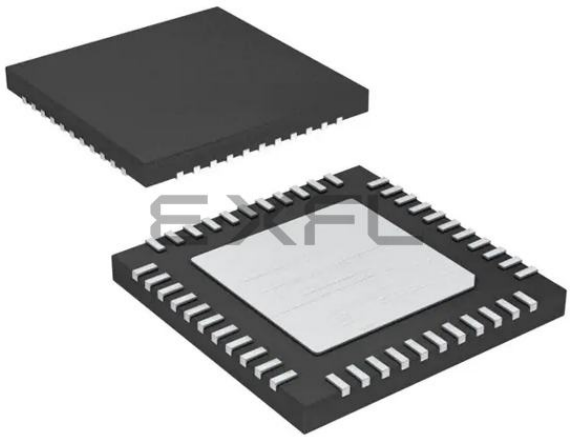


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	60 MIPs
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	35
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 19x12b; D/A 1x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33ep64gs504-e-ml

Logic Block Diagram

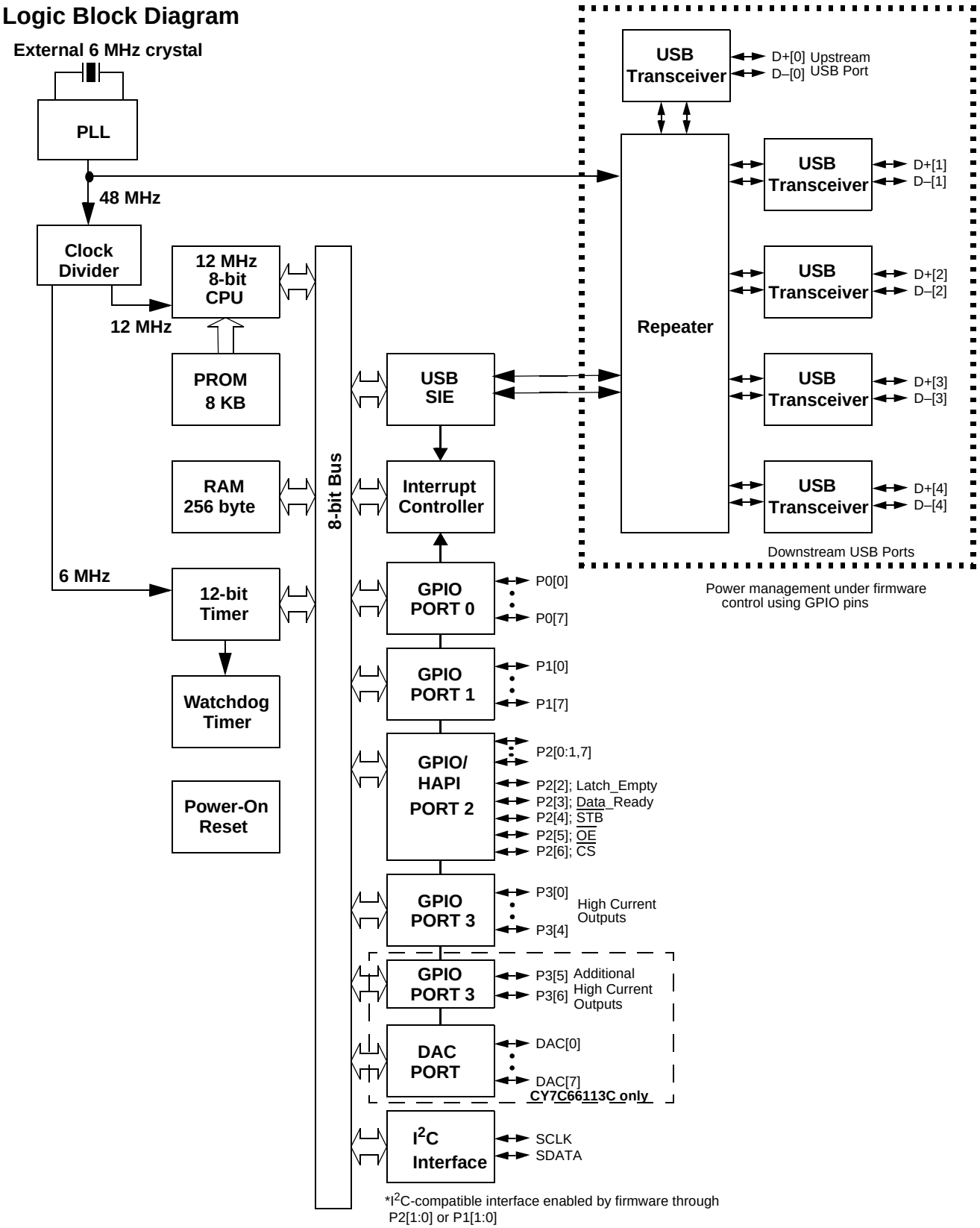


Table 1. Pad Coordinates in Microns (0,0) to Bond Pad Centers

Pad No.	Pin Name	X	Y	Pad #	Pin Name	X	Y
1	XtalOut	1274.2	3588.8	37	DAC6	2000.6	210.6
2	XtalIn	1132.8	3588.8	38	DAC4	2103.6	210.6
3	Vref	889.85	3588.8	39	DAC2	2206.6	210.6
4	Port11b	684.65	3588.8	40	Port06	2308.4	210.6
5	Port13	581.65	3588.8	41	Port04	2411.4	210.6
6	Port15	478.65	3588.8	42	Port02	2514.4	210.6
7	Vss	375.65	3588.8	43	Port00	2617.4	210.6
8	Port17	0	3408.35	44	Vpp	2992.4	25.4
9	Port31	0	3162.05	45	DAC0	2992.4	151.75
10	Du+	0	3060.55	46	Port26	2992.4	306.15
11	Du–	0	2752.4	47	DD+6	2992.4	407.65
12	Port33	0	2650.95	48	DD–6	2992.4	715.75
13	Vss	0	2474.6	49	Port24	2992.4	817.25
14	Port35	0	2368.45	50	Vss	2992.4	923.4
15	DD+1	0	2266.95	51	Port22	2992.4	1086.75
16	DD–1	0	1958.85	52	DD+5	2992.4	1188.25
17	Port37	0	1857.35	53	DD–5	2992.4	1496.35
18	Vref	0	1680.4	54	Port20	2992.4	1597.85
19	Port21	0	1567.4	55	Vref	2992.4	1710.8
20	DD+2	0	1465.95	56	Port36	2992.4	1874.75
21	DD–2	0	1157.85	57	DD+4	2992.4	1976.25
22	Port23	0	1056.35	58	DD–4	2992.4	2284.35
23	Vss	0	880	59	Port34	2992.4	2385.85
24	Port25	0	773.85	60	Vss	2992.4	2492
25	DD+7	0	672.35	61	Port32	2992.4	2655.35
26	DD–7	0	364.25	62	DD+3	2992.4	2756.85
27	Port27	0	262.75	63	DD–3	2992.4	3064.95
28	DAC7	0	100.75	64	Port30	2992.4	3166.45
29	Vss	0	0	65	Port16	2992.4	3412.25
30	Port07	375.2	210.6	66	Port14	2634.2	3588.8
31	Port05	478.2	210.6	67	Port12	2531.2	3588.8
32	Port03	581.2	210.6	68	Port10	2428.2	3588.8
33	Port01	684.2	210.6	69	Port11	2325.2	3588.8
34	DAC5	788.4	210.6	70	VCC	2221.75	3588.8
35	DAC3	891.4	210.6	71	PadOpt	2121.75	3588.8
36	DAC1	994.4	210.6				

Product Summary Tables

Pin Assignments

Table 2. Pin Assignments

Name	I/O	48-Pin	56-Pin QFN	56-Pin SSOP	Description
D+[0], D–[0]	I/O	8, 9	56, 1	8, 9	Upstream port, USB differential data.
D+[1], D–[1]	I/O	12, 13	5, 6	13, 14	Downstream port 1, USB differential data.
