Infineon Technologies - CY7C66113C-LTXCT 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?

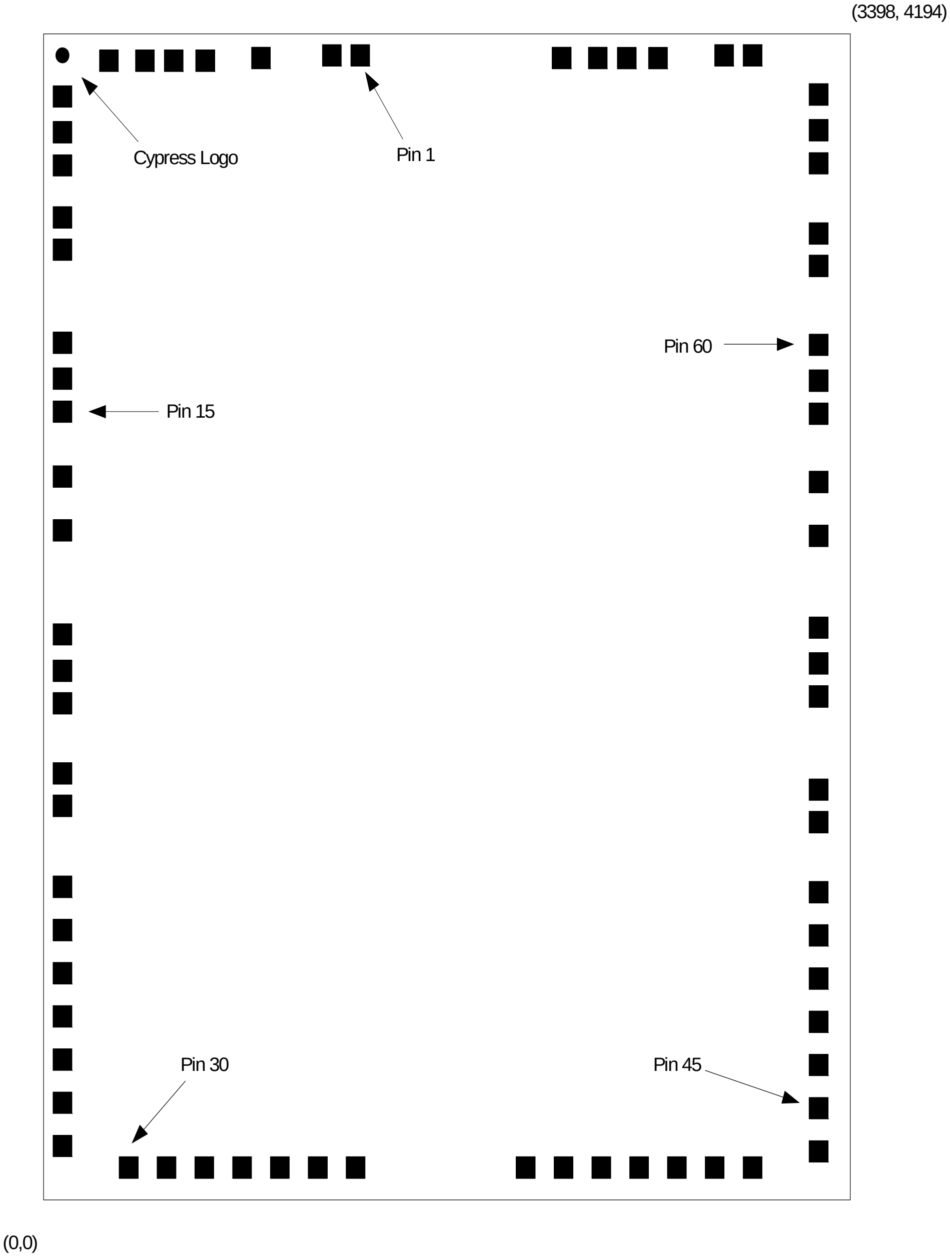
Application specific microcontrollers are engineered to

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-ltxct

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Figure 3. CY7C66113C DIE



DIE STEP: 3398 x 4194 microns
Die Size: 3322 x 4129 microns
Die Thickness: 14 mils = 355.6 microns
Pad Size: 80 x 80 microns

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

8-Bit Accumulator (A)

The accumulator is the general purpose register for the microcontroller.

8-Bit Temporary Register (X)

The “X” register is available to the firmware for temporary storage of intermediate results. The microcontroller performs indexed operations based on the value in X. Refer to the section, Indexed on page 13 for additional information.

8-Bit Program Stack Pointer (PSP)

During a reset, the Program Stack Pointer (PSP) is set to 0x00 and “grows” upward from this address. The PSP may be set by firmware, using the MOV PSP,A instruction. The PSP supports interrupt service under hardware control and CALL, RET, and RETI instructions under firmware control. The PSP is not readable by the firmware.

