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	CY7C641xx
RAM Size	256 x 8
Interface	I²C, USB, HAPI
Number of I/O	36
Voltage - Supply	4V ~ 5.25V
Operating Temperature	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	48-BSSOP (0.295", 7.50mm Width)
Supplier Device Package	48-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy7c64113-pvc

TABLE OF CONTENTS

LIST OF FIGURES

1.0 Features

- Full-speed USB Microcontroller
- 8-bit USB Optimized Microcontroller
  - Harvard architecture
  - 6-MHz external clock source
  - 12-MHz internal CPU clock
  - 48-MHz internal clock
- Internal memory
  - 256 bytes of RAM
  - 8 KB of PROM (CY7C64013, CY7C64113)
- Integrated Master/Slave I^2C-compatible Controller (100 kHz) enabled through General-Purpose I/O (GPIO) pins
- Hardware Assisted Parallel Interface (HAPI) for data transfer to external devices
- I/O ports
  - Three GPIO ports (Port 0 to 2) capable of sinking 7 mA per pin (typical)
  - An additional GPIO port (Port 3) capable of sinking 12 mA per pin (typical) for high current requirements: LEDs
  - Higher current drive achievable by connecting multiple GPIO pins together to drive a common output
  - Each GPIO port can be configured as inputs with internal pull-ups or open drain outputs or traditional CMOS outputs
  - A Digital to Analog Conversion (DAC) port with programmable current sink outputs is available on the CY7C64113 devices
  - Maskable interrupts on all I/O pins
- 12-bit free-running timer with one microsecond clock ticks
- Watchdog Timer (WDT)
- Internal Power-On Reset (POR)
- USB Specification Compliance
  - Conforms to USB Specification, Version 1.1
  - Conforms to USB HID Specification, Version 1.1
  - Supports up to five user configured endpoints
    - Up to four 8-byte data endpoints
    - Up to two 32-byte data endpoints
  - Integrated USB transceivers
- Improved output drivers to reduce EMI
- Operating voltage from 4.0V to 5.5V DC
- Operating temperature from 0 to 70 degrees Celsius
  - CY7C64013 available in 28-pin SOIC and 28-pin PDIP packages
  - CY7C64113 available in 48-pin SSOP packages
- Industry-standard programmer support

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)**

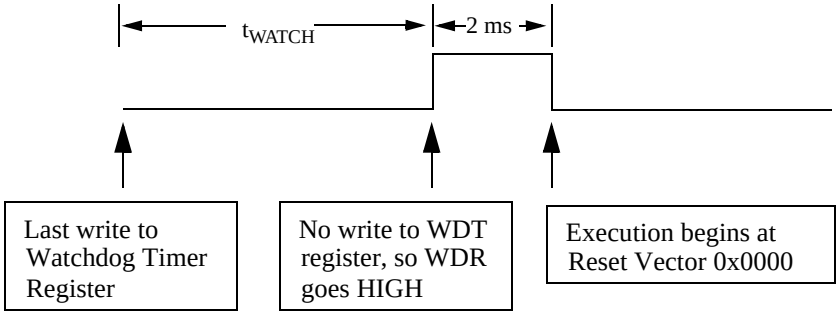


Figure 7-1. Watchdog Reset (WDR)

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
```

Port 3 Data **ADDRESS 0x03**

Bit #	7	6	5	4	3	2	1	0
Bit Name	P3.7	P3.6	P3.5	P3.4	P3.3	P32	P3.1	P3.0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Figure 9-5. Port 3 Data

Special care should be taken with any unused GPIO data bits. An unused GPIO data bit, either a pin on the chip or a port bit that is not bonded on a particular package, must not be left floating when the device enters the suspend state. If a GPIO data bit is left floating, the leakage current caused by the floating bit may violate the suspend current limitation specified by the USB Specifications. If a '1' is written to the unused data bit and the port is configured with open drain outputs, the unused data bit remains in an indeterminate state. Therefore, if an unused port bit is programmed in open-drain mode, it must be written with a '0.' Notice that the CY7C64013 part always requires that the data bits P1[7:3], P2[7,1,0], and P3[7:3] be written with a '0.'

In normal non-HAPI mode, reads from a GPIO port always return the present state of the voltage at the pin, independent of the settings in the Port Data Registers. If HAPI mode is activated for a port, reads of that port return latched data as controlled by the HAPI signals (see Section 14.0). During reset, all of the GPIO pins are set to a high-impedance input state ('1' in open drain mode). Writing a '0' to a GPIO pin drives the pin LOW. In this state, a '0' is always read on that GPIO pin unless an external source overdrives the internal pull-down device.