D+[2], D–[2]	I/O	15, 16	8, 9	16, 17	Downstream port 2, USB differential data.
D+[3], D–[3]	I/O	40, 41	40, 41	48, 49	Downstream port 3, USB differential data.
D+[4], D–[4]	I/O	35, 36	36, 37	44, 45	Downstream port 4, USB differential data.
P0[7:0]	I/O	21, 25, 22, 26, 23, 27, 24, 28	14, 15, 16, 17, 24, 25, 26, 27	22, 32, 23, 33, 24, 34, 25, 35	GPIO Port 0.
P1[7:0]	I/O	6, 43, 5, 44, 4, 45, 47, 46	52, 53, 54, 43, 44, 45, 46, 47	6, 51, 5, 52, 4, 53, 55, 54	GPIO Port 1.
P2[7:0]	I/O	19, 30, 18, 31, 17, 33, 14, 34	7, 10, 11, 12, 30, 31, 33, 34	20, 38, 19, 39, 18, 41, 15, 42	GPIO Port 2.
P3[6:0]	I/O	37, 10, 39, 7, 42	55, 2, 4, 35, 38, 39, 42,	43, 12, 46, 10, 47, 7, 50	GPIO Port 3, capable of sinking 12 mA (typical).
DAC[7:0]	I/O	n/a	13, 18, 19, 20, 21, 22, 23, 29	21, 29, 26, 30, 27, 31, 28, 37	Digital to Analog Converter (DAC) Port with programmable current sink outputs. DAC[1:0] offer a programmable range of 3.2 to 16 mA typical. DAC[7:2] have a programmable sink current range of 0.2 to 1.0 mA typical.
$XTAL_{IN}$	IN	2	50	2	6 MHz crystal or external clock input.
$XTAL_{OUT}$	OUT	1	49	1	6 MHz crystal out.
V_{PP}		29	28	36	Programming voltage supply, tie to ground during normal operation.
V_{CC}		48	48	56	Voltage supply.
GND		11, 20, 32, 38	3, 32	11, 40	Ground.
V_{REF}	IN	3	51	3	External 3.3V supply voltage for the differential data output buffers and the D+ pull up.

I/O Register Summary

I/O registers are accessed via the I/O Read (IORD) and I/O Write (IOWR, IOWX) instructions. IORD reads data from the selected port into the accumulator. IOWR performs the reverse; it writes data from the accumulator to the selected port. Indexed I/O Write (IOWX) adds the contents of X to the address in the instruction to form the port address and writes data from the accumulator to the specified port. Specifying address 0 such as IOWX 0h indicates the I/O register is selected solely by the contents of X.

All undefined registers are reserved. It is important not to write to reserved registers as this may cause an undefined operation or increased current consumption during operation. When writing to registers with reserved bits, the reserved bits must be written with ‘0.’

