CY7C68001
EZ-USB *SX2*
High-Speed USB Interface Device

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF TABLES

1.0 EZ-USB SX2 Features

1.1 Introduction

The SX2 device controller is designed to work with any standard microcontroller or digital signal processor. The controller has a built in USB transceiver and Serial Interface Engine interface, along with a command decoder for sending and receiving USB data. The controller has 4 endpoints and 4 FIFO buffers for maximum flexibility and throughput, as well as control endpoint 0. The controller has three address pins and either an 8- or 16- bit data bus for command and data input or output.

1.2 Features

- **USB 2.0 compliant**
- **Operates at High (480 Mbps) or Full (12 Mbps) speed**
- **Supports 5 endpoints**
  - Endpoint 0 for control transfers
  - Endpoints 2, 4, 6, 8 for application specific control and data
- **Standard microcontroller interface**

1.3 Block Diagram

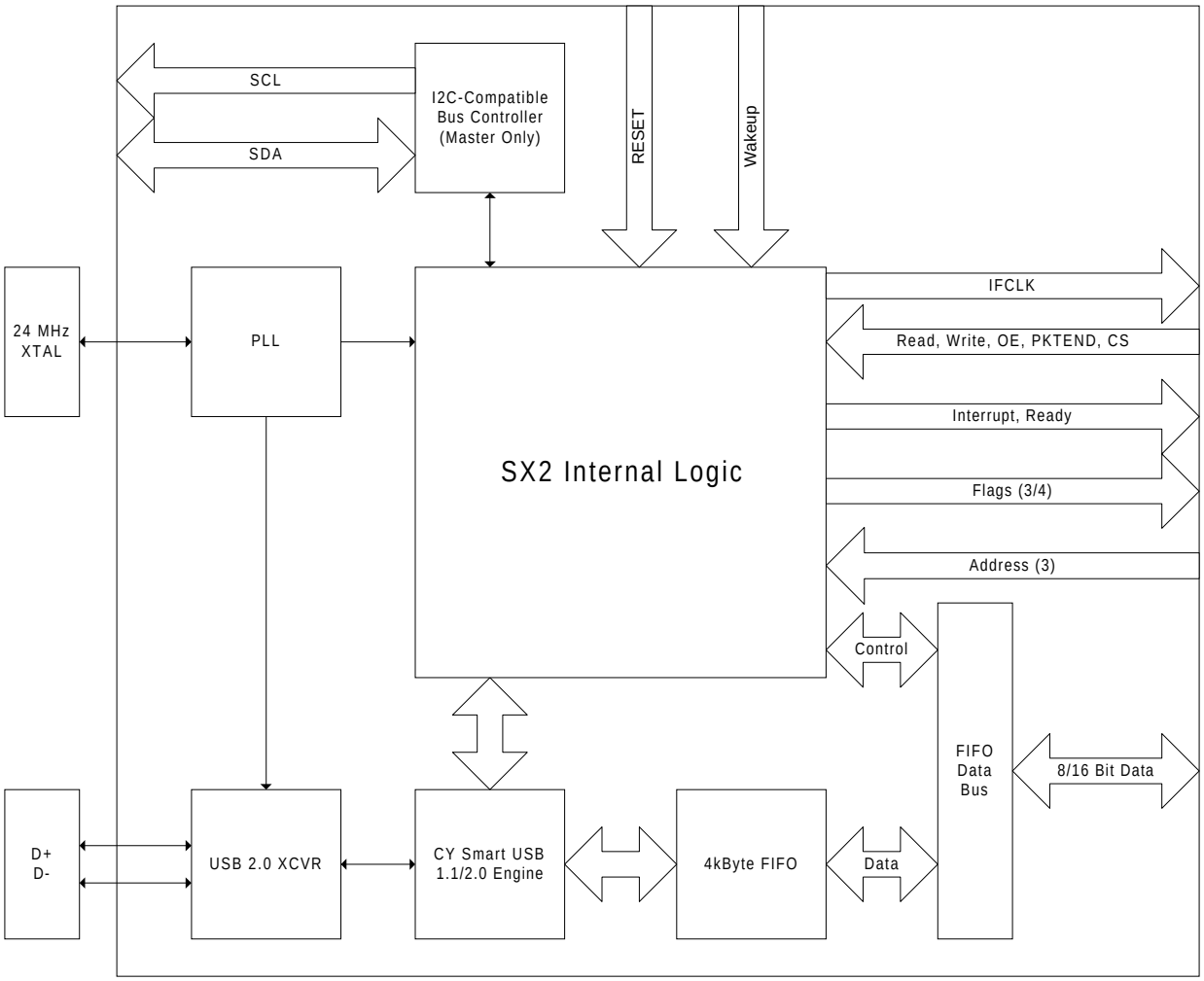

Figure 1-1. Block Diagram

2.0 Applications

- **DSL Modems**
- **ATA Interface**
- **Memory Card Readers**
- **Legacy Conversion Devices**
- **Cameras**
- **Scanners**
- **Home PNA**
- **Wireless LAN**
- **MP3 Players**
- **Networking**

3.0 Functional Overview

3.1 USB Signaling Speed

SX2 operates at two of the three rates defined in the Universal Serial Bus Specification Revision 2.0, dated April 27, 2000:

- **Full speed, with a signaling bit rate of 12 Mbits**
- **High speed, with a signaling bit rate of 480 Mbits**

SX2 does not support the low-speed signaling mode of 1.5 Mbits

3.2 I^2C-Compatible Bus

Master only, 100/400 kilobits. SCL and SDA pins have open-drain outputs and hysteresis inputs. These signals must be pulled up to 3.3V using 2.2-KΩ resistors, even if no I^2C-compatible device is connected.