9.1 GPIO Configuration Port

Every GPIO port can be programmed as inputs with internal pull-ups, outputs LOW or HIGH, or Hi-Z (floating, the pin is not driven internally). In addition, the interrupt polarity for each port can be programmed. The Port Configuration bits (*Figure 9-6*) and the Interrupt Enable bit (*Figure 9-7* through *Figure 9-10*) determine the interrupt polarity of the port pins.

GPIO Configuration **ADDRESS 0x08**

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

Figure 9-6. GPIO Configuration Register

As shown in *Table 9-1* below, a positive polarity on an input pin represents a rising edge interrupt (LOW to HIGH), and a negative polarity on an input pin represents a falling edge interrupt (HIGH to LOW).

The GPIO interrupt is generated when all of the following conditions are met: the Interrupt Enable bit of the associated Port Interrupt Enable Register is enabled, the GPIO Interrupt Enable bit of the Global Interrupt Enable Register (*Figure 16-1*) is enabled, the Interrupt Enable Sense (bit 2, *Figure 15-1*) is set, and the GPIO pin of the port sees an event matching the interrupt polarity.

The driving state of each GPIO pin is determined by the value written to the pin's Data Register (*Figure 9-2* through *Figure 9-5*) and by its associated Port Configuration bits as shown in the GPIO Configuration Register (*Figure 9-6*). These ports are configured on a per-port basis, so all pins in a given port are configured together. The possible port configurations are detailed in *Table 9-1*. As shown in this table below, when a GPIO port is configured with CMOS outputs, interrupts from that port are disabled.

During reset, all of the bits in the GPIO Configuration Register are written with '0' to select Hi-Z mode for all GPIO ports as the default configuration.

Table 9-1. GPIO Port Output Control Truth Table and Interrupt Polarity

Port Config Bit 1	Port Config Bit 0	Data Register	Output Drive Strength	Interrupt Enable Bit	Interrupt Polarity
1	1	0	Output LOW	0	Disabled
		1	Resistive	1	– (Falling Edge)
1	0	0	Output LOW	0	Disabled
		1	Output HIGH	1	Disabled
0	1	0	Output LOW	0	Disabled
		1	Hi-Z	1	– (Falling Edge)
0	0	0	Output LOW	0	Disabled
		1	Hi-Z	1	+ (Rising Edge)

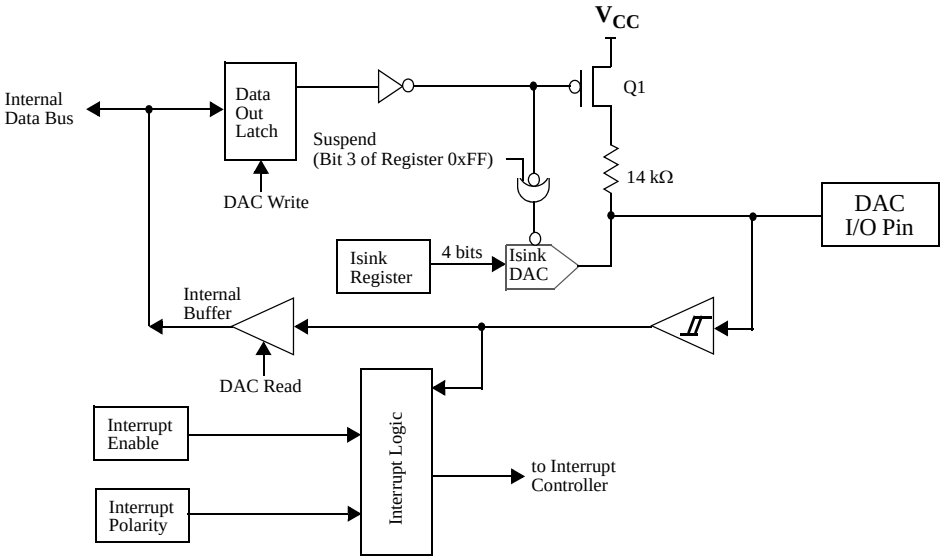


Figure 10-1. Block Diagram of a DAC Pin

The amount of sink current for the DAC I/O pin is programmable over 16 values based on the contents of the DAC Isink Register for that output pin. DAC[1:0] are high-current outputs that are programmable from 3.2 mA to 16 mA (typical). DAC[7:2] are low-current outputs, programmable from 0.2 mA to 1.0 mA (typical).