During an interrupt acknowledge, interrupts are disabled and the 14-bit program counter, carry flag, and zero flag are written as two bytes of data memory. The first byte is stored in the memory addressed by the PSP, then the PSP is incremented. The second byte is stored in memory addressed by the PSP, and the PSP is incremented again. The overall effect is to store the program counter and flags on the program “stack” and increment the PSP by two.

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

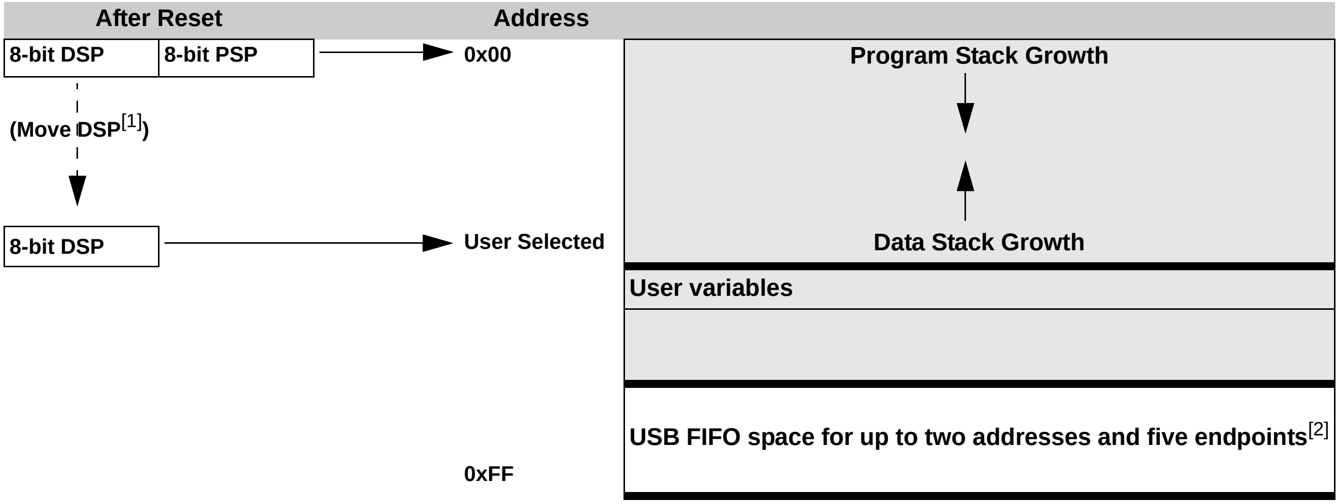
The Call Subroutine (CALL) instruction stores the program counter and flags on the program stack and increments the PSP by two.

The Return From Subroutine (RET) instruction restores the program counter but not the flags from the program stack and decrements the PSP by two.

Data Memory Organization

The CY7C66x13C microcontrollers provide 256 bytes of data RAM. Normally, the SRAM is partitioned into four areas: program stack, user variables, data stack, and USB endpoint FIFOs. The following is one example of where the program stack, data stack, and user variables areas are located.

Table 5. SRAM Areas



8-Bit Data Stack Pointer (DSP)

The Data Stack Pointer (DSP) supports PUSH and POP instructions that use the data stack for temporary storage. A PUSH instruction pre-decrements the DSP, then writes data to the memory location addressed by the DSP. A POP instruction reads data from the memory location addressed by the DSP, then post-increments the DSP.

During a reset, the DSP is reset to 0x00. A PUSH instruction when DSP equals 0x00 writes data at the top of the data RAM (address 0xFF). This writes data to the memory area reserved for USB endpoint FIFOs. Therefore, the DSP should be indexed at an appropriate memory location that does not compromise the Program Stack, user defined memory (variables), or the USB endpoint FIFOs.

For USB applications, the firmware should set the DSP to an appropriate location to avoid a memory conflict with RAM dedicated to USB FIFOs. The memory requirements for the USB endpoints are described in USB Device Endpoints on page 38. Example assembly instructions to do this with two device addresses (FIFOs begin at 0xD8) are shown:

- MOV A,20h; Move 20 hex into Accumulator (must be D8h or less)
- SWAP A,DSP; swap accumulator value into DSP register.

Notes

1. Refer to 8-Bit Data Stack Pointer (DSP) for a description of DSP.
2. Endpoint sizes are fixed by the Endpoint Size Bit (I/O register 0x1F, Bit 7), see Table 14.

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.