Table 3. I/O Register Summary

Register Name	I/O Address	Read/Write	Function	Page
Port 0 Data	0x00	R/W	GPIO Port 0 Data	16
Port 1 Data	0x01	R/W	GPIO Port 1 Data	16
Port 2 Data	0x02	R/W	GPIO Port 2 Data	16
Port 3 Data	0x03	R/W	GPIO Port 3 Data	16
Port 0 Interrupt Enable	0x04	W	Interrupt Enable for Pins in Port 0	18
Port 1 Interrupt Enable	0x05	W	Interrupt Enable for Pins in Port 1	18
Port 2 Interrupt Enable	0x06	W	Interrupt Enable for Pins in Port 2	18
Port 3 Interrupt Enable	0x07	W	Interrupt Enable for Pins in Port 3	18
GPIO Configuration	0x08	R/W	GPIO Port Configurations	17

Address Modes

The CY7C66013C and CY7C66113C microcontrollers support three addressing modes for instructions that require data operands: data, direct, and indexed.

Data (Immediate)

“Data” address mode refers to a data operand that is actually a constant encoded in the instruction. As an example, consider the instruction that loads A with the constant 0xD8:

```
MOV A, 0D8h.
```

This instruction requires two bytes of code where the first byte identifies the “MOV A” instruction with a data operand as the second byte. The second byte of the instruction is the constant “0xD8”. A constant may be referred to by name if a prior “EQU” statement assigns the constant value to the name. For example, the following code is equivalent to the example described earlier:

```
DSPINIT: EQU 0D8h
MOV A, DSPINIT.
```

Direct

“Direct” address mode is used when the data operand is a variable stored in SRAM. In that case, the one byte address of the variable is encoded in the instruction. As an example, consider an instruction that loads A with the contents of memory address location 0x10:

```
MOV A, [10h].
```

Normally, variable names are assigned to variable addresses using “EQU” statements to improve the readability of the assembler source code. As an example, the following code is equivalent to the example described earlier:

```
buttons: EQU 10h
MOV A, [buttons].
```

Indexed

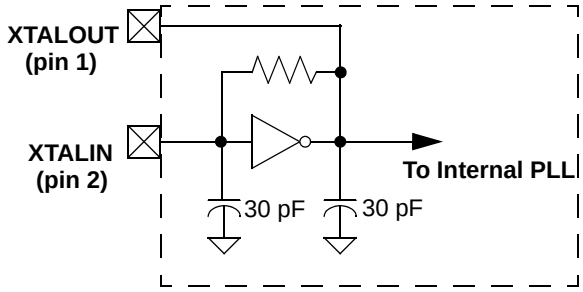
“Indexed” address mode allows the firmware to manipulate arrays of data stored in SRAM. The address of the data operand is the sum of a constant encoded in the instruction and the contents of the “X” register. Normally, the constant is the “base” address of an array of data and the X register contains an index that indicates which element of the array is actually addressed:

```
array: EQU 10h
MOV X, 3
MOV A, [X+array].
```

This has the effect of loading A with the fourth element of the SRAM “array” that begins at address 0x10. The fourth element would be at address 0x13.

Clocking

Figure 5. Clock Oscillator On-Chip Circuit



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

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

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.

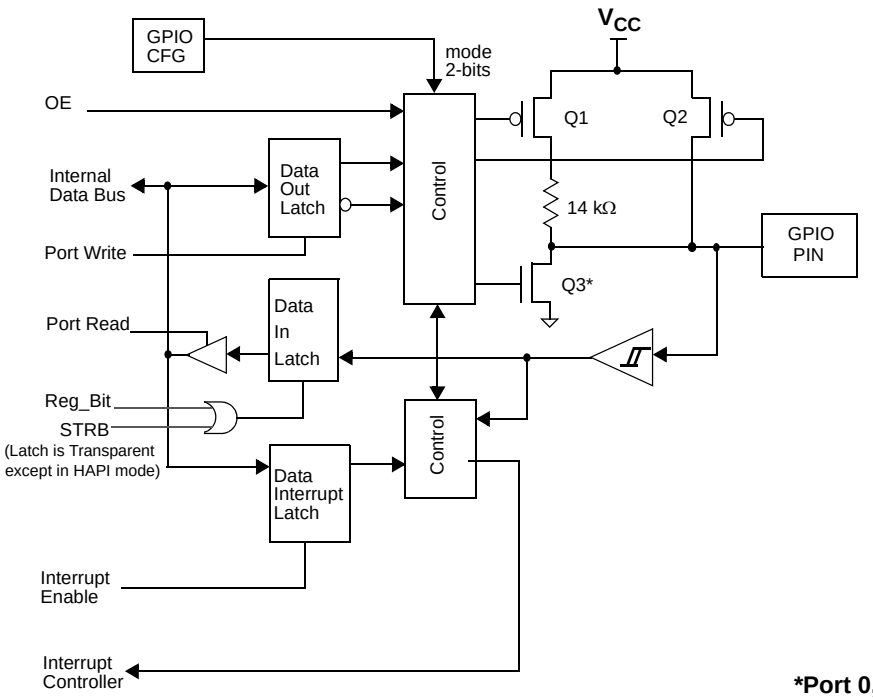
The GPIO interrupt allows the controller to wake up periodically and poll system components while maintaining a very low average power consumption. To achieve the lowest possible current during suspend mode, all I/O should be held at V_{CC} or Gnd. This also applies to internal port pins that may not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

Figure 8. Port 0 Data

Port 0 Data **ADDRESS 0x00**

Bit #	7	6	5	4	3	2	1	0
Bit Name	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
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. Port1 Data

Port 1Data **ADDRESS 0x01**

Bit #	7	6	5	4	3	2	1	0
Bit Name	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
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. Port 2 Data

Port 2 Data **ADDRESS 0x02**

Bit #	7	6	5	4	3	2	1	0
Bit Name	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
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 11. Port 3 Data

Port 3 Data **ADDRESS 0x03**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Reserved	P3.6 CY7C66113C only	P3.5 CY7C66113C only	P3.4	P3.3	P3.2	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