When the suspend bit in Processor Status and Control Register (see *Figure 15-1*) is set, the Isink DAC block of the DAC circuitry is disabled. Special care should be taken when the CY7C64x13 device is placed in the suspend mode. The DAC Port Data Register (see *Figure 10-2*) should normally be loaded with all '1's (0xFF) before setting the suspend bit. If any of the DAC bits are set to '0' when the device is suspended, that DAC input will float. The floating pin could result in excessive current consumption by the device, unless an external load places the pin in a deterministic state.

DAC Port Data **ADDRESS 0x30**

Bit #	7	6	5	4	3	2	1	0
Bit Name	DAC[7]	Reserved	Reserved	Reserved	Reserved	DAC[2]	DAC[1]	DAC[0]
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

Figure 10-2. DAC Port Data

Bit [1..0]: High Current Output 3.2 mA to 16 mA typical

1= I/O pin is an output pulled HGH through the 14-kΩ resistor. 0 = I/O pin is an input with an internal 14-kΩ pull-up resistor

Bit [3..2]: Low Current Output 0.2 mA to 1 mA typical

1= I/O pin is an output pulled HGH through the 14-kΩ resistor. 0 = I/O pin is an input with an internal 14-kΩ pull-up resistor

10.1 DAC Isink Registers

Each DAC I/O pin has an associated DAC Isink register to program the output sink current when the output is driven LOW. The first Isink register (0x38) controls the current for DAC[0], the second (0x39) for DAC[1], and so on until the Isink register at 0x3F controls the current to DAC[7].

DAC Sink Register **ADDRESS 0x38 -0x3F**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Reserved	Reserved	Reserved	Reserved	Isink[3]	Isink[2]	Isink[1]	Isink[0]
Read/Write					W	W	W	W
Reset	-	-	-	-	0	0	0	0

Figure 10-3. DAC Sink Register

Bit [4..0]: Isink [x] (x= 0..4)

Writing all ‘0’s to the Isink register causes 1/5 of the max current to flow through the DAC I/O pin. Writing all ‘1’s to the Isink register provides the maximum current flow through the pin. The other 14 states of the DAC sink current are evenly spaced between these two values.

Bit [7..5]: Reserved

10.2 DAC Port Interrupts

A DAC port interrupt can be enabled/disabled for each pin individually. The DAC Port Interrupt Enable register provides this feature with an interrupt enable bit for each DAC I/O pin.All of the DAC Port Interrupt Enable register bits are cleared to ‘0’ during a reset. All DAC pins share a common interrupt, as explained in Section 16.6.

DAC Port Interrupt **ADDRESS 0x31**

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

Figure 10-4. DAC Port Interrupt Enable

Bit [7..0]: Enable bit x (x= 0..2, 7)

1= Enables interrupts from the corresponding bit position; 0= Disables interrupts from the corresponding bit position

As an additional benefit, the interrupt polarity for each DAC pin is programmable with the DAC Port Interrupt Polarity register. Writing a ‘0’ to a bit selects negative polarity (falling edge) that causes an interrupt (if enabled) if a falling edge transition occurs on the corresponding input pin. Writing a ‘1’ to a bit in this register selects positive polarity (rising edge) that causes an interrupt (if enabled) if a rising edge transition occurs on the corresponding input pin. All of the DAC Port Interrupt Polarity register bits are cleared during a reset.

DAC Port Interrupt Polarity **ADDRESS 0x32**

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

Figure 10-5. DAC Port Interrupt Polarity

Bit [7..0]: Enable bit x (x= 0..2, 7)

1= Selects positive polarity (rising edge) that causes an interrupt (if enabled);

0= Selects negative polarity (falling edge) that causes an interrupt (if enabled)

11.0 12-Bit Free-Running Timer

The 12-bit timer provides two interrupts (128-µs and 1.024-ms) and allows the firmware to directly time events that are up to 4 ms in duration. The lower 8 bits of the timer can be read directly by the firmware. Reading the lower 8 bits latches the upper 4 bits into a temporary register. When the firmware reads the upper 4 bits of the timer, it is accessing the count stored in the temporary register. The effect of this logic is to ensure a stable 12-bit timer value can be read, even when the two reads are separated in time.

Timer LSB **ADDRESS 0x24**

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

Figure 11-1. Timer LSB Register

Bit [7:0]: Timer lower 8 bits

Bit 1 : Receive Stop

This bit is set when the slave is in receive mode and detects a stop bit on the bus. The Receive Stop bit is not set if the firmware terminates the I^2C transaction by not acknowledging the previous byte transmitted on the I^2C-compatible bus, e.g. in receive mode if firmware sets the Continue bit and clears the ACK bit.