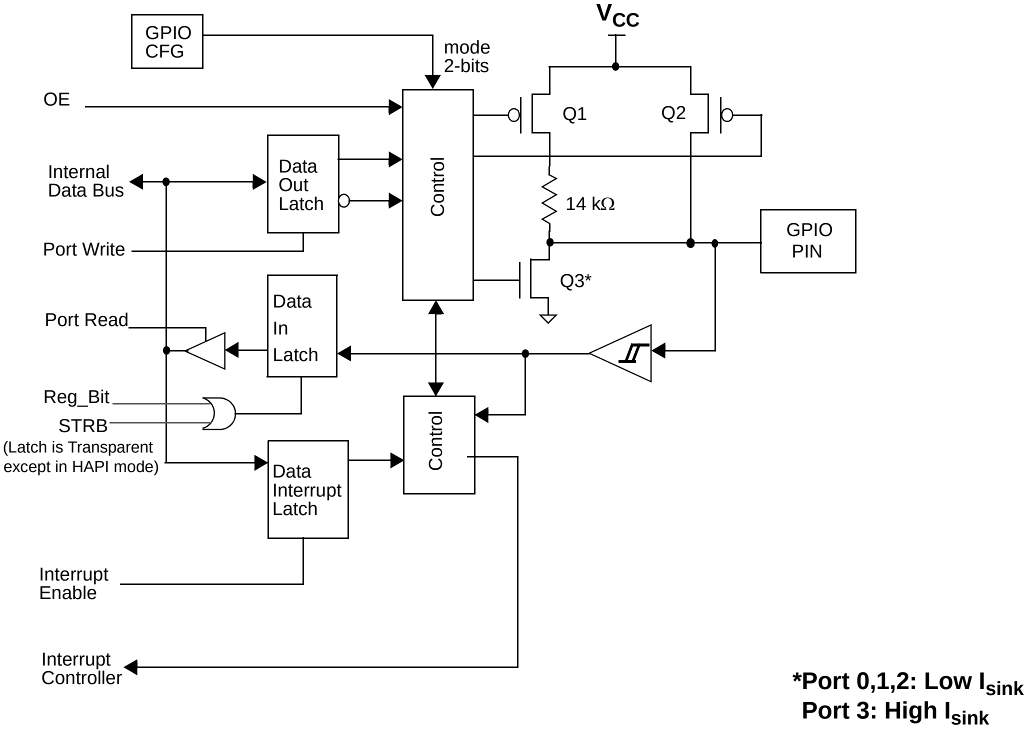
The GPIO interrupt allows the controller to wake up periodically and poll system components while maintaining a very low average power consumption. To achieve the lowest possible current during suspend mode, all I/O should be held at V_{CC} or Gnd. This also applies to internal port pins that may not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



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.

GPIO Interrupt Enable Ports

Each GPIO pin is individually enabled or disabled as an interrupt source. The Port 0–3 Interrupt Enable registers provide this feature with an interrupt enable bit for each GPIO pin. When HAPI mode is enabled the GPIO interrupts are blocked, including ports not used by HAPI, so GPIO pins are not used as interrupt sources.

During a reset, GPIO interrupts are disabled by clearing all of the GPIO interrupt enable ports. Writing a ‘1’ to a GPIO Interrupt Enable bit enables GPIO interrupts from the corresponding input pin. All GPIO pins share a common interrupt, as discussed in GPIO and HAPI Interrupt.

Figure 13. Port 0 Interrupt Enable

Port 0 Interrupt Enable **ADDRESS 0x04**

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

Figure 14. Port 1 Interrupt Enable

Port 1 Interrupt Enable **ADDRESS 0x05**

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

Figure 15. Port 2 Interrupt Enable

Port 2 Interrupt Enable **ADDRESS 0x06**

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

Figure 16. Port 3 Interrupt Enable

Port 3 Interrupt Enable **ADDRESS 0x07**

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

Figure 26. I^{2}C Data Register

I^{2}C Data **ADDRESS 0x29**

Bit #	7	6	5	4	3	2	1	0
Bit Name	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
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]: I^{2}C Data

Contains 8-bit data on the I^{2}C Bus.

Figure 27. I^{2}C Status and Control Register

I^{2}C 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	I^{2}C 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 I^{2}C Status and Control register bits are defined in Table 9, with a more detailed description following.

Table 9. I^{2}C Status and Control Register Bit Definitions

Bit	Name	Description
0	I^{2}C Enable	When set to ‘1’, the I^{2}C compatible function is enabled. When cleared, I^{2}C GPIO pins operate normally.
1	Received Stop	Reads 1 only in slave receive mode, when I^{2}C 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 I^{2}C 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.

Hardware Assisted Parallel Interface (HAPI)

The CY7C66x13C processor provides a hardware assisted parallel interface for bus widths of 8, 16, or 24 bits, to accommodate data transfer with an external microcontroller or similar device. Control bits for selecting the byte width are in the HAPI and I^{2}C Configuration Register (Figure 25), bits 1 and 0.