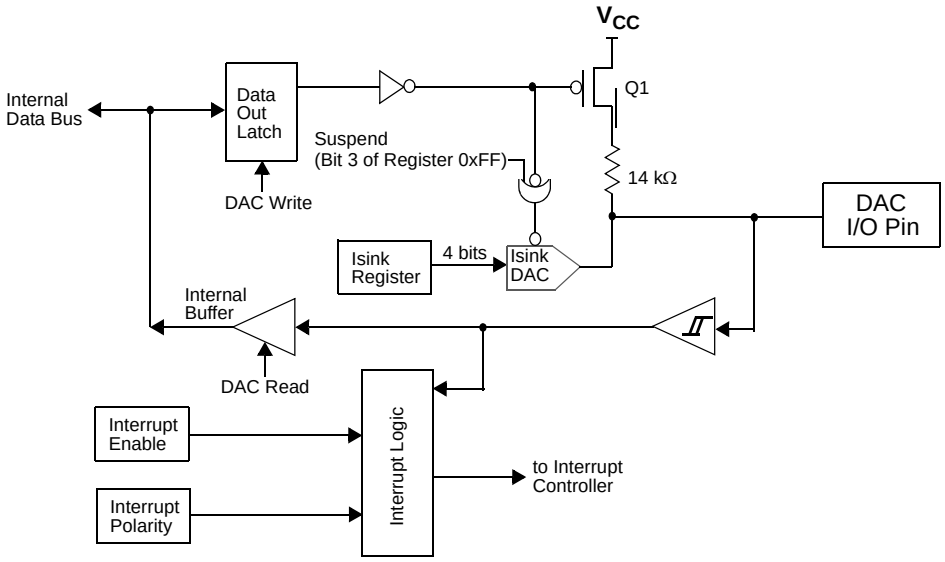
Special care should be taken with any unused GPIO data bits. An unused GPIO data bit, either a pin on the chip or a port bit that is not bonded on a particular package, must not be left floating when the device enters the suspend state. If a GPIO data bit is left floating, the leakage current caused 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 CY7C66013C always requires that P3[7:5] be written with a '0.' When the CY7C66113C is used the P3[7] should 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 Hardware Assisted Parallel Interface (HAPI)). 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.

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.

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].

Figure 19. DAC Sink Register

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

Bit [3..0]: Isink [x] (x= 0..3)

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..4]: Reserved

DAC Port Interrupts

A DAC port interrupt is enabled or 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 DAC Interrupt on page 30.

Figure 20. DAC Port Interrupt Enable

DAC Port Interrupt **ADDRESS 0x31**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Enable Bit 7	Enable Bit 6	Enable Bit 5	Enable Bit 4	Enable Bit 3	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

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

Figure 21. DAC Port Interrupt Polarity

DAC I/O Interrupt Polarity **ADDRESS 0x32**

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

Bit [7..0]: Polarity bit x (x= 0..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).

Processor Status and Control Register

Figure 28. Processor Status and Control Register

Processor Status and Control **ADDRESS 0xFF**

Bit #	7	6	5	4	3	2	1	0
Bit Name	IRQ Pending	Watchdog Reset	USB Bus Reset Interrupt	Power On Reset	Suspend	Interrupt Enable Sense	Reserved	Run
Read/Write	R	R/W	R/W	R/W	R/W	R	R/W	R/W
Reset	0	0	0	1	0	0	0	1

Bit 0: Run

This bit is manipulated by the HALT instruction. When Halt is executed, all the bits of the Processor Status and Control Register are cleared to 0. Since the run bit is cleared, the processor stops at the end of the current instruction. The processor remains halted until an appropriate reset occurs (power on or Watchdog). This bit should normally be written as a '1.'

Bit 1: Reserved

Bit 1 is reserved and must be written as a zero.

Bit 2: Interrupt Enable Sense

This bit indicates whether interrupts are enabled or disabled. Firmware has no direct control over this bit as writing a zero or one to this bit position has no effect on interrupts. A '0' indicates that interrupts are masked off and a '1' indicates that the interrupts are enabled. This bit is further gated with the bit settings of the Global Interrupt Enable Register (Figure 29) and USB End Point Interrupt Enable Register (Figure 30). Instructions DI, EI, and RETI manipulate the state of this bit.

Bit 3: Suspend

Writing a '1' to the Suspend bit halts the processor and cause the microcontroller to enter the suspend mode that significantly reduces power consumption. A pending, enabled interrupt or USB bus activity causes the device to come out of suspend. After coming out of suspend, the device resumes firmware execution at the instruction following the IOWR which put the part into suspend. An IOWR attempting to put the part into suspend is ignored if USB bus activity is present. See Suspend Mode on page 15 for more details on suspend mode operation.

Bit 4: Power on Reset

The POR is set to '1' during a power on reset. The firmware checks bits 4 and 6 in the reset handler to determine whether a reset was caused by a power on condition or a Watchdog timeout. A POR event may be followed by a WDR before firmware begins executing, as explained here.

Bit 5: USB Bus Reset Interrupt

The USB Bus Reset Interrupt bit is set when the USB Bus Reset is detected on receiving a USB Bus Reset signal on the upstream port. The USB Bus Reset signal is a single ended zero (SE0) that lasts from 12 to 16 µs. An SE0 is defined as the condition in which both the D+ line and the D– line are LOW at the same time.

Bit 6: WDR

The WDR is set during a reset initiated by the WDT. This indicates the WDT went for more than t_{WATCH} (8 ms minimum) between Watchdog clears. This occurs with a POR event.