Bit 0 : I^2C Enable

Set this bit to override GPIO definition with I^2C-compatible function on the two I^2C-compatible pins. When this bit is cleared, these pins are free to function as GPIOs. In I^2C-compatible mode, the two pins operate in open drain mode, independent of the GPIO configuration setting.

14.0 Hardware Assisted Parallel Interface (HAPI)

The CY7C64x13 processor provides a hardware assisted parallel interface for bus widths of 8, 16, or 24 bits, to accommodate data transfer with an external microcontroller or similar device. Control bits for selecting the byte width are in the HAPI/I^2C Configuration Register (*Figure 12-1*), bits 1 and 0.

Signals are provided on Port 2 to control the HAPI interface. *Table 14-1* describes these signals and the HAPI control bits in the HAPI/I^2C Configuration Register. Enabling HAPI causes the GPIO setting in the GPIO Configuration Register (0x08) to be overridden. The Port 2 output pins are in CMOS output mode and Port 2 input pins are in input mode (open drain mode with Q3 OFF in *Figure 9-1*).

Table 14-1. Port 2 Pin and HAPI Configuration Bit Definitions

Pin	Name	Direction	Description (Port 2 Pin)
P2[2]	LatEmptyPin	Out	Ready for more input data from external interface.
P2[3]	DReadyPin	Out	Output data ready for external interface.
P2[4]	$\overline{STB}$	In	Strobe signal for latching incoming data.
P2[5]	$\overline{OE}$	In	Output Enable, causes chip to output data.
P2[6]	CS	In	Chip Select (Gates $\overline{STB}$ and $\overline{OE}$).
Bit	**Name**	**R/W**	**Description (HAPI/I^2C Configuration Register)**
2	Data Ready	R	Asserted after firmware writes data to Port 0, until $\overline{OE}$ driven LOW.
3	Latch Empty	R	Asserted after firmware reads data from Port 0, until $\overline{STB}$ driven LOW.
4	DRDY Polarity	R/W	Determines polarity of Data Ready bit and DReadyPin: If 0, Data Ready is active LOW, DReadyPin is active HIGH. If 1, Data Ready is active HIGH, DReadyPin is active LOW.
5	LEMPTY Polarity	R/W	Determines polarity of Latch Empty bit and LatEmptyPin: If 0, Latch Empty is active LOW, LatEmptyPin is active HIGH. If 1, Latch Empty is active HIGH, LatEmptyPin is active LOW.

HAPI Read by External Device from CY7C64x13:

In this case (see *Figure 24-3*), firmware writes data to the GPIO ports. If 16-bit or 24-bit transfers are being made, Port 0 should be written last, since writes to Port 0 asserts the Data Ready bit and the DReady Pin to signal the external device that data is available.

The external device then drives the $\overline{OE}$ and $\overline{CS}$ pins active (LOW), which causes the HAPI data to be output on the port pins. When $\overline{OE}$ is returned HIGH (inactive), the HAPI/GPIO interrupt is generated. At that point, firmware can reload the HAPI latches for the next output, again writing Port 0 last.

The Data Ready bit reads the opposite state from the external DReadyPin on pin P2[3]. If the DRDY Polarity bit is 0, DReadyPin is active HIGH, and the Data Ready bit is active LOW.

HAPI Write by External Device to CY7C64x13:

In this case (see *Figure 24-4*), the external device drives the $\overline{STB}$ and $\overline{CS}$ pins active (LOW) when it drives new data onto the port pins. When this happens, the internal latches become full, which causes the Latch Empty bit to be deasserted. When $\overline{STB}$ is returned HIGH (inactive), the HAPI/GPIO interrupt is generated. Firmware then reads the parallel ports to empty the HAPI latches. If 16-bit or 24-bit transfers are being made, Port 0 should be read last because reads from Port 0 assert the Latch Empty bit and the LatEmptyPin to signal the external device for more data.

The Latch Empty bit reads the opposite state from the external LatEmptyPin on pin P2[2]. If the LEMPTY Polarity bit is 0, LatEmptyPin is active HIGH, and the Latch Empty bit is active LOW.

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.

