Infineon Technologies - CY7C66113C-LFXC Datasheet

Welcome to E-XFL.COM

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

Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

What Are Embedded - Microcontrollers - Application Specific?

Application specific microcontrollers are engineered to

Details	
Product Status	Obsolete
Applications	USB Hub/Microcontroller
Core Processor	M8
Program Memory Type	OTP (8kB)
Controller Series	USB Hub
RAM Size	256 x 8
Interface	I²C, USB, HAPI
Number of I/O	31
Voltage - Supply	4V ~ 5.25V
Operating Temperature	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	56-VFQFN Exposed Pad
Supplier Device Package	56-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy7c66113c-lfxc

Pin Configurations

Figure 1. CY7C66013C 48-Pin SSOP and CY7C66113C 56-Pin SSOP

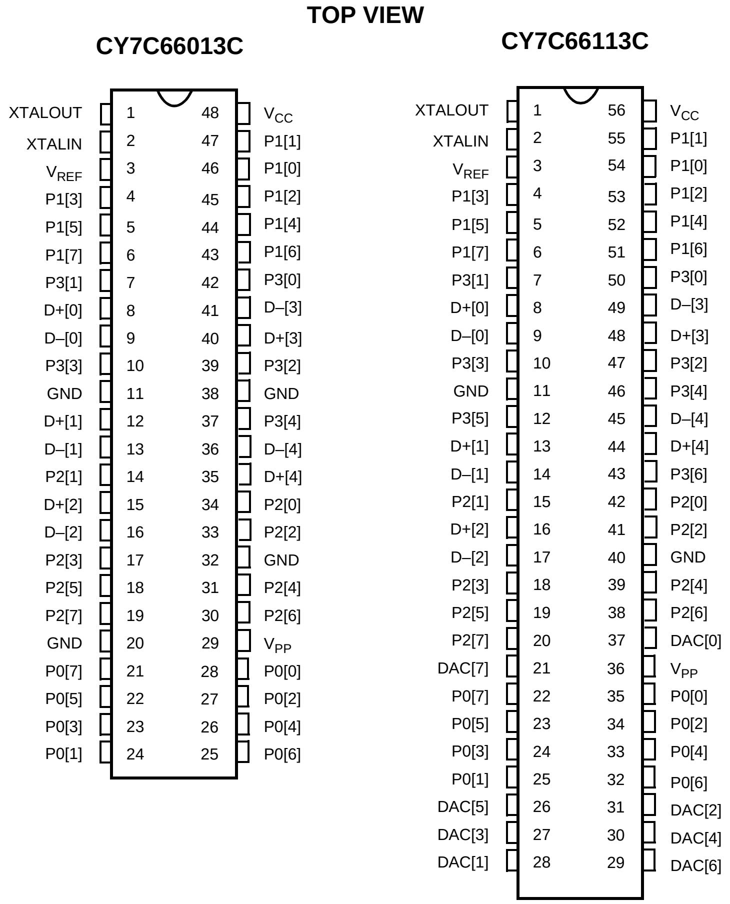


Table 3. I/O Register Summary (continued)

Register Name	I/O Address	Read/Write	Function	Page
HAPI and I^{2}C Configuration	0x09	R/W	HAPI Width and I^{2}C Position Configuration	22
USB Device Address A	0x10	R/W	USB Device Address A	38
EP A0 Counter Register	0x11	R/W	USB Address A, Endpoint 0 Counter	40
EP A0 Mode Register	0x12	R/W	USB Address A, Endpoint 0 Configuration	39
EP A1 Counter Register	0x13	R/W	USB Address A, Endpoint 1 Counter	40
EP A1 Mode Register	0x14	R/W	USB Address A, Endpoint 1 Configuration	40
EP A2 Counter Register	0x15	R/W	USB Address A, Endpoint 2 Counter	40
EP A2 Mode Register	0x16	R/W	USB Address A, Endpoint 2 Configuration	40
USB Status & Control	0x1F	R/W	USB Upstream Port Traffic Status and Control	37
Global Interrupt Enable	0x20	R/W	Global Interrupt Enable	27
Endpoint Interrupt Enable	0x21	R/W	USB Endpoint Interrupt Enables	27
Interrupt Vector	0x23	R	Pending Interrupt Vector Read/Clear	29
Timer (LSB)	0x24	R	Lower 8 Bits of Free-running Timer (1 MHz)	21
Timer (MSB)	0x25	R	Upper 4 Bits of Free-running Timer	21
WDT Clear	0x26	W	Watchdog Timer Clear	14
I^{2}C Control & Status	0x28	R/W	I^{2}C Status and Control	23
I^{2}C Data	0x29	R/W	I^{2}C Data	23
DAC Data	0x30	R/W	DAC Data	19
DAC Interrupt Enable	0x31	W	Interrupt Enable for each DAC Pin	20
DAC Interrupt Polarity	0x32	W	Interrupt Polarity for each DAC Pin	20
DAC Isink	0x38-0x3F	W	Input Sink Current Control for each DAC Pin	20
USB Device Address B	0x40	R/W	USB Device Address B (not used in 5-endpoint mode)	38
EP B0 Counter Register	0x41	R/W	USB Address B, Endpoint 0 Counter	40
EP B0 Mode Register	0x42	R/W	USB Address B, Endpoint 0 Configuration, or USB Address A, Endpoint 3 in 5-endpoint Mode	39
EP B1 Counter Register	0x43	R/W	USB Address B, Endpoint 1 Counter	40
EP B1 Mode Register	0x44	R/W	USB Address B, Endpoint 1 Configuration, or USB Address A, Endpoint 4 in 5-endpoint Mode	40
Hub Port Connect Status	0x48	R/W	Hub Downstream Port Connect Status	32
Hub Port Enable	0x49	R/W	Hub Downstream Ports Enable	33
Hub Port Speed	0x4A	R/W	Hub Downstream Ports Speed	33
Hub Port Control (Ports [4:1])	0x4B	R/W	Hub Downstream Ports Control	34
Hub Port Suspend	0x4D	R/W	Hub Downstream Port Suspend Control	35
Hub Port Resume Status	0x4E	R	Hub Downstream Ports Resume Status	36
Hub Ports SE0 Status	0x4F	R	Hub Downstream Ports SE0 Status	35
Hub Ports Data	0x50	R	Hub Downstream Ports Differential Data	35
Hub Downstream Force Low	0x51	R/W	Hub Downstream Ports Force LOW	34
Processor Status & Control	0xFF	R/W	Microprocessor Status and Control Register	26

Suspend Mode

The CY7C66x13C is 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, PLL, and the free-running and WDTs 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 from 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.

