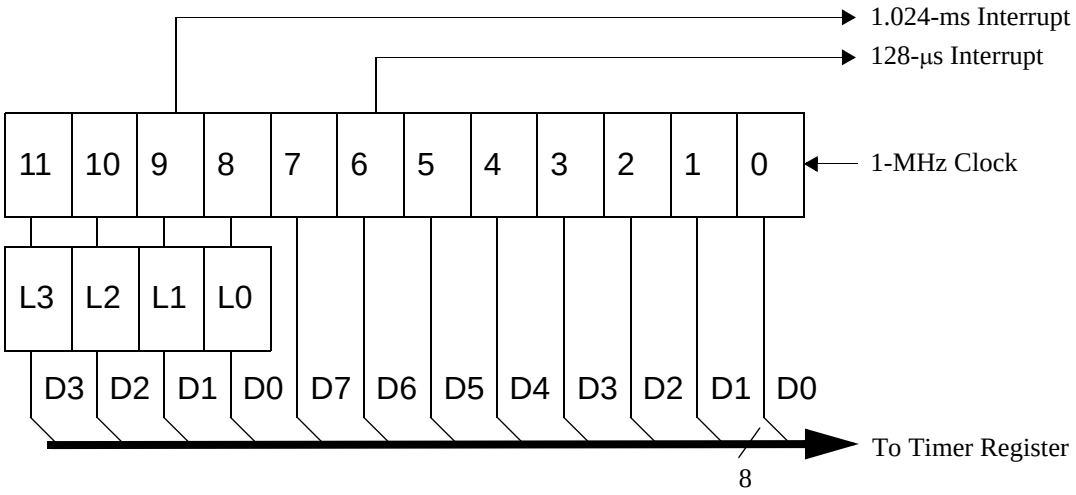


Figure 11-3. Timer Block Diagram

12.0 I²C and HAPI Configuration Register

Internal hardware supports communication with external devices through two interfaces: a two-wire I²C-compatible interface, and a HAPI for 1, 2, or 3 byte transfers. The I²C-compatible interface and HAPI functions, discussed in detail in Sections 13.0 and 14.0, share a common configuration register (see *Figure 12-1*). All bits of this register are cleared on reset.

I²C Configuration **ADDRESS 0x09**

Bit #	7	6	5	4	3	2	1	0
Bit Name	I²C Position	Reserved	LEMPTY Polarity	DRDY Polarity	Latch Empty	Data Ready	HAPI Port Width Bit 1	HAPI Port Width Bit 0
Read/Write	R/W	-	R/W	R/W	R	R	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Figure 12-1. HAPI/I²C Configuration Register

Note: *I²C-compatible function must be separately enabled as described in Section 13.0*

Bits [7,1:0] of the HAPI/I²C Configuration Register control the pin out configuration of the HAPI and I²C-compatible interfaces. Bits [5:2] are used in HAPI mode only, and are described in Section 14.0. *Table 12-1* shows the HAPI port configurations, and *Table 12-2* shows I²C pin location configuration options. These I²C-compatible options exist due to pin limitations in certain packages, and to allow simultaneous HAPI and I²C-compatible operation.

HAPI operation is enabled whenever either HAPI Port Width Bit (Bit 1 or 0) is non-zero. This affects GPIO operation as described in Section 14.0. I²C-compatible blocks must be separately enabled as described in Section 13.0.

The interrupt controller contains a separate flip-flop for each interrupt. See *Figure 16-3* for the logic block diagram of the interrupt controller. When an interrupt is generated, it is first registered as a pending interrupt. It stays pending until it is serviced or a reset occurs. A pending interrupt only generates an interrupt request if it is enabled by the corresponding bit in the interrupt enable registers. The highest priority interrupt request is serviced following the completion of the currently executing instruction.

When servicing an interrupt, the hardware does the following

1. Disables all interrupts by clearing the Global Interrupt Enable bit in the CPU (the state of this bit can be read at Bit 2 of the Processor Status and Control Register, *Figure 15-1*).
2. Clears the flip-flop of the current interrupt.
3. Generates an automatic CALL instruction to the ROM address associated with the interrupt being serviced (i.e., the Interrupt Vector, see Section 16.1).

The instruction in the interrupt table is typically a JMP instruction to the address of the Interrupt Service Routine (ISR). The user can re-enable interrupts in the interrupt service routine by executing an EI instruction. Interrupts can be nested to a level limited only by the available stack space.

The Program Counter value as well as the Carry and Zero flags (CF, ZF) are stored onto the Program Stack by the automatic CALL instruction generated as part of the interrupt acknowledge process. The user firmware is responsible for ensuring that the processor state is preserved and restored during an interrupt. The PUSH A instruction should typically be used as the first command in the ISR to save the accumulator value and the POP A instruction should be used to restore the accumulator value just before the RETI instruction. The program counter CF and ZF are restored and interrupts are enabled when the RETI instruction is executed.

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

16.1 Interrupt Vectors

The Interrupt Vectors supported by the USB Controller are listed in *Table 16-1*. The lowest-numbered interrupt (USB Bus Reset interrupt) has the highest priority, and the highest-numbered interrupt (I^2C interrupt) has the lowest priority.