Bit 7: IRQ Pending

The IRQ pending, when set, indicates that one or more of the interrupts is recognized as active. An interrupt remains pending until its interrupt enable bit is set (Figure 29, Figure 30) and interrupts are globally enabled. At that point, the internal interrupt handling sequence clears this bit until another interrupt is detected as pending.

During power up, the Processor Status and Control Register is set to 00010001, which indicates a POR (bit 4 set) has occurred and no interrupts are pending (bit 7 clear). During the 96 ms suspend at start up (explained in Power on Reset on page 14), a WDR also occurs unless this suspend is aborted by an upstream SE0 before 8 ms. If a WDR occurs during the power up suspend interval, firmware reads 01010001 from the Status and Control Register after power up. Normally, the POR bit should be cleared so a subsequent WDR is clearly identified. If an upstream bus reset is received before firmware examines this register, the Bus Reset bit may also be set.

During a WDR, the Processor Status and Control Register is set to 01XX0001, which indicates a WDR (bit 6 set) has occurred and no interrupts are pending (bit 7 clear). The WDR does not effect the state of the POR and the Bus Reset Interrupt bits.

Interrupts

Interrupts are generated by the GPIO and DAC pins, the internal timers, I²C compatible or HAPI operation, the internal USB hub, or on various USB traffic conditions. All interrupts are maskable by the Global Interrupt Enable Register and the USB End Point Interrupt Enable Register. Writing a '1' to a bit position enables the interrupt associated with that bit position.

Figure 29. Global Interrupt Enable Register

Global Interrupt Enable Register **ADDRESS 0X20**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Reserved	I²C Interrupt Enable	GPIO Interrupt Enable	DAC Interrupt Enable	USB Hub Interrupt Enable	1.024 ms Interrupt Enable	128 µs Interrupt Enable	USB Bus RST Interrupt Enable
Read/Write	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	0	0	0	0	0	0	0

Bit 0: USB Bus RST Interrupt Enable

1 = Enable Interrupt on a USB Bus Reset; 0 = Disable interrupt on a USB Bus Reset (refer to USB Bus Reset Interrupt).

Bit 1: 128 µs Interrupt Enable

1 = Enable Timer interrupt every 128 µs; 0 = Disable Timer Interrupt for every 128 µs.

Bit 2: 1.024 ms Interrupt Enable

1= Enable Timer interrupt every 1.024 ms; 0 = Disable Timer Interrupt every 1.024 ms.

Bit 3: USB Hub Interrupt Enable

1 = Enable Interrupt on a Hub status change; 0 = Disable interrupt due to hub status change. (Refer to USB Hub Interrupt.)

Bit 4: DAC Interrupt Enable

1 = Enable DAC Interrupt; 0 = Disable DAC interrupt.

Bit 5: GPIO Interrupt Enable

1 = Enable Interrupt on falling and rising edge on any GPIO; 0 = Disable Interrupt on falling and rising edge on any GPIO. (Refer to sections GPIO and HAPI Interrupt, GPIO Configuration Port, and GPIO Interrupt Enable Ports.)

Bit 6: I²C Interrupt Enable

1 = Enable Interrupt on I2C related activity; 0 = Disable I2C related activity interrupt. (Refer to I²C Interrupt.)

Bit 7: Reserved.

Figure 30. USB Endpoint Interrupt Enable Register

USB Endpoint Interrupt Enable **ADDRESS 0X21**

Bit #	7	6	5	4	3	2	1	0
Bit Name	Reserved	Reserved	Reserved	EPB1 Interrupt Enable	EPB0 Interrupt Enable	EPA2 Interrupt Enable	EPA1 Interrupt Enable	EPA0 Interrupt Enable
Read/Write	-	-	-	R/W	R/W	R/W	R/W	R/W
Reset	-	-	-	0	0	0	0	0

Bit 0: EPA0 Interrupt Enable

1 = Enable Interrupt on data activity through endpoint A0;
0 = Disable Interrupt on data activity through endpoint A0.

Bit 1: EPA1 Interrupt Enable

1 = Enable Interrupt on data activity through endpoint A1;
0 = Disable Interrupt on data activity through endpoint A1.

Bit 2: EPA2 Interrupt Enable

1 = Enable Interrupt on data activity through endpoint A2;
0 = Disable Interrupt on data activity through endpoint A2.

Bit 3: EPB0 Interrupt Enable

1 = Enable Interrupt on data activity through endpoint B0;
0 = Disable Interrupt on data activity through endpoint B0.

Bit 4: EPB1 Interrupt Enable

1 = Enable Interrupt on data activity through endpoint B1;
0 = Disable Interrupt on data activity through endpoint B1.

Bit [7..5]: Reserved

During a reset, the contents the Global Interrupt Enable Register and USB End Point Interrupt Enable Register are cleared, effectively, disabling all interrupts.

The interrupt controller contains a separate flip flop for each interrupt. See Figure 31 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