Signals are provided on Port 2 to control the HAPI interface. Table 10 describes these signals and the HAPI control bits in the HAPI and I^{2}C Configuration Register. Enabling HAPI causes the GPIO setting in the GPIO Configuration Register (Figure 10) to be overridden. The Port 2 output pins are in CMOS output mode and Port 2 input pins are in input mode (open drain mode with Q3 OFF in Figure 7).

Table 10. Port 2 Pin and HAPI Configuration Bit Definitions

Pin	Name	Direction	Description (Port 2 Pin)
P2[2]	LatEmptyPin	Out	Ready for more input data from external interface.
P2[3]	DReadyPin	Out	Output data ready for external interface.
P2[4]	STB	In	Strobe signal for latching incoming data.
P2[5]	OE	In	Output Enable, causes chip to output data.
P2[6]	CS	In	Chip Select (Gates $\overline{\text{STB}}$ and $\overline{\text{OE}}$).

Bit	Name	R/W	Description (HAPI and I^{2}C Configuration Register)
2	Data Ready	R	Asserted after firmware writes data to Port 0, until $\overline{\text{OE}}$ driven LOW.
3	Latch Empty	R	Asserted after firmware reads data from Port 0, until $\overline{\text{STB}}$ driven LOW.
4	DRDY Polarity	R/W	Determines polarity of Data Ready bit and DReadyPin: If 0, Data Ready is active LOW, DReadyPin is active HIGH. If 1, Data Ready is active HIGH, DReadyPin is active LOW.
5	LEMPTY Polarity	R/W	Determines polarity of Latch Empty bit and LatEmptyPin: If 0, Latch Empty is active LOW, LatEmptyPin is active HIGH. If 1, Latch Empty is active HIGH, LatEmptyPin is active LOW.

HAPI Read by External Device from CY7C66x13C

In this case (see Figure 50), firmware writes data to the GPIO ports. If 16-bit or 24-bit transfers are being made, Port 0 is written last, because writes to Port 0 asserts the Data Ready bit and the DReadyPin to signal the external device that data is available.

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

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

HAPI Write by External Device to CY7C66x13C

In this case (see Figure 52), the external device drives the $\overline{\text{STB}}$ and $\overline{\text{CS}}$ pins active (LOW) when it drives new data onto the port pins. When this happens, the internal latches become full, which causes the Latch Empty bit to be deasserted. When $\overline{\text{STB}}$ is returned HIGH (inactive), the HAPI and GPIO interrupt is generated. Firmware then reads the parallel ports to empty the HAPI latches. If 16-bit or 24-bit transfers are being made, Port 0 should be read last because reads from Port 0 assert the Latch Empty bit and the LatEmptyPin to signal the external device for more data.

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

Interrupts

Interrupts are generated by the GPIO and DAC pins, the internal timers, I^{2}C 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	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 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: I^{2}C Interrupt Enable

1 = Enable Interrupt on I2C related activity; 0 = Disable I2C related activity interrupt. (Refer to I^{2}C 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

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).

Hub Downstream Ports Status and Control

Data transfer on hub downstream ports is controlled according to the bit settings of the Hub Downstream Ports Control Register (Figure 36). Each downstream port is controlled by two bits, as defined in Table 12. The Hub Downstream Ports Control Register is cleared upon reset or bus reset, and the reset state is the state for normal USB traffic. Any downstream port being forced must be marked as disabled (Figure 35) for proper operation of the hub repeater.

Firmware uses this register for driving bus reset and resume signaling to downstream ports. Controlling the port pins through this register uses standard USB edge rate control according to the speed of the port, set in the Hub Port Speed Register.

The downstream USB ports are designed for connection of USB devices, but also serves as output ports under firmware control. This allows unused USB ports to be used for functions such as driving LEDs or providing additional input signals. Pulling up these pins to voltages above V_{REF} may cause current flow into the pin.

This register is not reset by bus reset. These bits must be cleared before going into suspend.

Figure 36. Hub Downstream Ports Control Register

Hub Downstream Ports Control Register **ADDRESS 0x4B**

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

Table 12. Control Bit Definition for Downstream Ports

Control Bits		Control Action
Bit1	Bit 0	
0	0	Not Forcing (Normal USB Function)
0	1	Force Differential ‘1’ (D+ HIGH, D– LOW)
1	0	Force Differential ‘0’ (D+ LOW, D– HIGH)
1	1	Force SE0 state

An alternate means of forcing the downstream ports is through the Hub Ports Force Low Register (Figure 37). With these registers the pins of the downstream ports are individually forced LOW, or left unforced. Unlike the Hub Downstream Ports Control Register, above, the Force Low Register does not produce standard USB edge rate control on the forced pins. However, this register allows downstream port pins to be held LOW in suspend. This register is used to drive SE0 on all downstream ports when unconfigured, as required in the USB 1.1 specification.