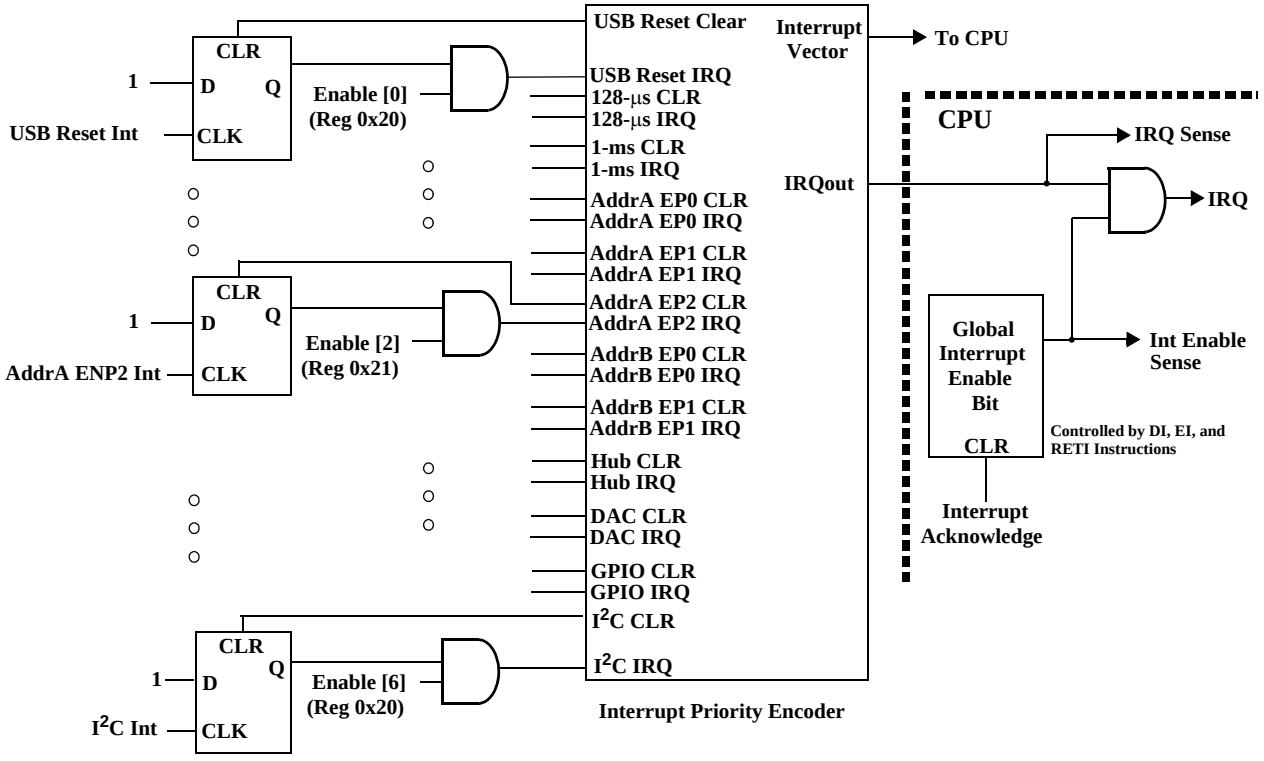


Figure 16-3. Interrupt Controller Function Diagram

Although Reset is not an interrupt, the first instruction executed after a reset is at PROM address 0x0000h—which corresponds to the first entry in the Interrupt Vector Table. Because the JMP instruction is two bytes long, the interrupt vectors occupy two bytes.

Table 16-1. Interrupt Vector Assignments

Interrupt Vector Number	ROM Address	Function
Not Applicable	0x0000	Execution after Reset begins here
1	0x0002	USB Bus Reset interrupt
2	0x0004	128-μs timer interrupt
3	0x0006	1.024-ms timer interrupt
4	0x0008	USB Address A Endpoint 0 interrupt
5	0x000A	USB Address A Endpoint 1 interrupt
6	0x000C	USB Address A Endpoint 2 interrupt
7	0x000E	USB Address A Endpoint 3 interrupt
8	0x0010	USB Address A Endpoint 4 interrupt
9	0x0012	Reserved
10	0x0014	DAC interrupt
11	0x0016	GPIO interrupt
12	0x0018	I^2C interrupt

16.2 Interrupt Latency

Interrupt latency can be calculated from the following equation:

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

For example, if a 5 clock cycle instruction such as JC is being executed when an interrupt occurs, the first instruction of the Interrupt Service Routine executes a minimum of 16 clocks (1+10+5) or a maximum of 20 clocks (5+10+5) after the interrupt is issued. For a 12-MHz internal clock (6-MHz crystal), 20 clock periods is 20 / 12 MHz = 1.667 μs.

16.3 USB Bus Reset Interrupt

The USB Controller recognizes a USB Reset when a Single Ended Zero (SE0) condition persists on the upstream USB port for 12–16 μs (the Reset may be recognized for an SE0 as short as 12 μs, but is always recognized for an SE0 longer than 16 μs). SE0 is defined as the condition in which both the D+ line and the D– line are LOW. Bit 5 of the Status and Control Register is set to record this event. The interrupt is asserted at the end of the Bus Reset. If the USB reset occurs during the start-up delay following a POR, the delay is aborted as described in Section 7.1. The USB Bus Reset Interrupt is generated when the SE0 state is deasserted.