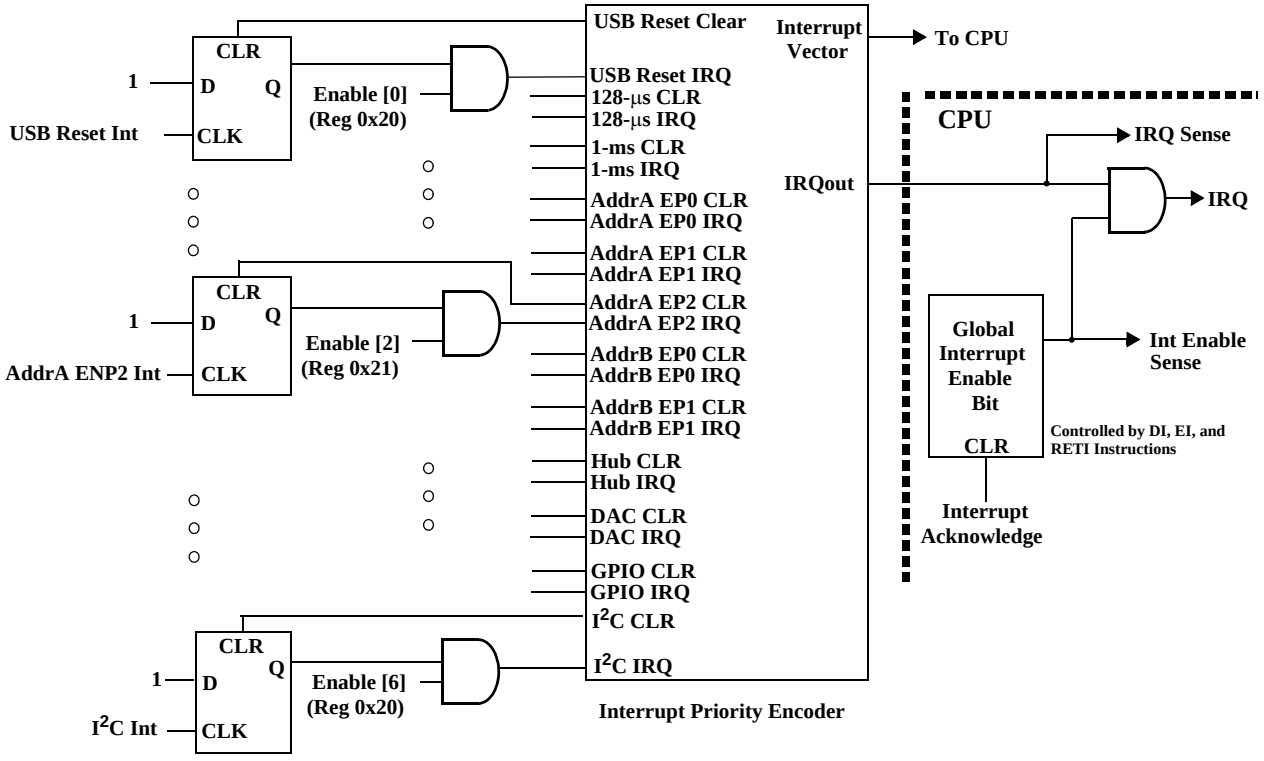


Figure 16-3. Interrupt Controller Function Diagram

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.

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.

19.0 USB Mode Tables

Table 19-1. USB Register Mode Encoding

Mode	Mode Bits	SETUP	IN	OUT	Comments
Disable	0000	ignore	ignore	ignore	Ignore all USB traffic to this endpoint
Nak In/Out	0001	accept	NAK	NAK	Forced from Setup on Control endpoint, from modes other than 0000
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
Isochronous Out	0101	ignore	ignore	always	For Isochronous endpoints
Status In Only	0110	accept	TX 0 BYte	stall	For Control Endpoints
Isochronous In	0111	ignore	TX Count	ignore	For Isochronous endpoints
Nak Out	1000	ignore	ignore	NAK	Is set by SIE on an ACK from mode 1001 (Ack Out)
Ack Out(STALL[3]=0) Ack Out(STALL[3]=1)	1001 1001	ignore ignore	ignore ignore	ACK stall	On issuance of an ACK this mode is changed by SIE to 1000 (NAK Out)
Nak Out - Status In	1010	accept	TX 0 BYte	NAK	Is set by SIE on an ACK from mode 1011 (Ack Out- Status In)
Ack Out - Status In	1011	accept	TX 0 BYte	ACK	On issuance of an ACK this mode is changed by SIE to 1010 (NAK Out - Status In)
Nak In	1100	ignore	NAK	ignore	Is set by SIE on an ACK from mode 1101 (Ack In)
Ack IN(STALL[3]=0) Ack IN(STALL[3]=1)	1101 1101	ignore ignore	TX Count stall	ignore ignore	On issuance of an ACK this mode is changed by SIE to 1100 (NAK In)
Nak In - Status Out	1110	accept	NAK	check	Is set by SIE on an ACK from mode 1111 (Ack In - Status Out)
Ack In - Status Out	1111	accept	TX Count	check	On issuance of an ACK this mode is changed by SIE to 1110 (NAK In - Status Out)

Mode

This lists the mnemonic given to the different modes that can be set in the Endpoint Mode Register by writing to the lower nibble (bits 0..3). The bit settings for different modes are covered in the column marked “Mode Bits”. The Status IN and Status OUT represent the Status stage in the IN or OUT transfer involving the control endpoint.