Figure 37. Hub Ports Force Low Register

Hub Ports Force Low **ADDRESS 0x51**

Bit #	7	6	5	4	3	2	1	0
Bit Name	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]
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 data state of downstream ports are read through the HUB Ports SE0 Status Register (Figure 38) and the Hub Ports Data Register (Figure 39). The data read from the Hub Ports Data Register is the differential data only and is independent of the settings of the Hub Ports Speed Register (Figure 34). When the SE0 condition is sensed on a downstream port, the corresponding bits of the Hub Ports Data Register hold the last differential data state before the SE0. Hub Ports SE0 Status Register and Hub Ports Data Register are cleared upon reset or bus reset.

USB Non Control Endpoint Mode Registers

The format of the non control endpoint mode registers is shown in Figure 45.

Figure 45. USB Non Control Endpoint Mode Registers

USB Non Control Device Endpoint Mode **ADDRESSES 0x14, 0x16, 0x44**

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

Bits[3..0]: Mode

These sets the mode which control how the control endpoint responds to traffic. The mode bit encoding is shown in Table 12.

Bit 4: ACK

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

Bits[6..5]: Reserved

Must be written zero during register writes.

Bit 7: STALL

If this STALL is set, the SIE stalls an OUT packet if the mode bits are set to ACK-IN, and the SIE stalls an IN packet if the mode bits are set to ACK-OUT. For all other modes, the STALL bit must be a LOW.

USB Endpoint Counter Registers

There are five Endpoint Counter registers, with identical formats for both control and non control endpoints. These registers contain byte count information for USB transactions, and bits for data packet status. The format of these registers is shown in Figure 46.

Figure 46. USB Endpoint Counter Registers

USB Endpoint Counter **ADDRESSES 0x11, 0x13, 0x15, 0x41, 0x43**

Bit #	7	6	5	4	3	2	1	0
Bit Name	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
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[5..0]: Byte Count

These counter bits indicate the number of data bytes in a transaction. For IN transactions, firmware loads the count with the number of bytes to be transmitted to the host from the endpoint FIFO. Valid values are 0 to 32, inclusive. For OUT or SETUP transactions, the count is updated by hardware to the number of data bytes received, plus two for the CRC bytes. Valid values are 2 to 34, inclusive.

Bit 6: Data Valid

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

Bit 7: Data 0/1 Toggle

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

Whenever the count updates from a SETUP or OUT transaction on endpoint 0, the counter register locks and cannot be written by the CPU. Reading the register unlocks it. This prevents firmware from overwriting a status update on incoming SETUP or OUT transactions before firmware has a chance to read the data. Only endpoint 0 counter register is locked when updated. The locking mechanism does not apply to the count registers of other endpoints.

Endpoint Mode and Count Registers Update and Locking Mechanism

The contents of the endpoint mode and counter registers are updated, based on the packet flow diagram in Figure 47. Two time points, UPDATE and SETUP, are shown in the same figure. The following activities occur at each time point:

SETUP:

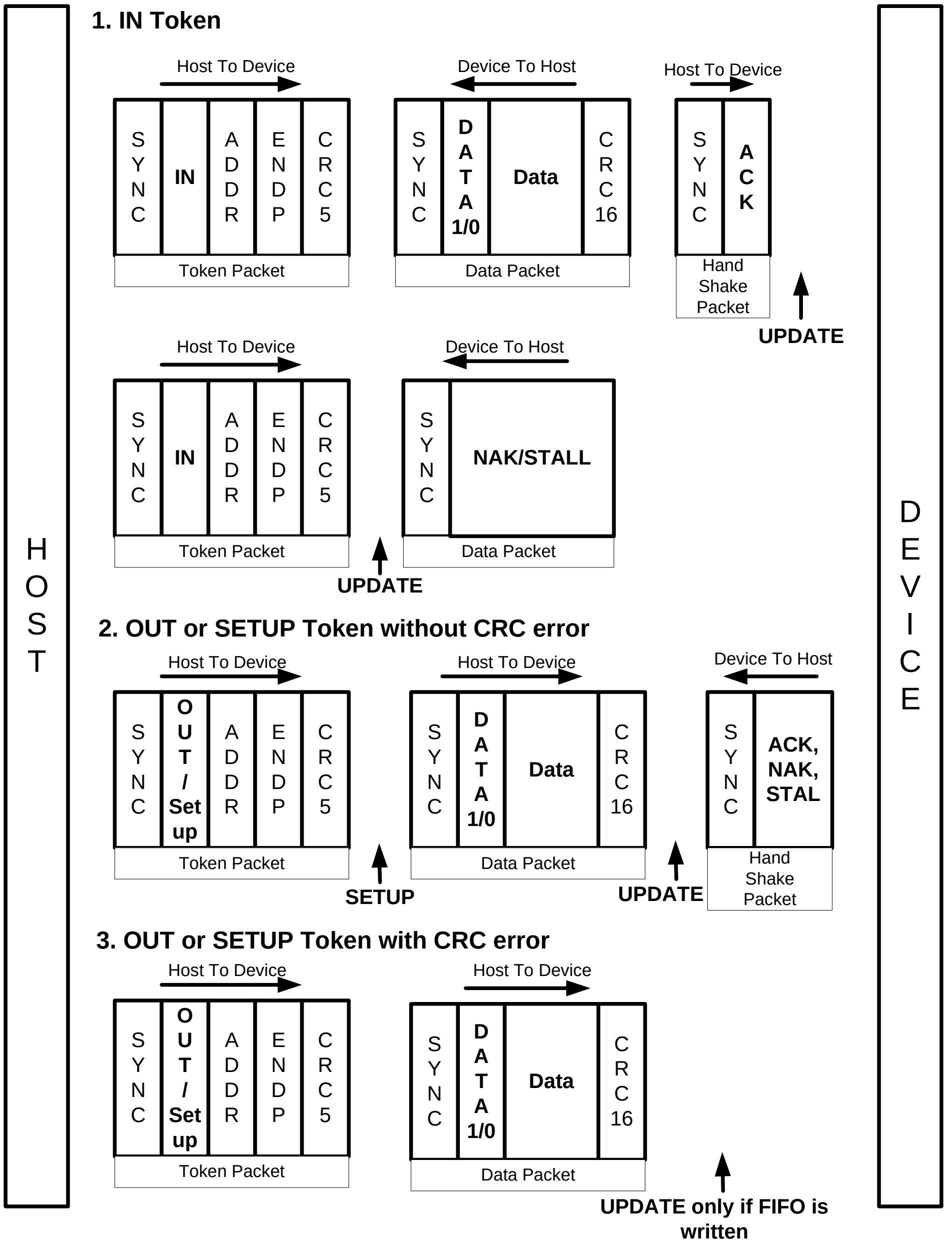
The SETUP bit of the endpoint 0 mode register is forced HIGH at this time. This bit is forced HIGH by the SIE until the end of the data phase of a control write transfer. The SETUP bit can not be cleared by firmware during this time.

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

UPDATE:

1. Endpoint Mode Register – All the bits are updated (except the SETUP bit of the endpoint 0 mode register).
2. Counter Registers – All bits are updated.
3. Interrupt – If an interrupt is to be generated as a result of the transaction, the interrupt flag for the corresponding endpoint is set at this time. For details on what conditions are required to generate an endpoint interrupt, refer to Table 16.
4. The contents of the updated endpoint 0 mode and counter registers are locked, except the SETUP bit of the endpoint 0 mode register which was locked earlier.