A USB Bus Reset clears the following registers:

SIE Section:USB Device Address Registers (0x10, 0x40)

16.4 Timer Interrupt

There are two periodic timer interrupts: the 128-μs interrupt and the 1.024-ms interrupt. The user should disable both timer interrupts before going into the suspend mode to avoid possible conflicts between servicing the timer interrupts first or the suspend request first.

16.5 USB Endpoint Interrupts

There are five USB endpoint interrupts, one per endpoint. A USB endpoint interrupt is generated after the USB host writes to a USB endpoint FIFO or after the USB controller sends a packet to the USB host. The interrupt is generated on the last packet of the transaction (e.g., on the host's ACK during an IN, or on the device ACK during on OUT). If no ACK is received during an IN transaction, no interrupt is generated.

16.6 DAC Interrupt

Each DAC I/O pin can generate an interrupt, if enabled. The interrupt polarity for each DAC I/O pin is programmable. A positive polarity is a rising edge input while a negative polarity is a falling edge input. All of the DAC pins share a single interrupt vector, which means the firmware needs to read the DAC port to determine which pin or pins caused an interrupt.

If one DAC pin has triggered an interrupt, no other DAC pins can cause a DAC interrupt until that pin has returned to its inactive (non-trigger) state or the corresponding interrupt enable bit is cleared. The USB Controller does not assign interrupt priority to different DAC pins and the DAC Interrupt Enable Register is not cleared during the interrupt acknowledge process.

USB Status and Control **ADDRESS 0x1F**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Endpoint Size	Endpoint Mode	D+ Upstream	D– Upstream	Bus Activity	Control Action Bit 2	Control Action Bit 1	Control Action Bit 0
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Figure 17-1. USB Status and Control Register

Bits[2..0] : Control Action

Set to control action as per *Table 17-1*.The three control bits allow the upstream port to be driven manually by firmware. For normal USB operation, all of these bits must be cleared. *Table 17-1* shows how the control bits affect the upstream port.

Table 17-1. Control Bit Definition for Upstream Port

Control Bits	Control Action
000	Not Forcing (SIE Controls Driver)
001	Force D+[0] HIGH, D–[0] LOW
010	Force D+[0] LOW, D–[0] HIGH
011	Force SE0; D+[0] LOW, D–[0] LOW
100	Force D+[0] LOW, D–[0] LOW
101	Force D+[0] HiZ, D–[0] LOW
110	Force D+[0] LOW, D–[0] HiZ
111	Force D+[0] HiZ, D–[0] HiZ

Bit 3 : Bus Activity

This is a “sticky” bit that indicates if any non-idle USB event has occurred on the upstream USB port. Firmware should check and clear this bit periodically to detect any loss of bus activity. Writing a ‘0’ to the Bus Activity bit clears it, while writing a ‘1’ preserves the current value. In other words, the firmware can clear the Bus Activity bit, but only the SIE can set it.

Bits 4 and 5 : D– Upstream and D+ Upstream

These bits give the state of each upstream port pin individually: 1 = HIGH, 0 = LOW.

Bit 6 : Endpoint Mode

This bit used to configure the number of USB endpoints. See *Section 18.2* for a detailed description.

Bit 7 : Endpoint Size

This bit used to configure the number of USB endpoints. See *Section 18.2* for a detailed description.

18.0 USB Serial Interface Engine Operation

USB Device Address A includes up to five endpoints: EPA0, EPA1, EPA2, EPA3, and EPA4. Endpoint (EPA0) allows the USB host to recognize, set-up, and control the device. In particular, EPA0 is used to receive and transmit control (including set-up) packets.

18.1 USB Device Address

The USB Controller provides one USB Device Address with five endpoints. The USB Device Address Register contents are cleared during a reset, setting the USB device address to zero and marking this address as disabled. *Figure 18-1* shows the format of the USB Address Registers.

USB Device Address **ADDRESSES 0x10**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Device Address Enable	Device Address Bit 6	Device Address Bit 5	Device Address Bit 4	Device Address Bit 3	Device Address Bit 2	Device Address Bit 1	Device Address Bit 0
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 18-1. USB Device Address Registers

Bits[6..0] :Device Address

Firmware writes this bits during the USB enumeration process to the non-zero address assigned by the USB host.

Bit 7 :Device Address Enable

Must be set by firmware before the SIE can respond to USB traffic to the Device Address.

Bit 7 (Device Address Enable) in the USB Device Address Register must be set by firmware before the SIE can respond to USB traffic to this address. The Device Addresses in bits [6:0] are set by firmware during the USB enumeration process to the non-zero address assigned by the USB host.

18.2 USB Device Endpoints

The CY7C64x13 controller supports one USB device address and five endpoints for communication with the host. The configuration of these endpoints, and associated FIFOs, is controlled by bits [7,6] of the USB Status and Control Register (0x1F). Bit 7 controls the size of the endpoints and bit 6 controls the number of endpoints. These configuration options are detailed in *Table 18-1*. The “unused” FIFO areas in the following table can be used by the firmware as additional user RAM space.