Mode Bits

These column lists the encoding for different modes by setting Bits[3..0] of the Endpoint Mode register. This modes represents how the SIE responds to different tokens sent by the host to an endpoint. For instance, if the mode bits are set to “0001” (NAK IN/OUT), the SIE will respond with an

- ACK on receiving a SETUP token from the host
- NAK on receiving an OUT token from the host
- NAK on receiving an IN token from the host

Refer to section 13.0 for more information on the SIE functioning

SETUP, IN and OUT

These columns shows the SIE’s response to the host on receiving a SETUP, IN and OUT token depending on the mode set in the Endpoint Mode Register.

A “Check” on the OUT token column, implies that on receiving an OUT token the SIE checks to see whether the OUT packet is of zero length and has a Data Toggle (DTOG) set to ‘1.’ If the DTOG bit is set and the received OUT Packet has zero length, the OUT is ACKed to complete the transaction. If either of this condition is not met the SIE will respond with a STALLL or just ignore the transaction.

A “TX Count” entry in the IN column implies that the SIE transmit the number of bytes specified in the Byte Count (bits 3..0 of the Endpoint Count Register) to the host in response to the IN token received.

A “TX0 Byte” entry in the IN column implies that the SIE transmit a zero length byte packet in response to the IN token received from the host.

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.