DAC Interrupt

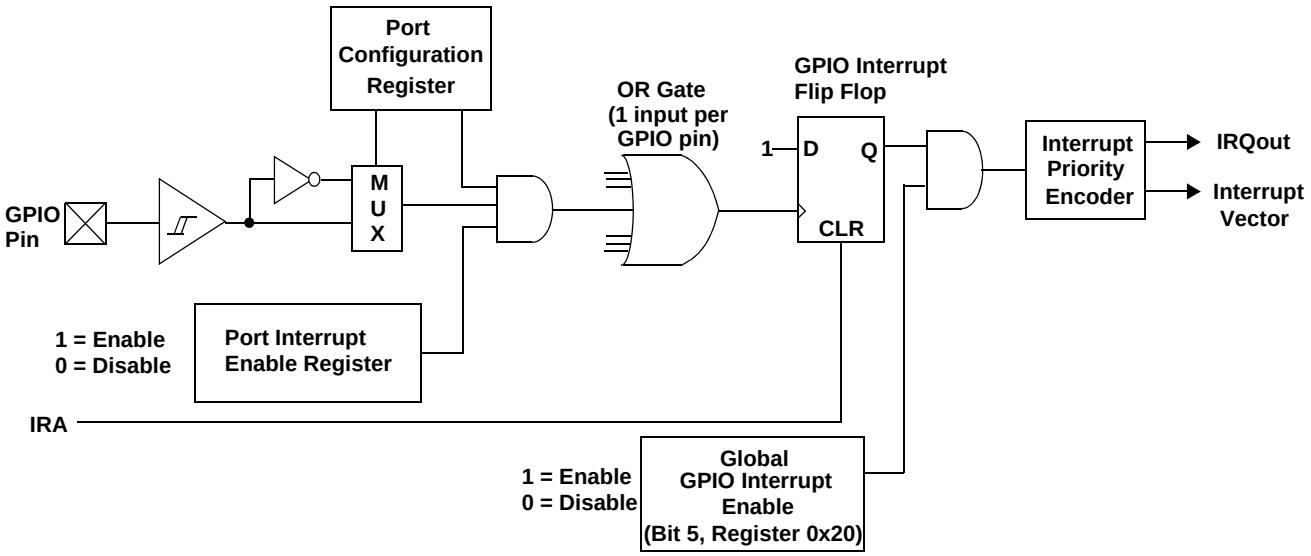
Each DAC I/O pin generates an interrupt, if enabled. The interrupt polarity for each DAC I/O pin is programmable. A positive polarity is a rising edge input while a negative polarity is a falling edge input. All of the DAC pins share a single interrupt vector, which means the firmware needs to read the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

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

GPIO and HAPI Interrupt

Each of the GPIO pins generates an interrupt, if enabled. The interrupt polarity is programmed for each GPIO port as part of the GPIO configuration. All of the GPIO pins share a single interrupt vector, which means the firmware needs to read the GPIO ports with enabled interrupts to determine which pin or pins caused an interrupt. A block diagram of the GPIO interrupt logic is shown in Figure 32. Refer to GPIO Configuration Port and GPIO Interrupt Enable Ports for more information about setting GPIO interrupt polarity and enabling individual GPIO interrupts.

Figure 32. GPIO Interrupt Structure



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

When HAPI is enabled, the HAPI logic takes over the interrupt vector and blocks any interrupt from the GPIO bits, including ports and bits not used by HAPI. Operation of the HAPI interrupt is independent of the GPIO specific bit interrupt enables, and is enabled or disabled only by bit 5 of the Global Interrupt Enable Register (0x20) when HAPI is enabled. The settings of the GPIO bit interrupt enables on ports and bits not used by HAPI still effect the CMOS mode operation of those ports and bits. The effect of modifying the interrupt bits while the Port Config bits are set to '10' is shown in Table 6. The events that generate HAPI interrupts are described in Hardware Assisted Parallel Interface (HAPI).

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 SIE Operation

The CY7C66x13C SIE supports operation as a single device or a compound device. This section describes the two device addresses, the configurable endpoints, and the endpoint function.

USB Device Addresses

The USB Controller provides two USB Device Address Registers: A (addressed at 0x10)and B (addressed at 0x40). Upon reset and under default conditions, Device A has three endpoints and Device B has two endpoints. The USB Device Address Register contents are cleared during a reset, setting the USB device addresses to zero and disabling these addresses. Figure 43 shows the format of the USB Address Registers.

Figure 43. USB Device Address Registers

USB Device Address (Device A, B) **ADDRESSES 0x10(A) and 0x40(B)**

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

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 responds to USB traffic to the Device Address.

USB Device Endpoints

The CY7C66x13C controller supports up to two addresses 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 (see Figure 42). Bit 7 controls the size of the endpoints and bit 6 controls the number of addresses. These configuration options are detailed in Table 14. Endpoint FIFOs are part of user RAM (as shown in Data Memory Organization on page 12).

Table 14. Memory Allocation for Endpoints

USB Status And Control Register (0x1F) Bits [7, 6]											
[0,0]			[1,0]			[0,1]			[1,1]		
Two USB Addresses: A (3 Endpoints) & B (2 Endpoints)			Two USB Addresses: A (3 Endpoints) &B (2 Endpoints)			One USB Address: A (5 Endpoints)			One USB Address: A (5 Endpoints)		
Label	Start Address	Size	Label	Start Address	Size	Label	Start Address	Size	Label	Start Address	Size
EPB1	0xD8	8	EPB0	0xA8	8	EPA4	0xD8	8	EPA3	0xA8	8
EPB0	0xE0	8	EPB1	0xB0	8	EPA3	0xE0	8	EPA4	0xB0	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).

USB Control Endpoint Mode Registers

All USB devices are required to have a control endpoint 0 (EPA0 and EPB0) 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 registers are cleared during reset. When USB Status And Control Register Bits [6,7] are set to [0,0] or [1,0], the endpoint 0 EPA0 and EPB0 mode registers use the format shown in Figure 44.

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 I2C 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.