Table 18-1. Memory Allocation for Endpoints

USB Status And Control Register (0x1F) Bits [7, 6]											
[0,0]			[1,0]			[0,1]			[1,1]		
Label	Start Address	Size	Label	Start Address	Size	Label	Start Address	Size	Label	Start Address	Size
unused	0xD8	8	unused	0xA8	8	EPA4	0xD8	8	EPA4	0xB0	8
unused	0xE0	8	unused	0xB0	8	EPA3	0xE0	8	EPA3	0xA8	8
EPA2	0xE8	8	EPA0	0xB8	8	EPA2	0xE8	8	EPA0	0xB8	8
EPA1	0xF0	8	EPA1	0xC0	32	EPA1	0xF0	8	EPA1	0xC0	32
EPA0	0xF8	8	EPA2	0xE0	32	EPA0	0xF8	8	EPA2	0xE0	32

When the SIE writes data to a FIFO, the internal data bus is driven by the SIE; not the CPU. This causes a short delay in the CPU operation. The delay is three clock cycles per byte. For example, an 8-byte data write by the SIE to the FIFO generates a delay of 2 μs (3 cycles/byte * 83.33 ns/cycle * 8 bytes).

18.3 USB Control Endpoint Mode Register

All USB devices are required to have a Control Endpoint 0 (EPA0) that is used to initialize and control each USB address. Endpoint 0 provides access to the device configuration information and allows generic USB status and control accesses. Endpoint 0 is bidirectional to both receive and transmit data. The other endpoints are unidirectional, but selectable by the user as IN or OUT endpoints.

The endpoint mode register is cleared during reset. The endpoint zero EPA0 mode register uses the format shown in *Figure 18-2*.

USB Device Endpoint Zero Mode **ADDRESSES 0x12)**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Endpoint 0 SETUP Received	Endpoint 0 IN Received	Endpoint 0 OUT Received	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0
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 18-2. USB Device Endpoint Zero Mode Registers

Bits[3..0] : Mode

These sets the mode which control how the control endpoint responds to traffic.

Bit 4 : ACK

This bit is set whenever the SIE engages in a transaction to the register’s endpoint that completes with an ACK packet.

Bit 5: Endpoint 0 OUT Received

1= Token received is an OUT token. 0= Token received is not an OUT token. This bit is set by the SIE to report the type of token received by the corresponding device address is an OUT token. The bit must be cleared by firmware as part of the USB processing.

Bit 6: Endpoint 0 IN Received

1= Token received is an IN token. 0= Token received is not an IN token. This bit is set by the SIE to report the type of token received by the corresponding device address is an IN token. The bit must be cleared by firmware as part of the USB processing.

This bit is set on receiving a proper CRC when the endpoint FIFO buffer is loaded with data during transactions. This bit is used OUT and SETUP tokens only. If the CRC is not correct, the endpoint interrupt occurs, but Data Valid is cleared to a zero.

Bit 7 : Data 0/1 Toggle

This bit selects the DATA packet's toggle state: 0 for DATA0, 1 for DATA1. For IN transactions, firmware must set this bit to the desired state. For OUT or SETUP transactions, the hardware sets this bit to the state of the received Data Toggle bit.

Whenever the count updates from a SETUP or OUT transaction on endpoint 0, the counter 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. Only endpoint 0 counter register is locked when updated. The locking mechanism does not apply to the count registers of other endpoints.

18.6 Endpoint Mode/Count Registers Update and Locking Mechanism

The contents of the endpoint mode and counter registers are updated, based on the packet flow diagram in *Figure 18-5*. Two time points, UPDATE and SETUP, are shown in the same figure. The following activities occur at each time point:

SETUP:

The SETUP bit of the endpoint 0 mode register is forced HIGH at this time. This bit is forced HIGH by the SIE until the end of the data phase of a control write transfer. The SETUP bit can not be cleared by firmware during this time.

The affected mode and counter registers of endpoint 0 are locked from any CPU writes once they are updated. These registers can be unlocked by a CPU read, only if the read operation occurs after the UPDATE. The firmware needs to perform a register read as a part of the endpoint ISR processing to unlock the effected registers. The locking mechanism on mode and counter registers ensures that the firmware recognizes the changes that the SIE might have made since the previous IO read of that register.

UPDATE:

1. Endpoint Mode Register – All the bits are updated (except the SETUP bit of the endpoint 0 mode register).
2. Counter Registers – All bits are updated.
3. Interrupt – If an interrupt is to be generated as a result of the transaction, the interrupt flag for the corresponding endpoint is set at this time. For details on what conditions are required to generate an endpoint interrupt, refer to *Table 19-2*.
4. The contents of the updated endpoint 0 mode and counter registers are locked, except the SETUP bit of the endpoint 0 mode register which was locked earlier.

Table 19-2. Details of Modes for Differing Traffic Conditions (see *Table 19-1* for the decode legend) (continued)