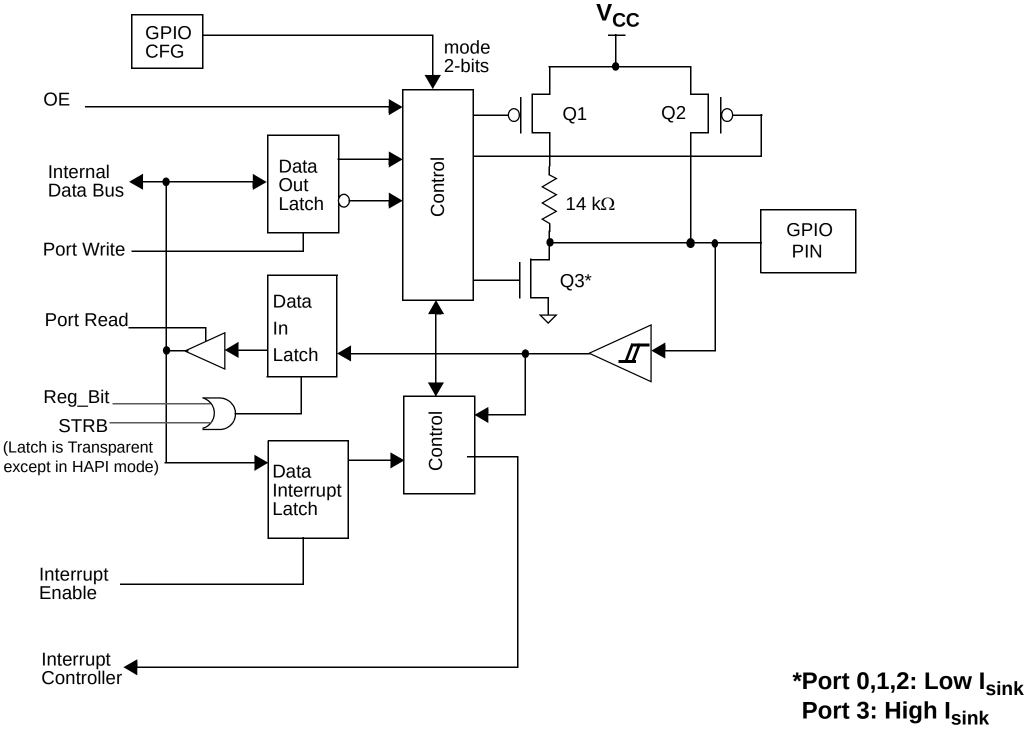
The GPIO interrupt allows the controller to wake up periodically and poll system components while maintaining a very low average power consumption. To achieve the lowest possible current during suspend mode, all I/O should be held at V_{CC} or Gnd. This also applies to internal port pins that may not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

Typical code for entering suspend is given here:

```
     ...  ; All GPIO set to low power state (no
floating pins)
     ...  ; Enable GPIO interrupts if desired
for wakeup
     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
```

General Purpose I/O (GPIO) Ports

Figure 7. Block Diagram of a GPIO Pin



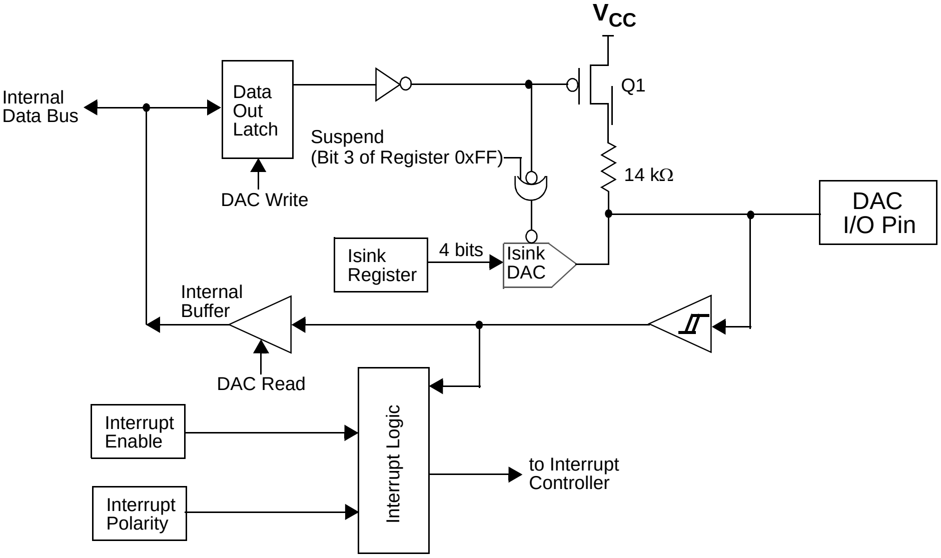
***Port 0,1,2: Low I_{sink}**
Port 3: High I_{sink}

There are up to 31 GPIO pins (P0[7:0], P1[7:0], P2[7:0], and P3[6:0]) for the hardware interface. The number of GPIO pins changes based on the package type of the chip. Each port is configured as inputs with internal pull ups, open drain outputs, or traditional CMOS outputs. Port 3 offers a higher current drive, with typical current sink capability of 12 mA. The data for each GPIO port is accessible through the data registers. Port data registers are shown in Figure 8 through Figure 11, and are set to 1 on reset.

DAC Port

The CY7C66113CC features a programmable sink current 8-bit port, which is also known as DAC port. Each of these port I/O pins have a programmable current sink. Writing a ‘1’ to a DAC I/O pin disables the output current sink (I_{sink} DAC) and drives the I/O pin HIGH through an integrated 14 kΩ resistor. When a ‘0’ is written to a DAC I/O pin, the I_{sink} DAC is enabled and the pull up resistor is disabled. This causes the I_{sink} DAC to sink current to drive the output LOW. Figure 17 shows a block diagram of the DAC port pin.

Figure 17. 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 (Figure 19) 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 (Figure 28) is set, the Isink DAC block of the DAC circuitry is disabled. Special care should be taken when the CY7C66113C device is placed in the suspend. The DAC Port Data Register (Figure 18) should normally be loaded with all ‘1’s (Figure 28) before setting the suspend bit. If any of the DAC bits are set to ‘0’ when the device is suspended, that DAC input floats. The floating pin could result in excessive current consumption by the device, unless an external load places the pin in a deterministic state.

Figure 18. DAC Port Data

DAC Port Data **ADDRESS 0x30**

Bit #	7	6	5	4	3	2	1	0
Bit Name	DAC[7]	DAC[6]	DAC[5]	DAC[4]	DAC[3]	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

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 [7..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.

Figure 26. I²C Data Register

I²C Data **ADDRESS 0x29**

Bit #	7	6	5	4	3	2	1	0
Bit Name	I²C Data 7	I²C Data 6	I²C Data 5	I²C Data 4	I²C Data 3	I²C Data 2	I²C Data 1	I²C Data 0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	X	X	X	X	X	X	X	X

Bits [7..0]: I²C Data

Contains 8-bit data on the I²C Bus.