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

SETUP (if accepting SETUPs)																									
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
See Table 11				Setup	<= 10	data	valid	updates	1	updates	1	UC	UC	1	0	0	0	1	ACK	yes
See Table 11				Setup	> 10	junk	x	updates	updates	updates	1	UC	UC	UC	No Change				ignore	yes
See Table 11				Setup	x	junk	invalid	updates	0	updates	1	UC	UC	UC	No Change				ignore	yes
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
DISABLED																									
0	0	0	0	x	x	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
Nak In/Out																									
0	0	0	1	Out	x	UC	x	UC	UC	UC	UC	UC	1	UC	No Change				NAK	yes
0	0	0	1	In	x	UC	x	UC	UC	UC	UC	1	UC	UC	No Change				NAK	yes
Ignore In/Out																									
0	1	0	0	Out	x	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
0	1	0	0	In	x	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
Stall In/Out																									
0	0	1	1	Out	x	UC	x	UC	UC	UC	UC	UC	1	UC	No Change				Stall	yes
0	0	1	1	In	x	UC	x	UC	UC	UC	UC	1	UC	UC	No Change				Stall	yes
CONTROL WRITE																									
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/premature status In																									
1	0	1	1	Out	<= 10	data	valid	updates	1	updates	UC	UC	1	1	1	0	1	0	ACK	yes
1	0	1	1	Out	> 10	junk	x	updates	updates	updates	UC	UC	1	UC	No Change				ignore	yes
1	0	1	1	Out	x	junk	invalid	updates	0	updates	UC	UC	1	UC	No Change				ignore	yes
1	0	1	1	In	x	UC	x	UC	UC	UC	UC	1	UC	1	No Change				TX 0	yes
NAK Out/premature status In																									
1	0	1	0	Out	<= 10	UC	valid	UC	UC	UC	UC	UC	1	UC	No Change				NAK	yes
1	0	1	0	Out	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
1	0	1	0	Out	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
1	0	1	0	In	x	UC	x	UC	UC	UC	UC	1	UC	1	No Change				TX 0	yes
Status In/extra Out																									
0	1	1	0	Out	<= 10	UC	valid	UC	UC	UC	UC	UC	1	UC	0	0	1	1	Stall	yes
0	1	1	0	Out	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
0	1	1	0	Out	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
0	1	1	0	In	x	UC	x	UC	UC	UC	UC	1	UC	1	No Change				TX 0	yes
CONTROL READ																									
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/premature status Out																									
1	1	1	1	Out	2	UC	valid	1	1	updates	UC	UC	1	1	No Change				ACK	yes
1	1	1	1	Out	2	UC	valid	0	1	updates	UC	UC	1	UC	0	0	1	1	Stall	yes
1	1	1	1	Out	!=2	UC	valid	updates	1	updates	UC	UC	1	UC	0	0	1	1	Stall	yes
1	1	1	1	Out	> 10	UC	x	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
1	1	1	1	Out	x	UC	invalid	UC	UC	UC	UC	UC	UC	UC	No Change				ignore	no
1	1	1	1	In	x	UC	x	UC	UC	UC	UC	1	UC	1	1	1	1	0	ACK (back)	yes

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	I²C 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
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
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

Absolute Maximum Ratings

Exceeding maximum ratings may impair the useful life of the device. These user guidelines are not tested.

Storage Temperature ..... –65°C to +150°C

Ambient Temperature with Power Applied..... 0°C to +70°C

Supply Voltage on V_{CC} relative to V_{SS} .....–0.5V to +7.0V

DC Input Voltage ..... –0.5V to +V_{CC} +0.5V

DC Voltage Applied to Outputs in High-Z State –0.5V to +V_{CC} +0.5V

Power Dissipation..... 500 mW

Static Discharge Voltage .....> 2000V

Latch Up Current ..... > 200 mA

Max Output Sink Current into Port 0, 1, 2, 3, and DAC[1:0] Pins 60 mA

Max Output Sink Current into DAC[7:2] Pins..... 10 mA

Max Output Source Current from Port 1, 2, 3, 4 ..... 30 mA

Electrical Characteristics (Fosc = 6 MHz; Operating Temperature = 0 to 70°C, V_{CC} = 4.0V to 5.25V)

Parameter	Description	Conditions	Min	Max	Unit
General					
V_{REF}	Reference Voltage	3.3V ±5%	3.15	3.45	V
V_{pp}	Programming Voltage (disabled)		–0.4	0.4	V
I_{CC}	V_{CC} Operating Current	No GPIO source current		50	mA
I_{SB1}	Supply Current—Suspend Mode			50	µA
I_{ref}	Vref Operating Current	No USB Traffic[9]		10	mA
I_{il}	Input Leakage Current	Any pin		1	µA
USB Interface					
V_{di}	Differential Input Sensitivity	| (D+)–(D–) |	0.2		V
V_{cm}	Differential Input Common Mode Range		0.8	2.5	V
V_{se}	Single Ended Receiver Threshold		0.8	2.0	V
C_{in}	Transceiver Capacitance			20	pF
I_{lo}	Hi-Z State Data Line Leakage	0V < V_{in} < 3.3V	–10	10	µA
R_{ext}	External USB Series Resistor	In series with each USB pin	19	21	Ω
R_{UUP}	External Upstream USB pull up Resistor	1.5 kΩ ±5%, D+ to V_{REG}	1.425	1.575	kΩ
R_{UDN}	External Downstream Pull down Resistors	15 kΩ ±5%, downstream USB pins	14.25	15.75	kΩ
Power-on Reset					
t_{vccs}	V_{CC} Ramp Rate	Linear ramp 0V to V_{CC}[10]	0	100	ms
USB Upstream/Downstream Port					
V_{UOH}	Static Output High	15 kΩ ±5% to Gnd	2.8	3.6	V
V_{UOL}	Static Output Low	1.5 kΩ ±5% to V_{REF}		0.3	V
Z_O	USB Driver Output Impedance	Including R_{ext} Resistor	28	44	Ω
General Purpose I/O (GPIO)					
R_{up}	pull up Resistance (typical 14 kΩ)		8.0	24.0	kΩ
V_{ITH}	Input Threshold Voltage	All ports, LOW to HIGH edge	20%	40%	V_{CC}
V_H	Input Hysteresis Voltage	All ports, HIGH to LOW edge	2%	8%	V_{CC}
VOL	Port 0,1,2,3 Output Low Voltage	I_{OL} = 3 mA I_{OL} = 8 mA		0.4 2.0	V V

Notes

9. Add 18 mA per driven USB cable (upstream or downstream). This is based on transitions every two full speed bit times on average.
10. Power on Reset occurs whenever the voltage on V_{CC} is below approximately 2.5V.