1	1	1	0	Out	2	UC	valid	0	1	updates	UC	UC	1	UC	0	0	1	1	Stall	yes
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/extra In																				
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	1	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																				
Properties of Incoming Packet								**Changes made by SIE to Internal Registers and Mode Bits**												
Mode Bits				**token**	**count**	**buffer**	**dval**	**DTOG**	**DVAL**	**COUNT**	**Setup**	**In**	**Out**	**ACK**	**Mode Bits**				**Response**	**Intr**
Normal Out/erroneous In																				
1	0	0	1	Out	<= 10	data	valid	updates	1	updates	UC	UC	UC	1	1	0	0	0	ACK	yes
1	0	0	1	Out	> 10	junk	x	updates	updates	updates	UC	UC	UC	UC	NoChange				ignore	yes
1	0	0	1	Out	x	junk	invalid	updates	0	updates	UC	UC	UC	UC	NoChange				ignore	yes
1	0	0	1	In	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
																			(STALL[3] = 0)	
1	0	0	1	In	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				Stall	no
																			(STALL[3] = 1)	
NAK Out/erroneous In																				
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
Isochronous endpoint (Out)																				
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																				
Properties of Incoming Packet								**Changes made by SIE to Internal Registers and Mode Bits**												
Mode Bits				**token**	**count**	**buffer**	**dval**	**DTOG**	**DVAL**	**COUNT**	**Setup**	**In**	**Out**	**ACK**	**Mode Bits**				**Response**	**Intr**
Normal In/erroneous Out																				
1	1	0	1	Out	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
																			(STALL[3] = 0)	
1	1	0	1	Out	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				stall	no
																			(STALL[3] = 1)	
1	1	0	1	In	x	UC	x	UC	UC	UC	UC	1	UC	1	1	1	0	0	ACK (back)	yes
NAK In/erroneous Out																				
1	1	0	0	Out	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
1	1	0	0	In	x	UC	x	UC	UC	UC	UC	1	UC	UC	NoChange				NAK	yes
Isochronous endpoint (In)																				
0	1	1	1	Out	x	UC	x	UC	UC	UC	UC	UC	UC	UC	NoChange				ignore	no
0	1	1	1	In	x	UC	x	UC	UC	UC	UC	1	UC	UC	NoChange				TX	yes

Note:
3. STALL bit is bit 7 of the USB Non-Control Device Endpoint Mode registers. For more information, refer to Sec.