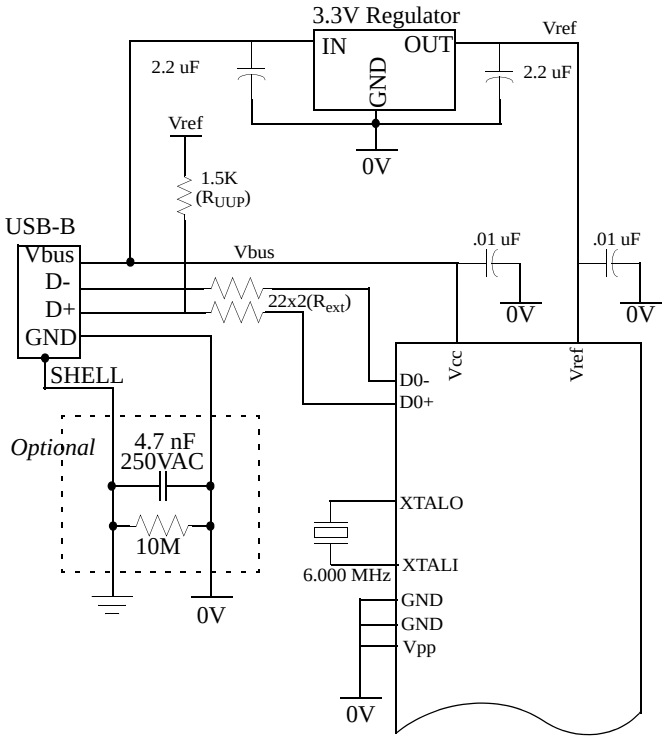
20.0 Register Summary

	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



23.0 Electrical Characteristics (continued)

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

Parameter	Description	Conditions	Min.	Max.	Unit
	DAC Interface				
R_{up}	DAC Pull-up Resistance (typical 14 kΩ)		8.0	24.0	kΩ
$I_{sink0(0)}$	DAC[7:2] Sink current (0)	V_{out} = 2.0V DC	0.1	0.3	mA
$I_{sink0(F)}$	DAC[7:2] Sink current (F)	V_{out} = 2.0V DC	0.5	1.5	mA
$I_{sink1(0)}$	DAC[1:0] Sink current (0)	V_{out} = 2.0V DC	1.6	4.8	mA
$I_{sink1(F)}$	DAC[1:0] Sink current (F)	V_{out} = 2.0V DC	8	24	mA
I_{range}	Programmed Isink Ratio: max/min	V_{out} = 2.0V DC[6]	4	6	
T_{ratio}	Tracking Ratio DAC[1:0] to DAC[7:2]	V_{out} = 2.0V[7]	14	22	
$I_{sinkDAC}$	DAC Sink Current	V_{out} = 2.0V DC	1.6	4.8	mA
I_{lin}	Differential Nonlinearity	DAC Port[8]		0.6	LSB

24.0 Switching Characteristics (f_{OSC} = 6.0 MHz)

Parameter	Description	Min.	Max.	Unit
	Clock Source			
f_{OSC}	Clock Rate	6 ±0.25%		MHz
t_{cyc}	Clock Period	166.25	167.08	ns
t_{CH}	Clock HIGH time	0.45 t_{CYC}		ns
t_{CL}	Clock LOW time	0.45 t_{CYC}		ns
	USB Full Speed Signaling[9]			
t_{rfs}	Transition Rise Time	4	20	ns
t_{ffs}	Transition Fall Time	4	20	ns
t_{rfmfs}	Rise / Fall Time Matching; (t_r/t_f)	90	111	%
$t_{dratefs}$	Full Speed Date Rate	12 ±0.25%		Mb/s
	DAC Interface			
t_{sink}	Current Sink Response Time		0.8	µs
	HAPI Read Cycle Timing			
t_{RD}	Read Pulse Width	15		ns