Figure 47. Token and Data Packet Flow Diagram

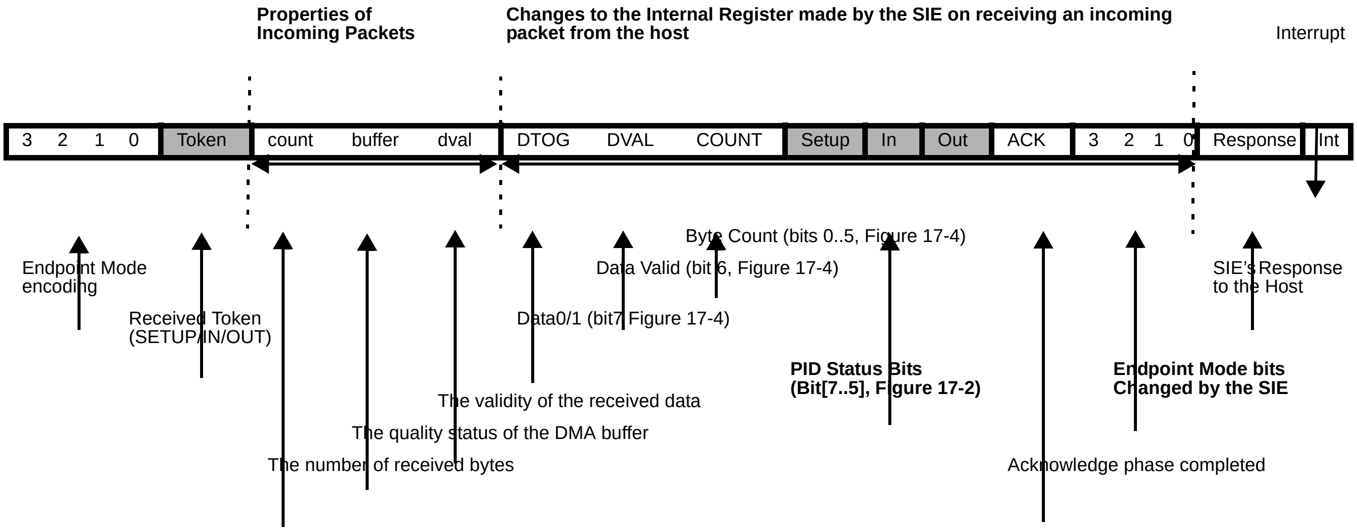


stage OUT token. The firmware needs to update the mode for the SIE to respond appropriately. See Table 12 for more details on what modes are changed by the SIE. A disabled endpoint remains disabled until changed by firmware, and all endpoints reset to the disabled mode (0000). Firmware normally enables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

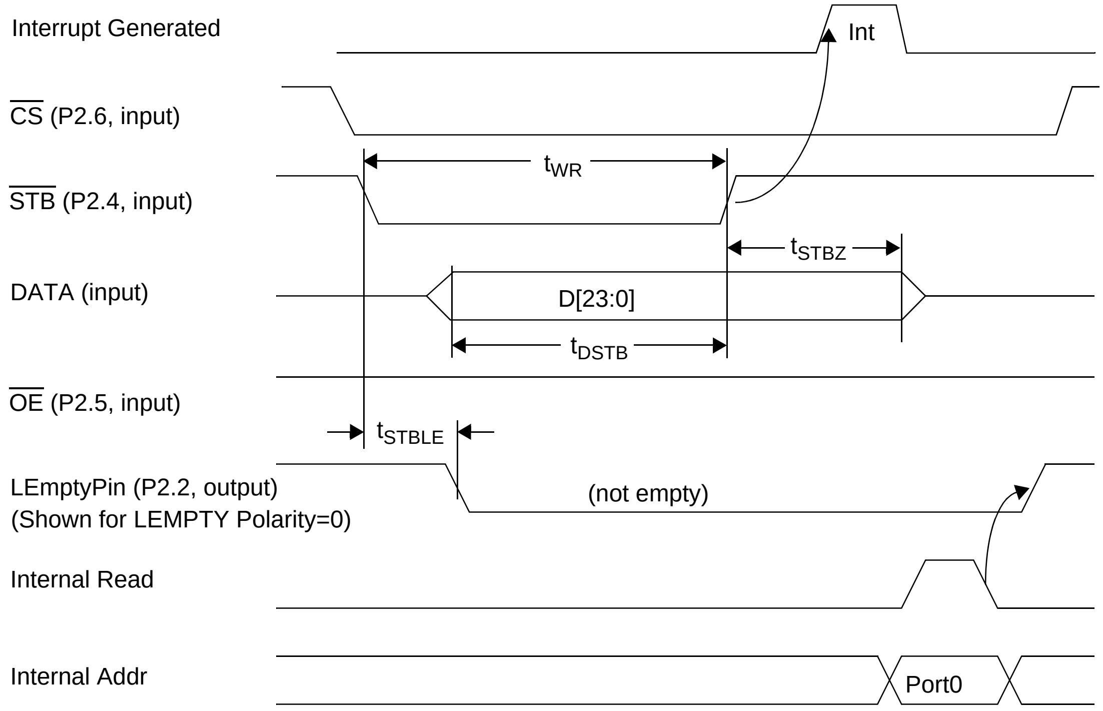
The response of the SIE are summarized as follows:

- 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)