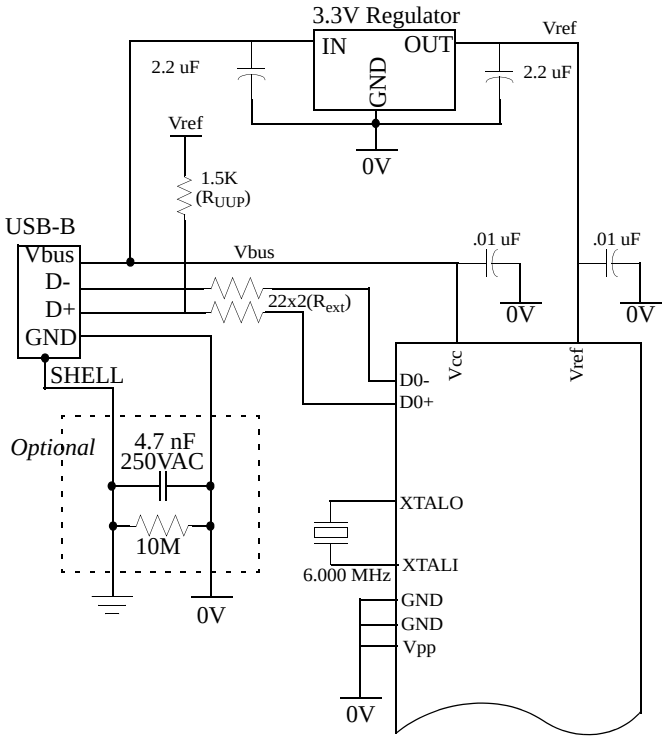
20.0 Register Summary

	Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Read/Write/ Both/-	Default/ Reset
GPIO CONFIGURATION PORTS 0, 1, 2 AND 3	0x00	Port 0 Data	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	BBBBBBBB	11111111
	0x01	Port 1 Data	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	BBBBBBBB	11111111
	0x02	Port 2 Data	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	BBBBBBBB	11111111
	0x03	Port 3 Data	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	BBBBBBBB	11111111
	0x04	Port 0 Interrupt Enable	P0.7 Intr Enable	P0.6 Intr Enable	P0.5 Intr Enable	P0.4 Intr Enable	P0.3 Intr Enable	P0.2 Intr Enable	P0.1 Intr Enable	P0.0 Intr Enable	WWWWWWWW	00000000
	0x05	Port 1 Interrupt Enable	P1.7 Intr Enable	P1.6 Intr Enable	P1.5 Intr Enable	P1.4 Intr Enable	Reserved	P1.2 Intr Enable	P1.1 Intr Enable	P1.0 Intr Enable	WWWWWWWW	00000000
	0x06	Port 2 Interrupt Enable	P2.7 Intr Enable	P2.6 Intr Enable	P2.5 Intr Enable	P2.4 Intr Enable	P2.3 Intr Enable	Reserved	Reserved	Reserved	WWWWWWWW	00000000
	0x07	Port 3 Interrupt Enable	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	P3.1 Intr Enable	P3.0 Intr Enable	WWWWWWWW	00000000
	0x08	GPIO Configuration	Port 3 Config Bit 1	Port 3 Config Bit 0	Port 2 Config Bit 1	Port 2 Config Bit 0	Port 1 Config Bit 1	Port 1 Config Bit 0	Port 0 Config Bit 1	Port 0 Config Bit 0	BBBBBBBB	00000000
HAPI I²C	0x09	HAPI/I²C Configuration	I²C Position	Reserved	Reserved	Reserved	Reserved	Reserved	I²C Port Width	Reserved	BBBBBBBB	00000000
ENDPOINT A0, AI AND A2 CONFIGURATION	0x10	USB Device Address A	Device Address A Enable	Device Address A Bit 6	Device Address A Bit 5	Device Address A Bit 4	Device Address A Bit 3	Device Address A Bit 2	Device Address A Bit 1	Device Address A Bit 0	BBBBBBBB	00000000
	0x11	EP A0 Counter Register	Data 0/1 Toggle	Data Valid	Byte Count Bit 5	Byte Count Bit 4	Byte Count Bit 3	Byte Count Bit 2	Byte Count Bit 1	Byte Count Bit 0	BBBBBBBB	00000000
	0x12	EP A0 Mode Register	Endpoint0 SETUP Received	Endpoint0 IN Received	Endpoint0 OUT Received	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0	BBBBBBBB	00000000
	0x13	EP A1 Counter Register	Data 0/1 Toggle	Data Valid	Byte Count Bit 5	Byte Count Bit 4	Byte Count Bit 3	Byte Count Bit 2	Byte Count Bit 1	Byte Count Bit 0	BBBBBBBB	00000000
	0x14	EP A1 Mode Register	STALL	-	-	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0	BBBBBBBB	00000000
	0x15	EP A2 Counter Register	Data 0/1 Toggle	Data Valid	Byte Count Bit 5	Byte Count Bit 4	Byte Count Bit 3	Byte Count Bit 2	Byte Count Bit 1	Byte Count Bit 0	BBBBBBBB	00000000
	0x16	EP A2 Mode Register	STALL	-	-	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0	BBBBBBBB	00000000
USB CS	0x1F	USB Status and Control	Endpoint Size	Endpoint Mode	D+ Upstream	D– Upstream	Bus Activity	Control Bit 2	Control Bit 1	Control Bit 0	BBRRBBBB	-0xx0000
INTERRUPT	0x20	Global Interrupt Enable	Reserved	I²C Interrupt Enable	GPIO Interrupt Enable	Reserved	USB Hub Interrupt Enable	1.024-ms Interrupt Enable	128-µs Interrupt Enable	USB Bus RESET Interrupt Enable	-BBBBBBB	-0000000
	0x21	Endpoint Interrupt Enable	Reserved	Reserved	Reserved	EPB1 Interrupt Enable	EPB0 Interrupt Enable	EPA2 Interrupt Enable	EPA1 Interrupt Enable	EPA0 Interrupt Enable	---BBBBB	---00000
TIMER	0x24	Timer (LSB)	Timer Bit 7	Timer Bit 6	Timer Bit 5	Timer Bit 4	Timer Bit 3	Timer Bit 2	Timer Bit 1	Timer Bit 0	RRRRRRRR	00000000
	0x25	Timer (MSB)	Reserved	Reserved	Reserved	Reserved	Timer Bit 11	Timer Bit 10	Time Bit 9	Timer Bit 8	----rrrr	----0000
	0x26	WDT Clear	x	x6	x	x	x 3	x2	x	x	WWWWWWWW	xxxxxxxx
I²C	0x28	I²C Control and Status	MSTR Mode	Continue/ Busy	Xmit Mode	ACK	Addr	ARB Lost/ Restart	Received Stop	I²C Enable	BBBBBBBB	00000000
	0x29	I²C Data	I²C Data 7	I²C Data 6	I²C Data 5	I²C Data 4	I²C Data 3	I²C Data 2	I²C Data 1	I²C Data 0	BBBBBBBB	XXXXXXXX
DAC PORT	0x30	DAC Data	Timer Bit 7	Timer Bit 6	Timer Bit 5	Timer Bit 4	Timer Bit 3	Timer Bit 2	Timer Bit 1	Timer Bit 0	RRRRRRRR	00000000
	0x31	DAC Interrupt Enable)	Reserved	Reserved	Reserved	Reserved	Timer Bit 11	Timer Bit 10	Time Bit 9	Timer Bit 8	----rrrr	----0000
	0x32	DAC Interrupt Polarity										
	0x38- 0x3F	DAS Isink	x	x6	x	x	x 3	x2	x	x	WWWWWWWW	
ENDPOINT A3, A4 CONFIGURATION	0x40	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	BBBBBBBB	00000000
	0x41	EP A3 Counter Register	Data 0/1 Toggle	Data Valid	Byte Count Bit 5	Byte Count Bit 4	Byte Count Bit 3	Byte Count Bit 2	Byte Count Bit 1	Byte Count Bit 0	BBBBBBBB	00000000
	0x42	EP A3 Mode Register	Endpoint 0 SETUP Received	Endpoint 0 IN Received	Endpoint 0 OUT Received	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0	BBBBBBBB	00000000
	0x43	EP A4 Counter Register	Data 0/1 Toggle	Data Valid	Byte Count Bit 5	Byte Count Bit 4	Byte Count Bit 3	Byte Count Bit 2	Byte Count Bit 1	Byte Count Bit 0	BBBBBBBB	00000000
	0x44	EP A4 Mode Register	STALL	-	-	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0	BBBBBBBB	00000000

	Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Read/Write/ Both/-	Default/ Reset
RESERVED	0x48	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x49	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x4A	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x4B	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x4C	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	--000000
	0x4D	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x4E	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x4F	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x50	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x51	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0x52	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	00000000
	0xFF	Process Status & Control	IRQ Pending	Watchdog Reset	USB Bus Reset Interrupt	Power-On Reset	Suspend	Interrupt Enable Sense	Reserved	Run	RBBBBRBB	00010001