t_{OED}	OE LOW to Data Valid[10, 11]		40	ns
t_{OEZ}	OE HIGH to Data High-Z[11]		20	ns
t_{OEDR}	OE LOW to Data_Ready Deasserted[10, 11]	0	60	ns
	HAPI Write Cycle Timing			
t_{WR}	Write Strobe Width	15		ns
t_{DSTB}	Data Valid to STB HIGH (Data Set-up Time)[11]	5		ns
t_{STBZ}	STB HIGH to Data High-Z (Data Hold Time)[11]	15		ns
t_{STBLE}	STB LOW to Latch_Empty Deasserted[10, 11]	0	50	ns
	Timer Signals			
t_{watch}	Watchdog Timer Period	8.192	14.336	ms

Notes:

6. Irange: $I_{sinkn}(15)/ I_{sinkn}(0)$ for the same pin.
7. $T_{ratio} = I_{sink1}[1:0](n)/I_{sink0}[7:2](n)$ for the same n, programmed.
8. I_{lin} measured as largest step size vs. nominal according to measured full scale and zero programmed values.
9. Per Table 7-6 of revision 1.1 of USB specification.
10. For 25-pF load.
11. Assumes chip select $\overline{CS}$ is asserted (LOW).

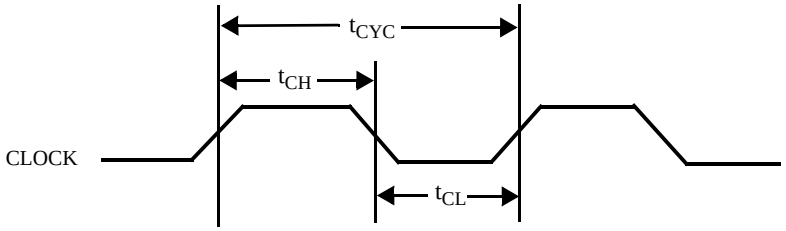


Figure 24-1. Clock Timing

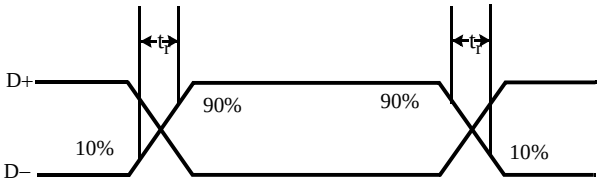


Figure 24-2. USB Data Signal Timing

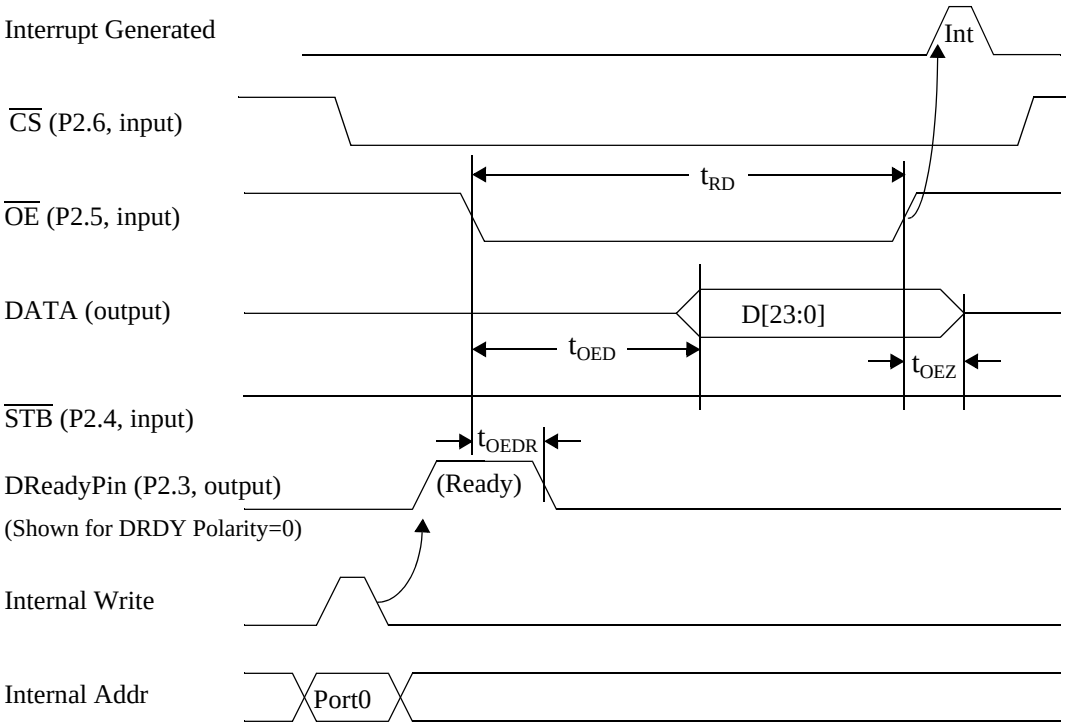


Figure 24-3. HAPI Read by External Interface from USB Microcontroller

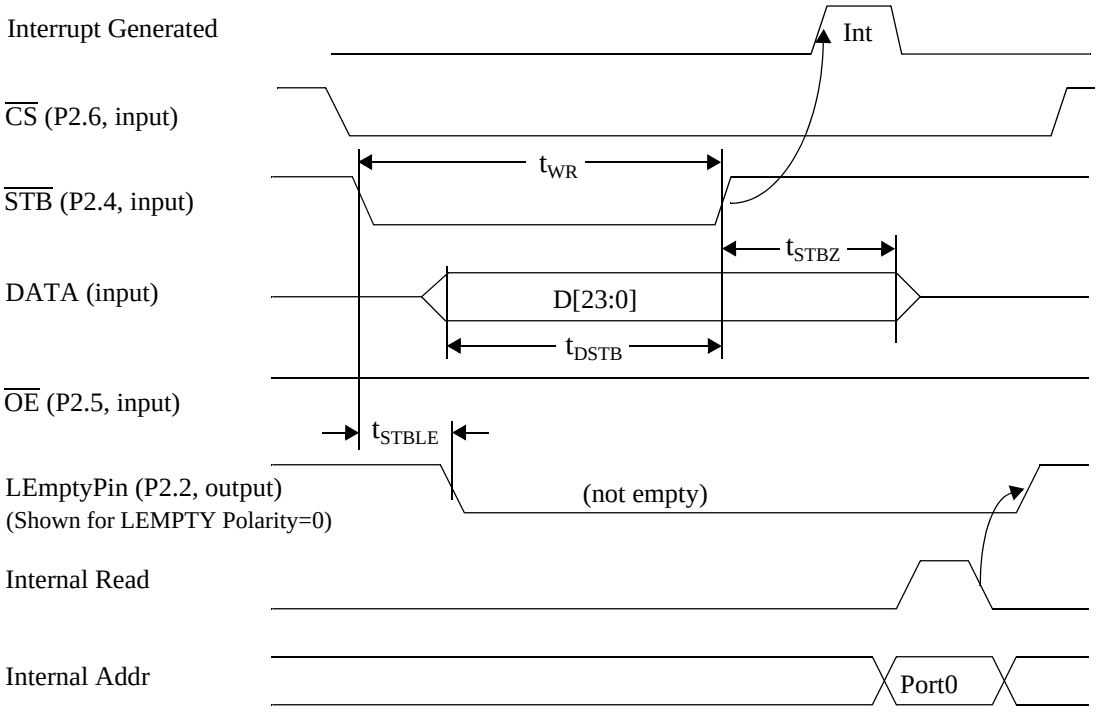


Figure 24-4. HAPI Write by External Device to USB Microcontroller

25.0 Ordering Information

Ordering Code	PROM Size	Package Name	Package Type	Operating Range
CY7C64013-SC	8 KB	S21	28-Pin (300-Mil) SOIC	Commercial
CY7C64013-PC	8 KB	P21	28-Pin (300-Mil) PDIP	Commercial
CY7C64113-PVC	8 KB	O48	48-Pin (300-Mil) SSOP	Commercial