Figure 27. I²C Status and Control Register

I²C Status and Control **ADDRESS 0x28**

Bit #	7	6	5	4	3	2	1	0
Bit Name	MSTR Mode	Continue/Busy	Xmit Mode	ACK	Addr	ARB Lost/Restart	Received Stop	I²C Enable
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

The I²C Status and Control register bits are defined in Table 9, with a more detailed description following.

Table 9. I²C Status and Control Register Bit Definitions

Bit	Name	Description
0	I²C Enable	When set to ‘1’, the I²C compatible function is enabled. When cleared, I²C GPIO pins operate normally.
1	Received Stop	Reads 1 only in slave receive mode, when I²C Stop bit detected (unless firmware did not ACK the last transaction).
2	ARB Lost/Restart	Reads 1 to indicate master has lost arbitration. Reads 0 otherwise. Write to 1 in master mode to perform a restart sequence (also set Continue bit).
3	Addr	Reads 1 during first byte after start/restart in slave mode, or if master loses arbitration. Reads 0 otherwise. This bit should always be written as 0.
4	ACK	In receive mode, write 1 to generate ACK, 0 for no ACK. In transmit mode, reads 1 if ACK was received, 0 if no ACK received.
5	Xmit Mode	Write to 1 for transmit mode, 0 for receive mode.
6	Continue/Busy	Write 1 to indicate ready for next transaction. Reads 1 when I²C compatible block is busy with a transaction, 0 when transaction is complete.
7	MSTR Mode	Write to 1 for master mode, 0 for slave mode. This bit is cleared if master loses arbitration. Clearing from 1 to 0 generates Stop bit.

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 is read at Bit 2 of the Processor Status and Control Register, Figure 28).
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 (that is, the Interrupt Vector).

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

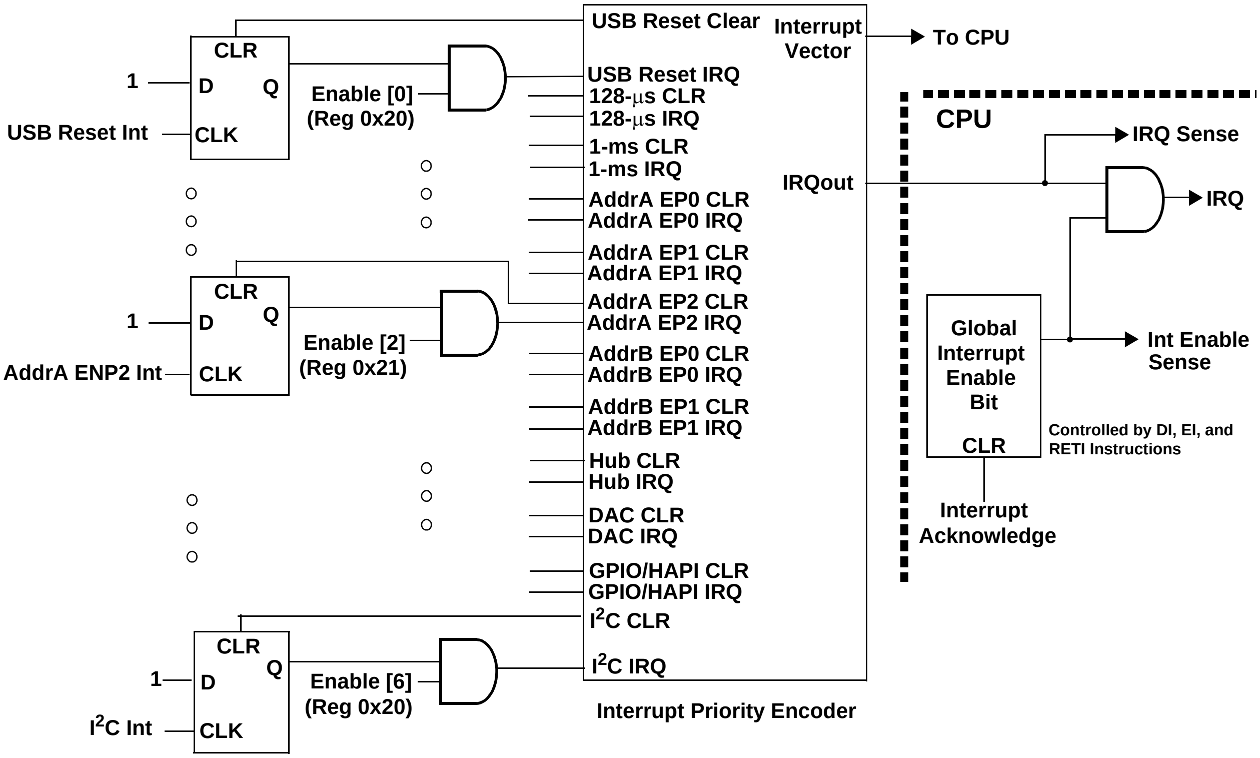
The Program Counter value and the Carry and Zero flags (CF, ZF) are stored onto the Program Stack by the automatic CALL instruction generated as part of the interrupt acknowledge process. The user firmware is responsible for ensuring that the processor state is preserved and restored during an interrupt. 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 are used to disable and enable interrupts, respectively. These instructions affect only the Global Interrupt Enable bit of the CPU. If desired, EI is 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 is detected by examining the IRQ Sense bit (Bit 7 in the Processor Status and Control Register).

Interrupt Vectors

The Interrupt Vectors supported by the USB Controller are listed in Table 11. 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 31. 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 11. 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 B Endpoint 0 interrupt
8	0x0010	USB Address B Endpoint 1 interrupt
9	0x0012	USB Hub interrupt
10	0x0014	DAC interrupt
11	0x0016	GPIO and HAPI interrupt
12	0x0018	I^2C interrupt

Interrupt Latency

Interrupt latency is 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 five 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.

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. SE0 is defined as the condition in which both the D+ line and the D– line are LOW. A USB Bus Reset may be recognized for an SE0 as short as 12 µs, but is always recognized for an SE0 longer than 16 µs. When a USB Bus Reset is detected, bit 5 of the Processor Status and Control Register *(*Figure 28) is set to record this event. In addition, the controller clears the following registers:

SIE Section:	USB Device Address Registers (0x10, 0x40)
Hub Section:	Hub Ports Connect Status (0x48)
	Hub Ports Enable (0x49)
	Hub Ports Speed (0x4A)
	Hub Ports Suspend (0x4D)
	Hub Ports Resume Status (0x4E)
	Hub Ports SE0 Status (0x4F)
	Hub Ports Data (0x50)
	Hub Downstream Force (0x51).

A USB Bus Reset Interrupt is generated at the end of the USB Bus Reset condition when the SE0 state is deasserted. If the USB reset occurs during the start up delay following a POR, the delay is aborted as described in Power on Reset on page 14.

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.

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. For example, 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.