Note:
B: Read and Write
W: Write
R: Read

21.0 Sample Schematic



22.0 Absolute Maximum Ratings

Storage Temperature .....–65°C to +150°C
Ambient Temperature with Power Applied.....0°C to +70°C
Supply voltage on V_{CC} relative to V_{SS} ..... –0.5V to +7.0V
DC Input Voltage..... –0.5V to +V_{CC}+0.5V
DC Voltage Applied to Outputs in High Z State –0.5V to +V_{CC}+0.5V
Power Dissipation .....500 mW
Static Discharge Voltage .....>2000V
Latch-up Current ..... >200 mA
Max Output Sink Current into Port 0, 1, 2, 3, and DAC[1:0] Pins 60 mA
Max Output Sink Current into DAC[7:2] Pins..... 10 mA

23.0 Electrical Characteristics

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

Parameter	Description	Conditions	Min.	Max.	Unit
	General				
V_{REF}	Reference Voltage	3.3V ±5%	3.15	3.45	V
V_{pp}	Programming Voltage (disabled)		–0.4	0.4	V
I_{CC}	V_{CC} Operating Current	No GPIO source current		50	mA
I_{SB1}	Supply Current—Suspend Mode			50	μA
I_{ref}	V_{REF} Operating Current	Note 5		30	mA
I_{il}	Input Leakage Current	Any pin		1	μA
	USB Interface				
V_{di}	Differential Input Sensitivity	| (D+)–(D–) |	0.2		V
V_{cm}	Differential Input Common Mode Range		0.8	2.5	V
V_{se}	Single Ended Receiver Threshold		0.8	2.0	V
C_{in}	Transceiver Capacitance			20	pF
I_{lo}	Hi-Z State Data Line Leakage	0 V < V_{in} < 3.3 V	–10	10	μA
R_{ext}	External USB Series Resistor	In series with each USB pin	19	21	Ω
R_{UUP}	External Upstream USB Pull-up Resistor	1.5 kΩ ±5%, D+ to V_{REG}	1.425	1.575	kΩ
	Power On Reset				
t_{vccs}	V_{CC} Ramp Rate	Linear ramp 0V to V_{CC}[4]	0	100	ms
	USB Upstream				
V_{UOH}	Static Output High	15 kΩ ±5% to Gnd	2.8	3.6	V
V_{UOL}	Static Output Low	1.5 kΩ ±5% to V_{REF}		0.3	V
Z_O	USB Driver Output Impedance	Including R_{ext} Resistor	28	44	Ω
	General Purpose I/O (GPIO)				
R_{up}	Pull-up Resistance (typical 14 kΩ)		8.0	24.0	kΩ
V_{ITH}	Input Threshold Voltage	All ports, LOW to HIGH edge	20%	40%	V_{CC}
V_H	Input Hysteresis Voltage	All ports, HIGH to LOW edge	2%	8%	V_{CC}
V_{OL}	Port 0,1,2,3 Output Low Voltage	I_{OL} = 3 mA I_{OL} = 8 mA		0.4 2.0	V V
V_{OH}	Output High Voltage	I_{OH} = 1.9 mA (all ports 0,1,2,3)	2.4		V

Notes:
4. Power-on Reset occurs whenever the voltage on V_{CC} is below approximately 2.5V.
5. This is based on transitions every 2 full-speed bit times on average.