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	

Figure 52. HAPI Write by External Device to USB Microcontroller

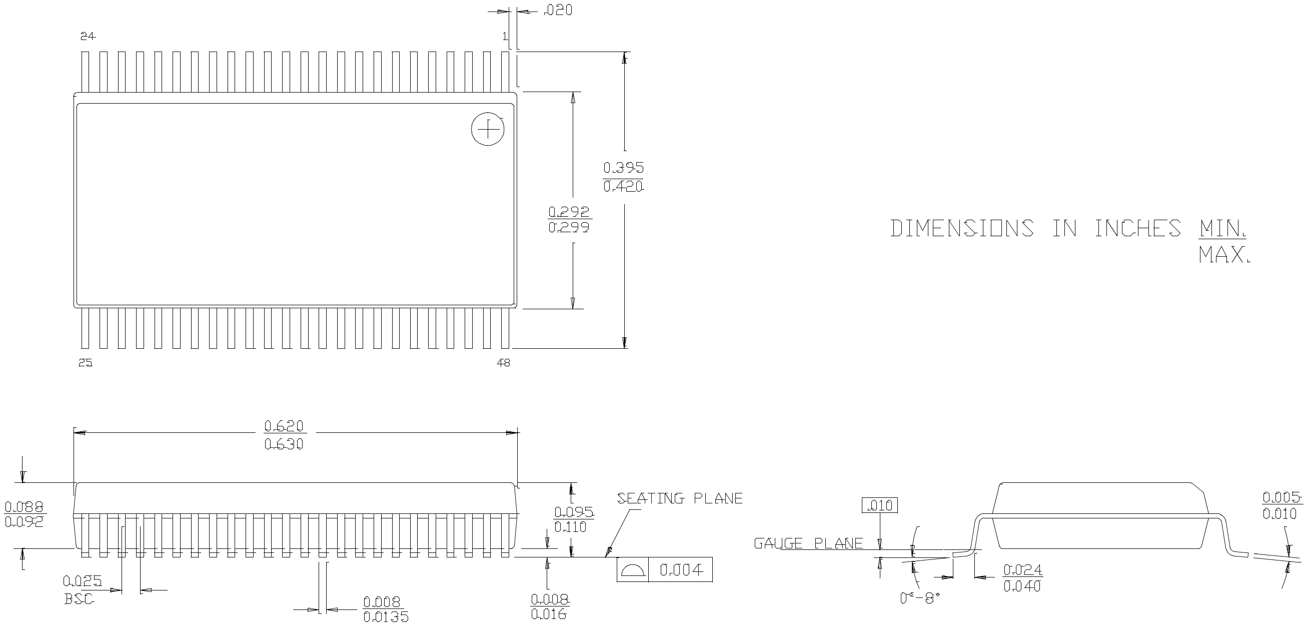


Ordering Information

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

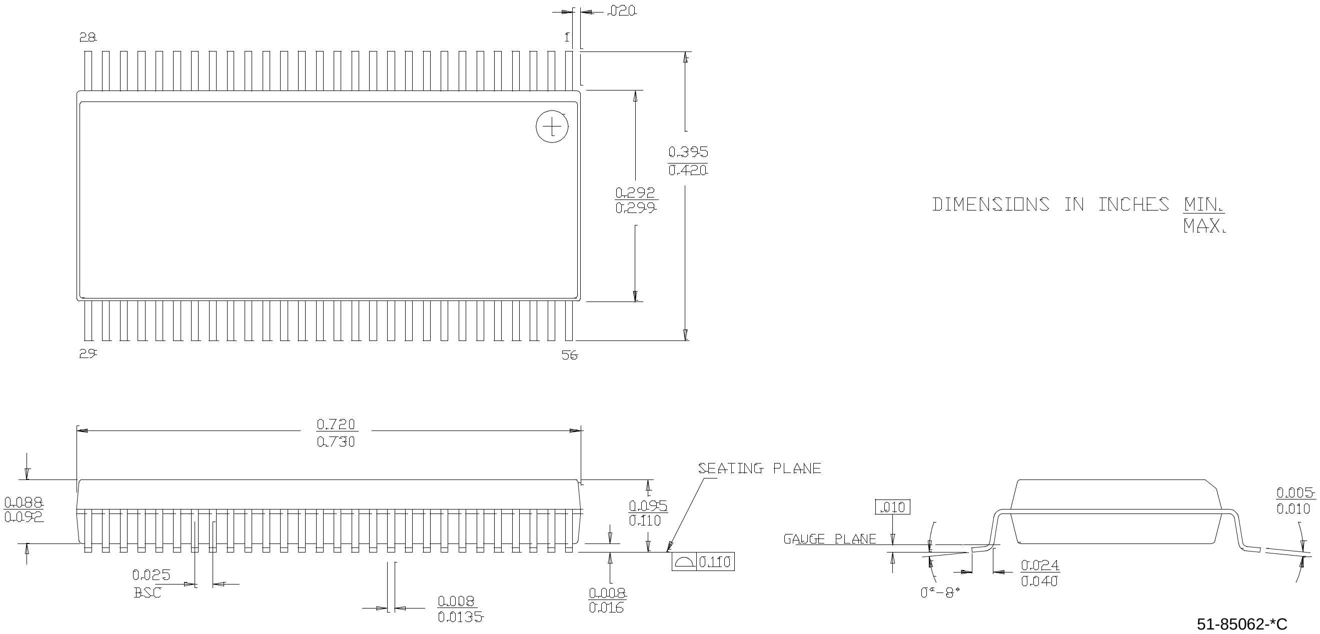
Package Diagrams

Figure 53. 48-Pin Shrunk Small Outline Package O48



51-85061-*C

Figure 54. 56-Pin Shrunk Small Outline Package O56



51-85062-*C