USB Hub Interrupt

A USB hub interrupt is generated by the hardware after a connect/disconnect change, babble, or a resume event is detected by the USB repeater hardware. The babble and resume events are additionally gated by the corresponding bits of the Hub Port Enable Register (Figure 35). The connect and disconnect event on a port does not generate an interrupt if the SIE does not drive the port (that is, the port is being forced).

Figure 34. Hub Ports Speed

Hub Ports Speed **ADDRESS 0x4A**

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

Bit [0..3]: Port x Speed (where x = 1..4)

Set to 1 if the device plugged in to Port x is Low speed; Set to 0 if the device plugged in to Port x is Full speed.

Bit [7..4]: Reserved.

The Hub Ports Speed register is cleared to zero by reset or bus reset. This must be set by the firmware on issuing a port reset. The Reserved bits [7..4] should always read as ‘0.’

Enabling and Disabling a USB Device

After a USB device connection is detected, firmware must update status change bits in the hub status change data structure that is polled periodically by the USB host. The host responds by sending a packet that instructs the hub to reset and enable the downstream port. Firmware then sets the bit in the Hub Ports Enable register, Figure 35, for the downstream port. The hub repeater hardware responds to an enable bit in the Hub Ports Enable register by enabling the downstream port, so that USB traffic flows to and from that port.

If a port is marked enabled and is not suspended, it receives all USB traffic from the upstream port, and USB traffic from the downstream port is passed to the upstream port (unless babble is detected). Low speed ports do not receive full speed traffic from the upstream port.

When firmware writes to the Hub Ports Enable register to enable a port, the port is not enabled until the end of any packet currently being transmitted. If there is no USB traffic, the port is enabled immediately.

When a USB device disconnection is detected, firmware must update status bits in the hub change status data structure that is polled periodically by the USB host. In suspend, a connect or disconnect event generates an interrupt (if the hub interrupt is enabled) even if the port is disabled.

Figure 35. Hub Ports Enable Register

Hub Ports Enable Register **ADDRESS 0x49**

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

Bit [0..3]: Port x Enable (where x = 1..4)

Set to 1 if Port x is enabled; Set to 0 if Port x is disabled.

Bit [7..4]: Reserved.

The Hub Ports Enable register is cleared to zero by reset or bus reset to disable all downstream ports as the default condition. A port is also disabled by internal hub hardware (enable bit cleared) if babble is detected on that downstream port. Babble is defined as:

- Any non idle downstream traffic on an enabled downstream port at EOF2
- Any downstream port with upstream connectivity established at EOF2 (that is, no EOP received by EOF2).

Figure 38. Hub Ports SE0 Status Register

Hub Ports SE0 Status **ADDRESS 0x4F**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Reserved	Reserved	Reserved	Reserved	Port 4 SE0 Status	Port 3 SE0 Status	Port 2 SE0 Status	Port 1 SE0 Status
Read/Write	-	-	-	-	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit [0..3]: Port x SE0 Status (where x = 1..4)

Set to 1 if a SE0 is output on the Port x bus; Set to 0 if a Non-SE0 is output on the Port x bus.

Bit [7..4]: Reserved.

Figure 39. Hub Ports Data Register

Hub Ports Data **ADDRESS 0x50**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Reserved	Reserved	Reserved	Reserved	Port 4 Diff. Data	Port 3 Diff. Data	Port 2 Diff. Data	Port 1 Diff. Data
Read/Write	-	-	-	-	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit [0..3]: Port x Diff Data (where x = 1..4)

Set to 1 if D+ > D– (forced differential 1, if signal is differential, i.e. not a SE0 or SE1). Set to 0 if D– > D+ (forced differential 0, if signal is differential, i.e., not a SE0 or SE1);

Bit [7..4]: Reserved.

Downstream Port Suspend and Resume

The Hub Ports Suspend Register (Figure 40) and Hub Ports Resume Status Register (Figure 41) indicate the suspend and resume conditions on downstream ports. The suspend register must be set by firmware for any ports that are selectively suspended. Also, this register is only valid for ports that are selectively suspended.

If a port is marked as selectively suspended, normal USB traffic is not sent to that port. Resume traffic is also prevented from going to that port, unless the Resume comes from the selectively suspended port. If a resume condition is detected on the port, hardware reflects a Resume back to the port, sets the Resume bit in the Hub Ports Resume Register, and generates a hub interrupt. If a disconnect occurs on a port marked as selectively suspended, the suspend bit is cleared.

The Device Remote Wakeup bit (bit 7) of the Hub Ports Suspend Register controls whether or not the resume signal is propagated by the hub after a connect or a disconnect event. If the Device Remote Wakeup bit is set, the hub automatically propagates the resume signal after a connect or a disconnect event. If the Device Remote Wakeup bit is cleared, the hub does not propagate the resume signal. The setting of the Device Remote Wakeup flag has no impact on the propagation of the resume signal after a downstream remote wakeup event. The hub automatically propagates the resume signal after a remote wakeup event, regardless of the state of the Device Remote wakeup bit. The state of this bit has no impact on the generation of the hub interrupt. These registers are cleared on reset or USB bus reset.

Figure 40. Hub Ports Suspend Register

Hub Ports Suspend **ADDRESS 0x4D**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Device Remote Wakeup	Reserved	Reserved	Reserved	Port 4 Selective Suspend	Port 3 Selective Suspend	Port 2 Selective Suspend	Port 1 Selective Suspend
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit [0..3]: Port x Selective Suspend (where x = 1..4)

Set to 1 if Port x is Selectively Suspended; Set to 0 if Port x Do not suspend.

Bit 7: Device Remote Wakeup.

When set to 1, Enable hardware upstream resume signaling for connect and disconnect events during global resume.

When set to 0, Disable hardware upstream resume signaling for connect and disconnect events during global resume.

USB Upstream Port Status and Control

USB status and control is regulated by the USB Status and Control Register, as shown in Figure 42. All bits in the register are cleared during reset.

Figure 42. USB Status and Control Register

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

Bits[2..0]: Control Action

Set to control action as per Table 13.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 13 shows how the control bits affect the upstream port.

Table 13. 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 clears 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 USB Device Endpoints for a detailed description.

Bit 7: Endpoint Size

This bit used to configure the number of USB endpoints. See USB Device Endpoints for a detailed description.

The hub generates an EOP at EOF1 in accordance with the USB 1.1 Specification, *Section 11.2.2.*

USB Mode Tables

Table 15. 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[4]=0) Ack Out(STALL[4]=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[4]=0) Ack IN(STALL[4]=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 are 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 responds 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 I^{2}C Compatible Controller on page 22 for more information on 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 responds 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.

An “Ignore” in any of the columns means that the device does not send any handshake tokens (no ACK) to the host.

An “Accept” in any of the columns means that the device responds with an ACK to a valid SETUP transaction to the host.