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

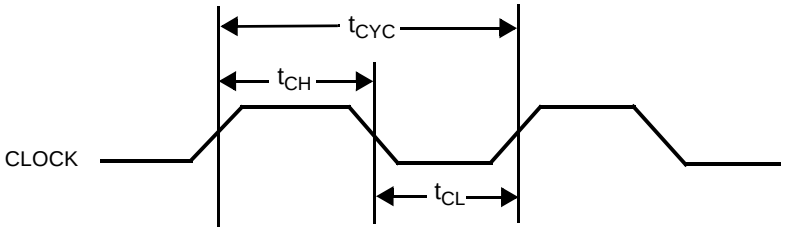


Figure 50. USB Data Signal Timing

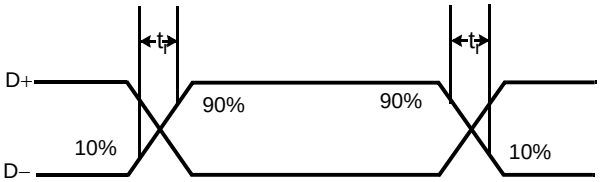
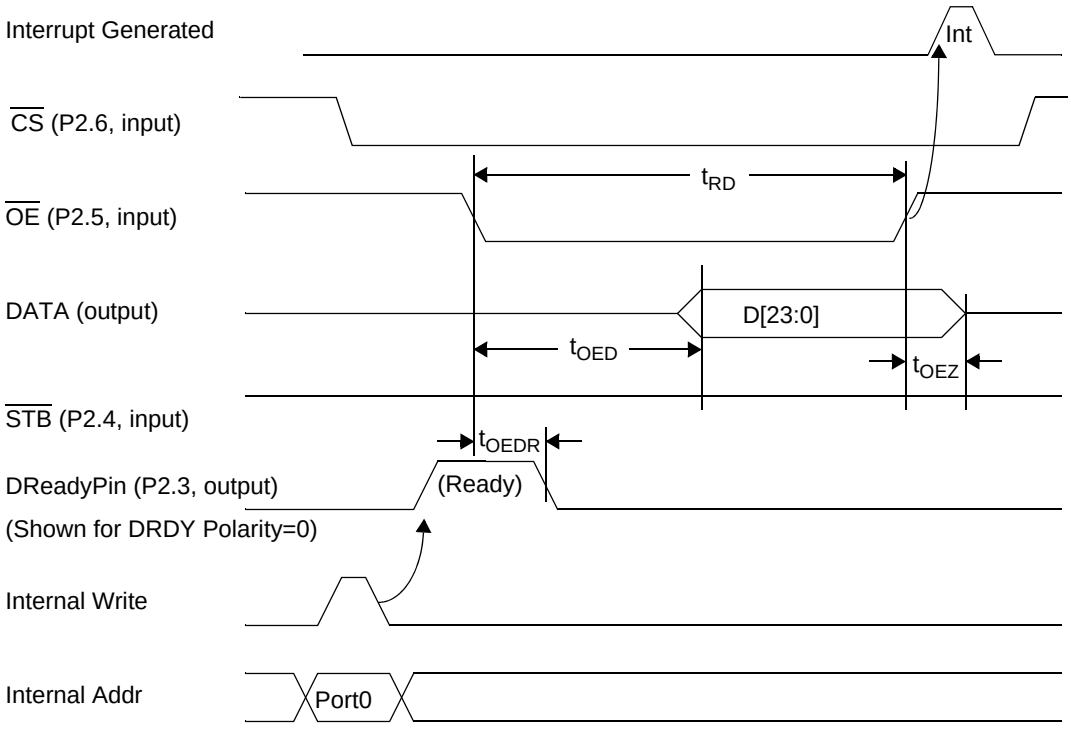


Figure 51. HAPI Read by External Interface from USB Microcontroller



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

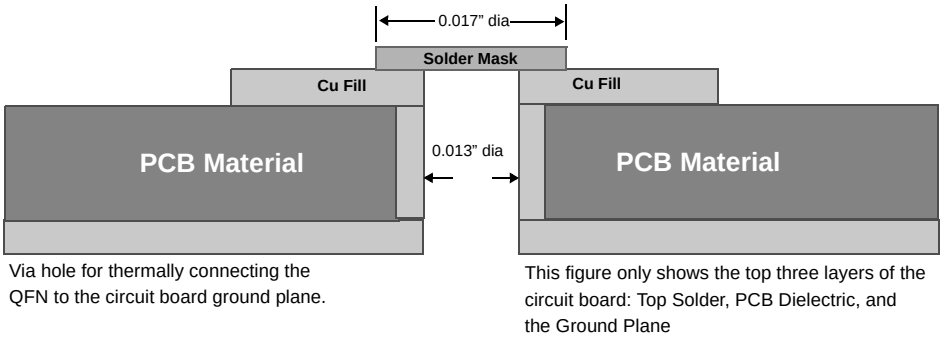


Figure 58. Plot of the Solder Mask (White Area)