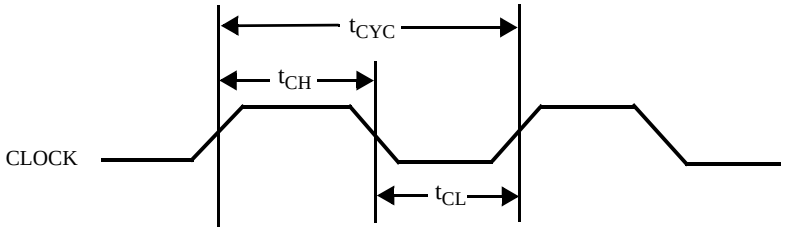


Figure 24-1. Clock Timing

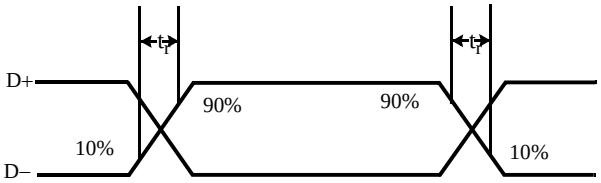


Figure 24-2. USB Data Signal Timing

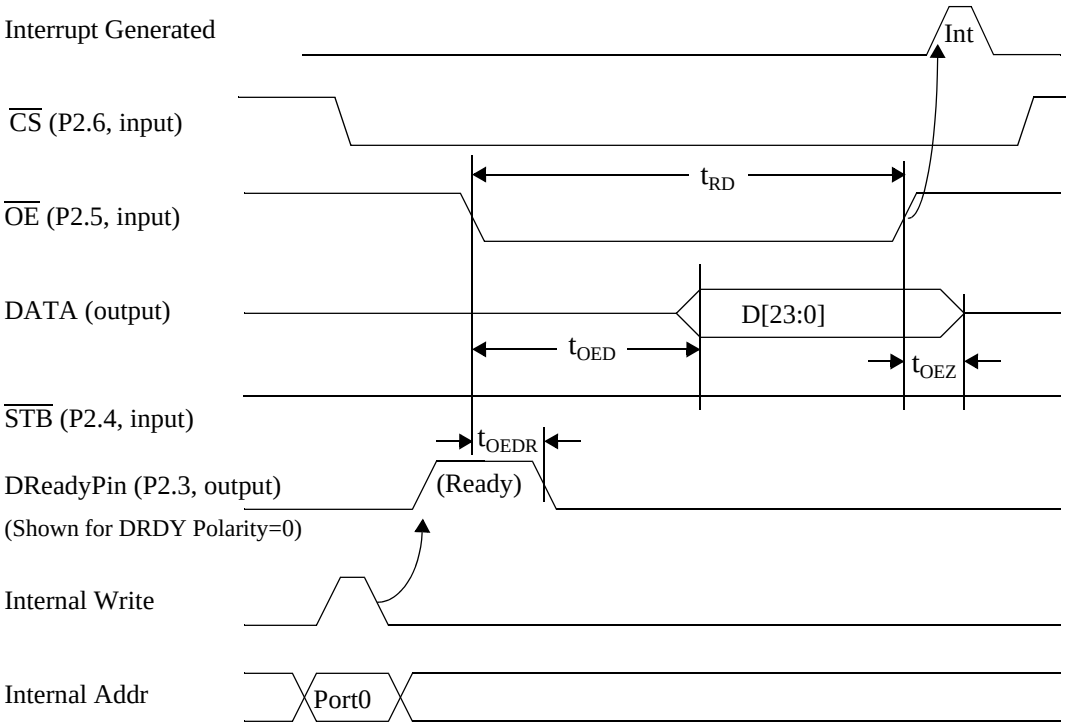
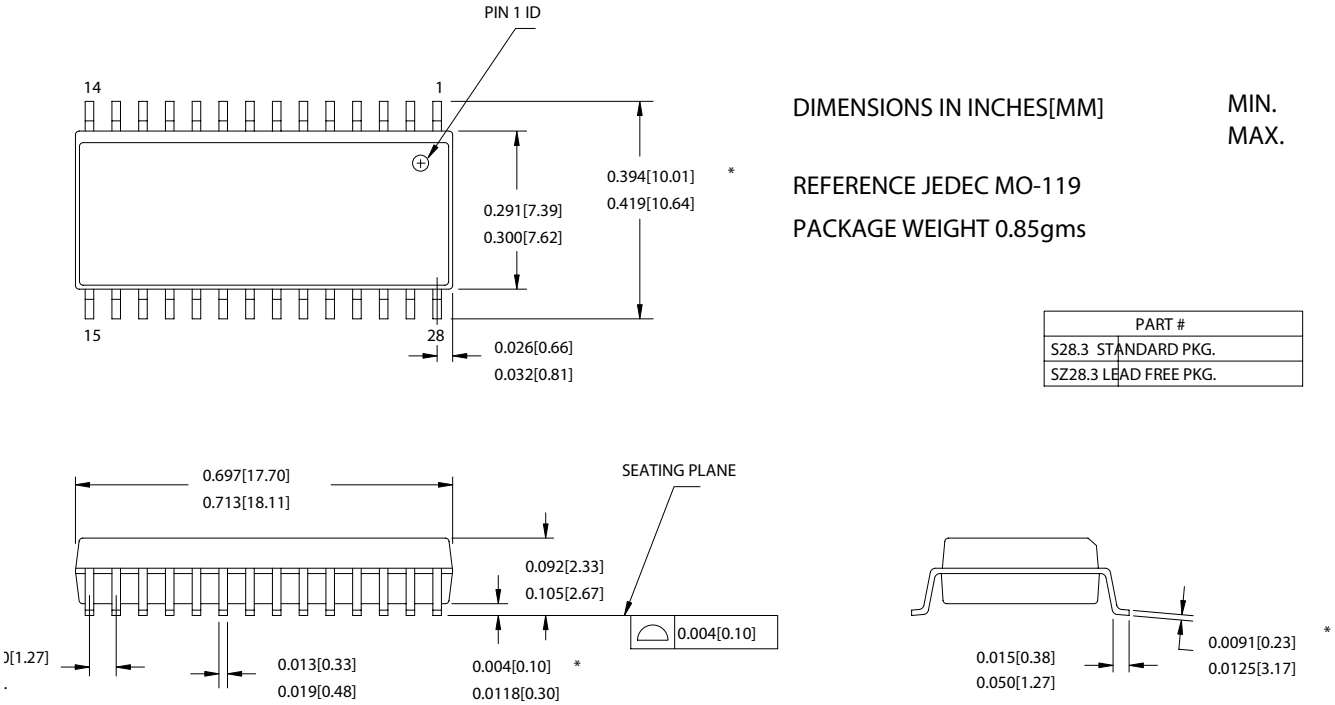


Figure 24-3. HAPI Read by External Interface from USB Microcontroller

26.0 Package Diagrams (continued)

28-Lead (300-Mil) Molded SOIC S21



All product and company names mentioned in this document are the trademarks of their respective holders.

Document History Page

Document Title: CY7C64013, CY7C64113 Full-Speed USB (12 Mbps) Function Document Number: 38-08001				
REV.	**ECN NO.**	**Issue Date**	**Orig. of Change**	**Description of Change**
**	109962	12/16/01	SZV	Change from Spec number: 38-00626 to 38-08001
*A	129715	02/05/04	MON	Added register bit definitions Added default bit state of each register Corrected the Schematic (location of the Pull up on D+) Added register summary Modified tables 19-1 and 19-2 Provided more explanation regarding locking/unlocking mechanism of the mode register.