Comments

Some Mode Bits are automatically changed by the SIE in response to certain USB transactions. For example, if the Mode Bits [3:0] are set to '1111' which is ACK IN-Status OUT mode as shown in Table 14, the SIE changes the endpoint Mode Bits [3:0] to NAK IN-Status OUT mode (1110) after ACK’ing a valid status

Note

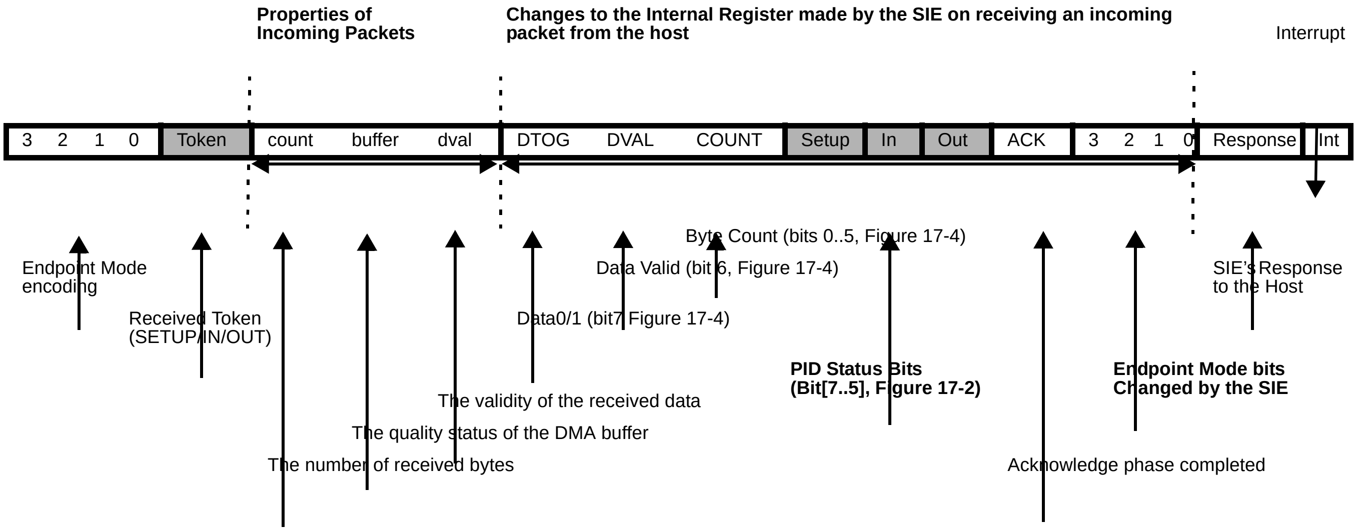
[4] STALL bit is bit 7 of the USB Non Control Device Endpoint Mode registers. For more information, refer to USB Non Control Endpoint Mode Registers on page 40.

stage OUT token. The firmware needs to update the mode for the SIE to respond appropriately. See Table 12 for more details on what modes are changed by the SIE. A disabled endpoint remains disabled until changed by firmware, and all endpoints reset to the disabled mode (0000). Firmware normally enables the endpoint mode after a SetConfiguration request.

Any SETUP packet to an enabled endpoint with mode set to accept SETUPs are changed by the SIE to 0001 (NAKing INs and OUTs). Any mode set to accept a SETUP sends an ACK handshake to a valid SETUP token.

The control endpoint has three status bits for identifying the token type received (SETUP, IN, or OUT), but the endpoint must be placed in the correct mode to function as such. Non control endpoints should not be placed into modes that accept SETUPs. Note that most modes that control transactions involving an ending ACK, are changed by the SIE to a corresponding mode which NAKs subsequent packets following the ACK. Exceptions are modes 1010 and 1110.

Table 16. Decode Table for Table 17


Properties of Incoming Packets
Changes to the Internal Register made by the SIE on receiving an incoming packet from the host
Interrupt
3 2 1 0
Token
count
buffer
dval
DTOG
DVAL
COUNT
Setup
In
Out
ACK
3 2 1 0
Response
Int
Byte Count (bits 0..5, Figure 17-4)
Endpoint Mode encoding
Data Valid (bit 6, Figure 17-4)
SIE's Response to the Host
Received Token (SETUP/IN/OUT)
Data0/1 (bit7 Figure 17-4)
PID Status Bits (Bit[7..5], Figure 17-2)
Endpoint Mode bits Changed by the SIE
The validity of the received data
The quality status of the DMA buffer
The number of received bytes
Acknowledge phase completed

Legend:	TX: transmit	UC : unchanged
	RX: receive	TX0:Transmit 0 length packet
	▨	available for Control endpoint only
	x: don't care	

The response of the SIE are summarized as follows:

- The SIE only responds to valid transactions, and ignores invalid ones.
- The SIE generates an interrupt when a valid transaction is completed or when the FIFO is corrupted. FIFO corruption occurs during an OUT or SETUP transaction to a valid internal address, that ends with a invalid CRC.
- An incoming Data packet is valid if the count is ≤ Endpoint Size + 2 (includes CRC) and passes all error checking.
- An IN is ignored by an OUT configured endpoint and visa versa.
- The IN and OUT PID status is updated at the end of a transaction.
- The SETUP PID status is updated at the beginning of the Data packet phase.
- The entire Endpoint 0 mode register and the Count register are locked to CPU writes at the end of any transaction to that endpoint in which an ACK is transferred. These registers are only unlocked by a CPU read of the register, which should be done by the firmware only after the transaction is complete. This represents about a 1 μs window in which the CPU is locked from register writes to these USB registers. Normally the firmware should perform a register read at the beginning of the Endpoint ISRs to unlock and get the mode register information. The interlock on the Mode and Count registers ensures that the firmware recognizes the changes that the SIE might have made during the previous transaction. Note that the setup bit of the mode register is NOT locked. This means that before writing to the mode register, firmware must first read the register to make sure that the setup bit is not set (which indicates a setup was received, while processing the current USB request). This read unlocks the register. So care must be taken not to overwrite the register elsewhere.

Table 17. Details of Modes for Differing Traffic Conditions (see Table 16 for the decode legend) (continued)

Nak In/premature status Out																				
1	1	1	0	Out	2	UC	valid	1	1	updates	UC	UC	1	1	No Change				ACK	yes
1	1	1	0	Out	2	UC	valid	0	1	updates	UC	UC	1	UC	0	0	1	1	Stall	yes
1	1	1	0	Out	!=2	UC	valid	updates	1	updates	UC	UC	1	UC	0	0	1	1	Stall	yes
1	1	1	0	Out	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
1	1	1	0	Out	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
1	1	1	0	In	x	UC	x	UC	UC	UC	UC	1	UC	UC	No Change				NAK	yes
Status Out/extra In																				
0	0	1	0	Out	2	UC	valid	1	1	updates	UC	UC	1	1	No Change				ACK	yes
0	0	1	0	Out	2	UC	valid	0	1	updates	UC	UC	1	UC	0	0	1	1	Stall	yes
0	0	1	0	Out	!=2	UC	valid	updates	1	updates	UC	UC	1	UC	0	0	1	1	Stall	yes
0	0	1	0	Out	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
0	0	1	0	Out	x	UC	invalid	UC	UC	UC	UC	1	UC	UC	No Change				ignore	no
0	0	1	0	In	x	UC	x	UC	UC	UC	UC	1	UC	UC	0	0	1	1	Stall	yes
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	1	1	1	0	0	0	ACK	yes
1	0	0	1	Out	> 10	junk	x	updates	updates	updates	UC	UC	1	UC	No Change				ignore	yes
1	0	0	1	Out	x	junk	invalid	updates	0	updates	UC	UC	1	UC	No Change				ignore	yes
1	0	0	1	In	x	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
																			(STALL[4] = 0)	
1	0	0	1	In	x	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				Stall	no
																			(STALL[4] = 1)	
NAK Out/erroneous In																				
1	0	0	0	Out	<= 10	UC	valid	UC	UC	UC	UC	UC	1	UC	No Change				NAK	yes
1	0	0	0	Out	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
1	0	0	0	Out	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
1	0	0	0	In	x	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
Isochronous endpoint (Out)																				
0	1	0	1	Out	x	updates	updates	updates	updates	updates	UC	UC	1	1	No Change				RX	yes
0	1	0	1	In	x	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				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	No Change				ignore	no
																			(STALL[4] = 0)	
1	1	0	1	Out	x	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				stall	no
																			(STALL[4] = 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	No Change				ignore	no

Table 17. Details of Modes for Differing Traffic Conditions (see Table 16 for the decode legend) (continued)

1	1	0	0	In	x	UC	x	UC	UC	UC	UC	1	UC	UC	No Change	NAK	yes
Isochronous endpoint (In)																	
0	1	1	1	Out	x	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change	ignore	no
0	1	1	1	In	x	UC	x	UC	UC	UC	UC	1	UC	UC	No Change	TX	yes

Register Summary

	Addr	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Read/Write/ Both[5, 6, 7]	Default/ Reset [8]
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	Reserved	P3.6 CY7C66113C only	P3.5 CY7C66113C only	P3.4	P3.3	P3.2	P3.1	P3.0	bbbbbbbb	-1111111
	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	P1.3 Intr Enable	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	P2.2 Intr Enable	P2.1 Intr Enable	P2.0 Intr Enable	wwwwwwww	00000000
	0x07	Port 3 Interrupt Enable	Reserved	P3.6 Intr Enable CY7C66113C only	P3.5 Intr Enable CY7C66113C only	P3.4 Intr Enable	P3.3 Intr Enable	P3.2 Intr Enable	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 I^2C	0x09	HAPI/I^2C Configuration	I^2C Position	Reserved	LEMPTY Polarity	DRDY Polarity	Latch Empty	Data Ready	Port Width bit 1	Port Width bit 0	b-bbrrbb	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
Endpoint A0, AI AND A2 Configuration	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

Notes

[5] B: Read and Write.

[6] W: Write.

[7] R: Read.

[8] X: Unknown

Register Summary (continued)

	Addr	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Read/Write/ Both[5, 6, 7]	Default/ Reset [8]
INTERRUPT	0x20	Global Interrupt Enable	Reserved	I^{2}C Interrupt Enable	GPIO Interrupt Enable	DAC Interrupt Enable	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
I^{2}C	0x28	I^{2}C Control and Status	MSTR Mode	Continue/ Busy	Xmit Mode	ACK	Addr	ARB Lost/ Restart	Received Stop	I^{2}C Enable	bbbbbbbb	00000000
	0x29	I^{2}C Data	I^{2}C Data 7	I^{2}C Data 6	I^{2}C Data 5	I^{2}C Data 4	I^{2}C Data 3	I^{2}C Data 2	I^{2}C Data 1	I^{2}C Data 0	bbbbbbbb	xxxxxxxx
ENDPOINT B0, B1 CONFIGURATION	0x40	USB Device Address B	Device Address B Enable	Device Address B Bit 6	Device Address B Bit 5	Device Address B Bit 4	Device Address B Bit 3	Device Address B Bit 2	Device Address B Bit 1	Device Address B Bit 0	bbbbbbbb	00000000
	0x41	EP B0 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 B0 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 B1 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 B1 Mode Register	STALL	-	-	ACK	Mode Bit 3	Mode Bit 2	Mode Bit 1	Mode Bit 0	b--bbbbb	00000000
HUB PORT CONTROL, STATUS, SUSPEND RESUME, SE0, FORCE LOW	0x48	Hub Port Connect Status	Reserved	Reserved	Reserved	Reserved	Port 4 Connect Status	Port 3 Connect Status	Port 2 Connect Status	Port 1 Connect Status	----bbbb	00000000
	0x49	Hub Port Enable	Reserved	Reserved	Reserved	Reserved	Port 4 Enable	Port 3 Enable	Port 2 Enable	Port 1 Enable	----bbbb	00000000
	0x4A	Hub Port Speed	Reserved	Reserved	Reserved	Reserved	Port 4 Speed	Port 3 Speed	Port 2 Speed	Port 1 Speed	----bbbb	00000000
	0x4B	Hub Port Control (Ports 4:1)	Port 4 Control Bit 1	Port 4 Control Bit 0	Port 3 Control Bit 1	Port 3 Control Bit 0	Port 2 Control Bit 1	Port 2 Control Bit 0	Port 1 Control Bit 1	Port 1 Control Bit 0	bbbbbbbb	00000000
	0x4D	Hub Port Suspend	Device Remote Wakeup	Reserved	Reserved	Reserved	Port 4 Selective Suspend	Port 3 Selective Suspend	Port 2 Selective Suspend	Port 1 Selective Suspend	b---bbbb	00000000
	0x4E	Hub Port Resume Status	Reserved	Reserved	Reserved	Reserved	Resume 4	Resume 3	Resume 2	Resume 1	----rrrr	00000000
	0x4F	Hub Port SE0 Status	Reserved	Reserved	Reserved	Reserved	Port 4 SE0 Status	Port 3 SE0 Status	Port 2 SE0 Status	Port 1 SE0 Status	----rrrr	00000000
	0x50	Hub Ports Data	Reserved	Reserved	Reserved	Reserved	Port 4 Diff. Data	Port 3 Diff. Data	Port 2 Diff. Data	Port 1 Diff. Data	----rrrr	00000000
	0x51	Hub Port Force Low (Ports 4:1)	Force Low D+[4]	Force Low D–[4]	Force Low D+[3]	Force Low D–[3]	Force Low D+[2]	Force Low D–[2]	Force Low D+[1]	Force Low D–[1]	bbbbbbbb	00000000
	0xFF	Process Status & Control	IRQ Pending	WDR	USB Bus Reset Interrupt	Power-on Reset	Suspend	Interrupt Enable Sense	Reserved	Run	rbbbbrbb	00010001

Switching Characteristics (F_{OSC} = 6.0 MHz)

Parameter	Description	Min	Max	Unit
HAPI Write Cycle Timing				
t_{WR}	Write Strobe Width	15		ns
t_{DSTB}	Data Valid to STB HIGH (Data Setup Time)[16]	5		ns
t_{STBZ}	STB HIGH to Data High-Z (Data Hold Time)[16]	15		ns
t_{STBLE}	STB LOW to Latch_Empty Deasserted[15, 16]	0	50	ns
Timer Signals				
t_{watch}	WDT Period	8.192	14.336	ms

Figure 49. Clock Timing

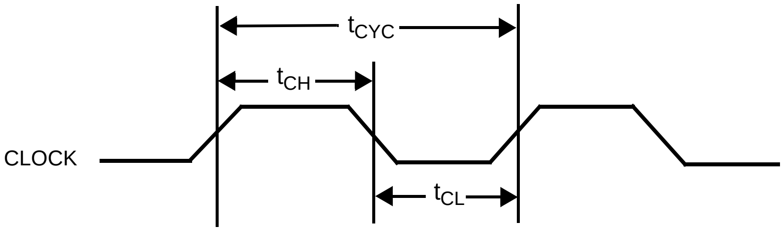


Figure 50. USB Data Signal Timing

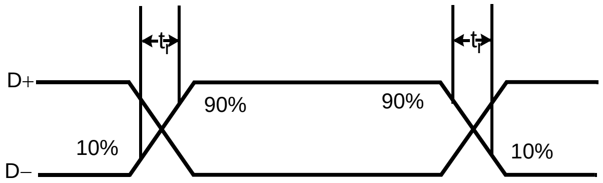


Figure 51. HAPI Read by External Interface from USB Microcontroller

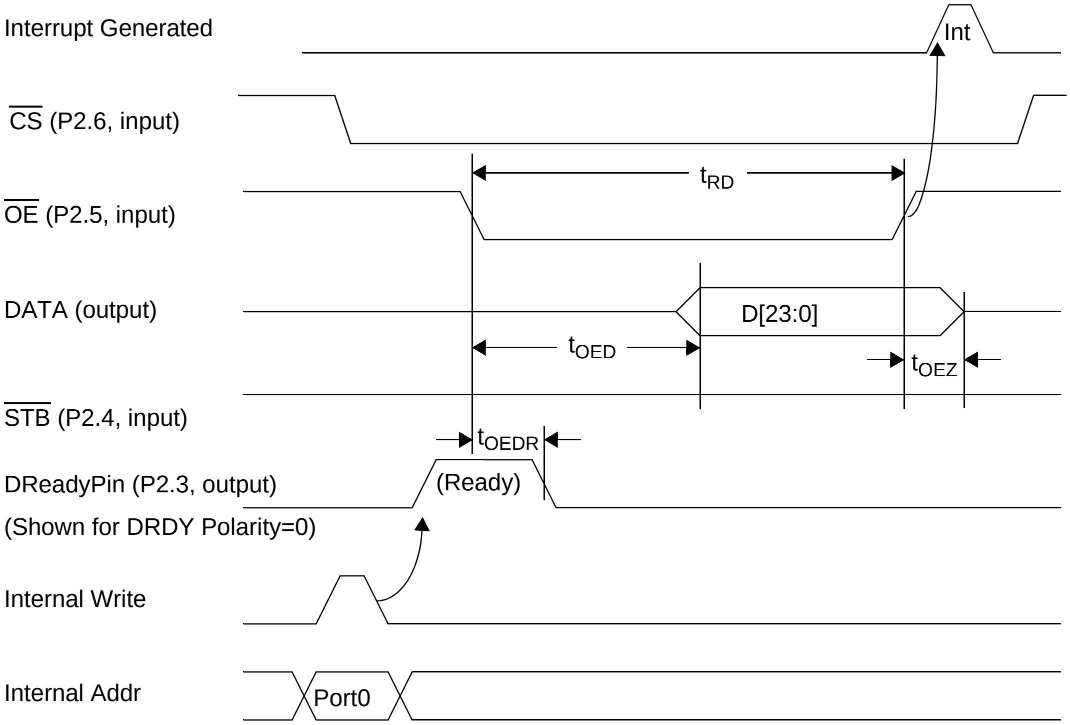
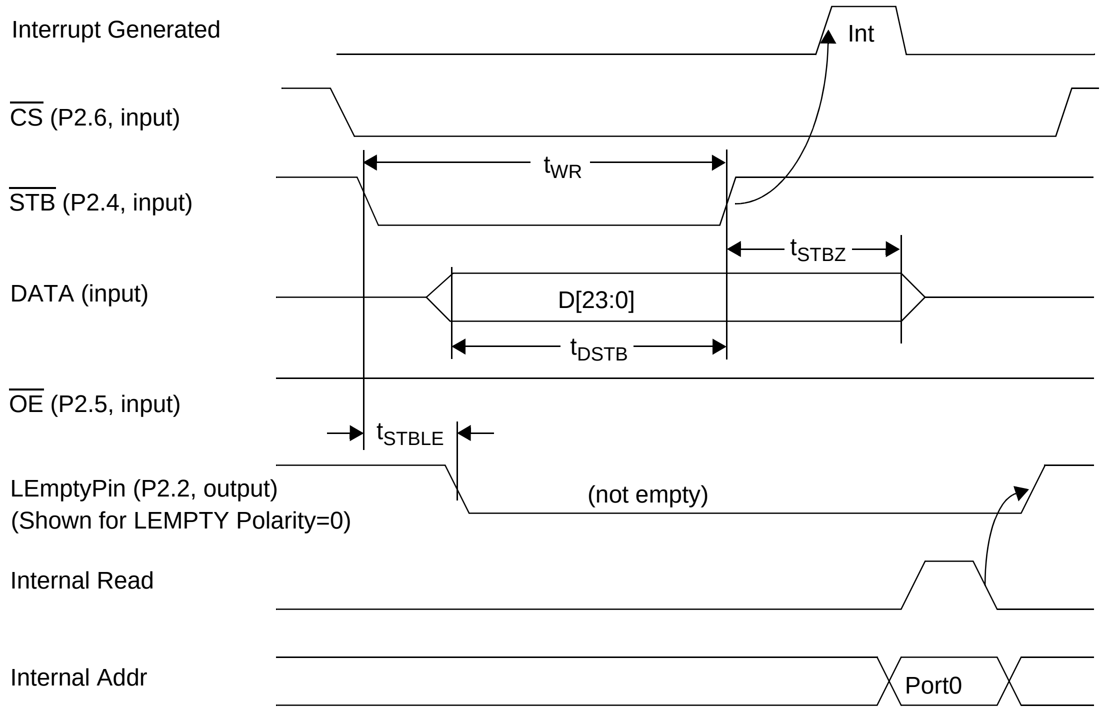


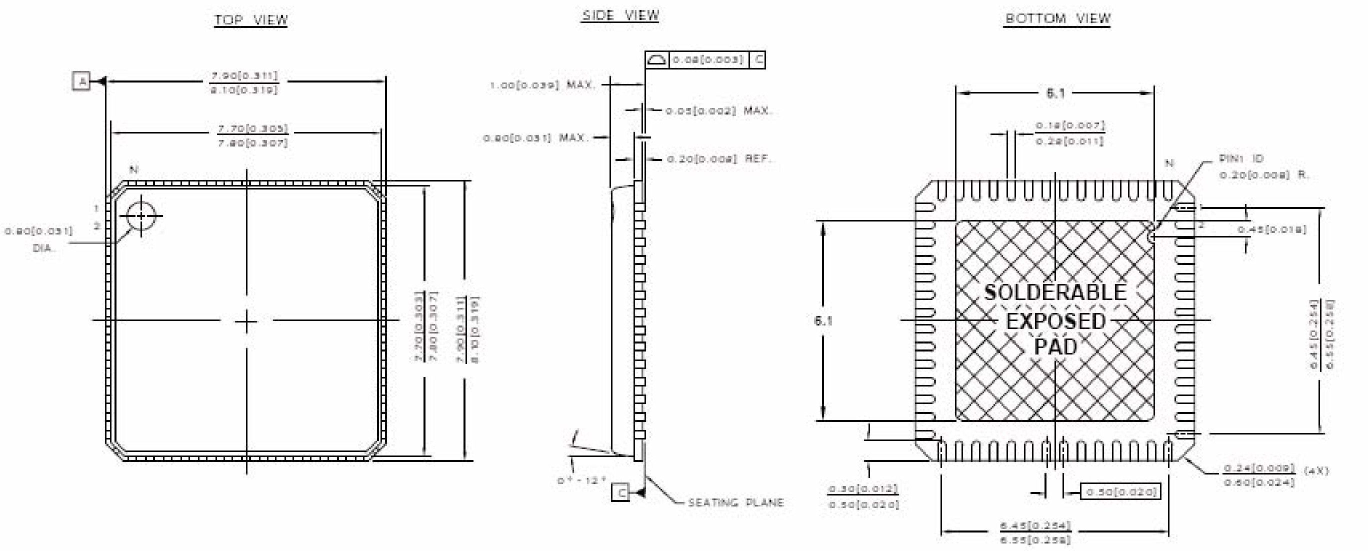
Figure 52. HAPI Write by External Device to USB Microcontroller



Ordering Information

Ordering Code	PROM Size	Package Type	Operating Range
CY7C66013C-PVXC	8 KB	48-pin (300-Mil) SSOP	Commercial
CY7C66113C-PVXC	8 KB	56-pin (300-Mil) SSOP	Commercial
CY7C66113C-LFXC	8 KB	56-pin QFN	Commercial
CY7C66113C-PVXCT	8 KB	56-pin (300-Mil) SSOP	Commercial
CY7C66113C-XC	8 KB	Die	Commercial
CY7C66113C-LTXC	8 KB	56-pin QFN	Commercial
CY7C66113C-LTXCT	8 KB	56-pin QFN	Commercial

Figure 55. 56-Pin QFN 8 x 8 MM LF56A



NOTES:

1. ▨ HATCH AREA IS SOLDERABLE EXPOSED METAL.
2. REFERENCE JEDEC#: MO-220
3. PACKAGE WEIGHT: 0.162g
4. ALL DIMENSIONS ARE IN MM [MIN/MAX]
5. PACKAGE CODE

PART #	DESCRIPTION
LF56	STANDARD
LY56	PB-FREE

51-85144 *G

Quad Flat Package No Leads (QFN) Package Design Notes

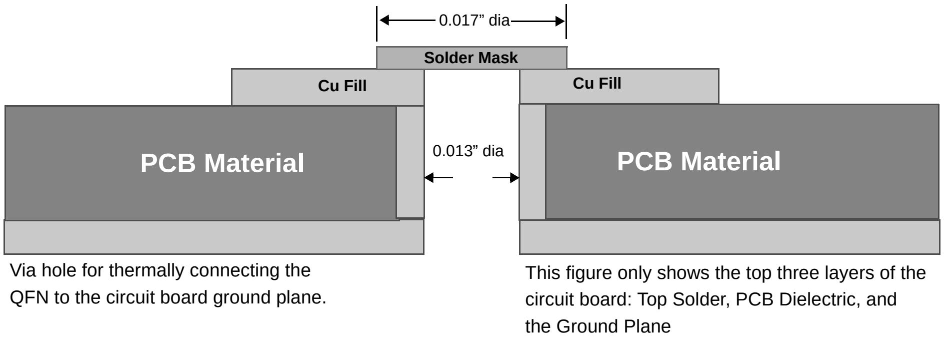
Electrical contact of the part to the Printed Circuit Board (PCB) is made by soldering the leads on the bottom surface of the package to the PCB. Hence, special attention is required to the heat transfer area below the package to provide a good thermal bond to the circuit board. A Copper (Cu) fill is to be designed into the PCB as a thermal pad under the package. Heat is transferred from the FX1 through the device's metal paddle on the bottom side of the package. Heat from here, is conducted to the PCB at the thermal pad. It is then conducted from the thermal pad to the PCB inner ground plane by a 5 x 5 array of via. A via is a plated through hole in the PCB with a finished diameter of 13 mil. The QFN's metal die paddle must be soldered to the PCB's thermal pad. Solder mask is placed on the board top side over each via to resist solder flow into the via. The mask on the top side also minimizes outgassing during the solder reflow process.

For further information on this package design please refer to the application note *Surface Mount Assembly of AMKOR's MicroLeadFrame (MLF) Technology*. This application note can be downloaded from AMKOR's website from the following URL http://www.amkor.com/products/notes_papers/MLF_AppNote_0902.pdf. The application note provides detailed information on board mounting guidelines, soldering flow, rework process, etc.

Figure 29 displays a cross sectional area underneath the package. The cross section is of only one via. The thickness of the solder paste template should be 5 mil. It is recommended that "No Clean" type 3 solder paste is used for mounting the part. Nitrogen purge is recommended during reflow.

Figure 58 is a plot of the solder mask pattern. This pad is thermally connected and is not electrically connected inside the chip. To minimize EMI, this pad should be connected to the ground plane of the circuit board.

Figure 57. Cross Section of the Area Underneath the QFN Package



Via hole for thermally connecting the QFN to the circuit board ground plane.

This figure only shows the top three layers of the circuit board: Top Solder, PCB Dielectric, and the Ground Plane

Figure 58. Plot of the Solder Mask (White Area)