3.3 Buses

8- or 16- bit bidirectional data bus.

3.4 USB Boot Methods

During the power-up sequence, internal logic checks for the presence of an I^2C-compatible EEPROM whose first byte is 0xC4. If found, it enumerates using the descriptor in the EEPROM and signals to the external processor when enumeration is complete. If not found, it waits for the external processor to write the descriptor to the SX2 before it enumerates.

NOTE: The SCL and SDA pins must be pulled up, even if an EEPROM is not connected, for this detection method to work properly.

3.5 Interrupt System

3.5.1 Architecture

The SX2 has an interrupt signal that indicates to the external processor that it has an interrupt condition or data that needs to be read. The SX2 has seven interrupt sources. Each interrupt can be enabled or disabled by setting the corresponding bit in the INTENABLE register. When an interrupt occurs, the interrupt pin will be asserted and the corresponding bit will be set in the IRQ register. Reading the IRQ register automatically clears the interrupt. Only one interrupt request will occur at any time; the SX2 buffers multiple pending interrupts. The interrupt priority is (highest) X,X,X,X,X,X,X (lowest).

If the external processor has initiated a read register sequence, then the SX2 will buffer interrupts until the external processor has read the data. This insures that after a read sequence has begun, the next interrupt that is received from the SX2 will indicate that the corresponding data is available.

3.5.2 EP0TXD

If enabled, the interrupt pin will be asserted and this bit will be set when a packet has been transferred from the SX2 to the host. This will only occur if the packet has originated from the external processor and not from the SX2 automatically handling endpoint zero traffic.

3.5.3 EP0RXD

If enabled, the interrupt pin will be asserted and this bit will be set when a packet has been received from the host. This will only occur if the packet can not be automatically handled by the SX2.

3.5.4 I2CDONE

If enabled, the interrupt pin will be asserted and this bit will be set when the I^2C-compatible bus engine has completed a transmission. This interrupt is used to gate successive I^2C-compatible bus reads or writes.

3.5.5 ENUMOK

If enabled, the interrupt pin will be asserted and this bit will be set when the SX2 has finished the enumeration process. This is determined by the host issuing a SET_CONFIGURATION request.

3.5.6 SUSP/RES

If enabled, the interrupt pin will be asserted and this bit will be set when the SX2 detects a suspend or resume condition on the USB bus.

3.5.7 DMAEOT

If enabled, the interrupt pin will be asserted and this bit will be set when the SX2 has finished transferring the number of packets specified in the DMACOUNTER register. The contents of the DMACOUNTER register are unaffected.

3.6 Reset and Wakeup

3.6.1 Reset Pin

An input pin (RESET#) resets the chip. This pin has hysteresis and is active LOW. The internal PLL stabilizes approximately X µs after Vcc has reached 3.3 volts. Typically, an external RC network (R=100K, C = 0.1µF) is used to provide the RESET# signal.

3.6.2 Wake Up Pin

The SX2 exits the power down (USB suspend) state when one of the following occurs:

- **USB bus signals resume**
- **External logic asserts the WAKEUP pin**

3.7 Endpoint RAM

3.7.1 Size

- **8x512 bytes (Endpoints 2,4,6,8)**
- **1x64 bytes (Endpoint 0)**

3.7.2 Organization

- **EP0 Bidirectional endpoint zero, 64 byte buffer.**
- **EP2,4,6,8 Eight 512-byte buffers, bulk, interrupt, or isochronous. EP2 & 6 can be either double, triple, or quad buffered. For High Speed endpoint configuration options see *Figure 3-1*.**

3.7.3 Endpoint Configurations (High speed mode)

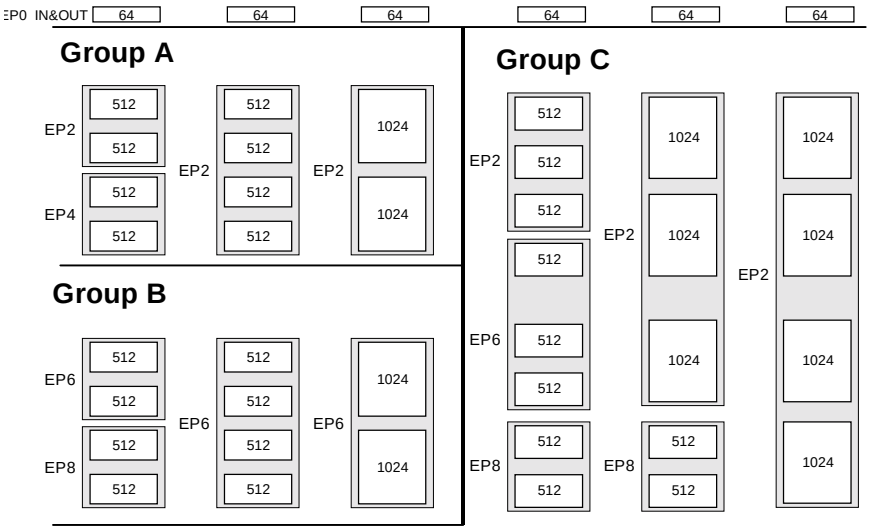

Figure 3-1. Endpoint Configuration

Endpoint 0 is the same for every configuration. Endpoint zero is only the CONTROL endpoint. For Endpoints 2,4,6,8, refer to Figure 3-1. Endpoints 2,4,6, and 8 may be configured by choosing either:

1. one configuration from Group A and one form Group B, or
2. one configuration from Group C

Some example endpoint configurations:

- **EP2: 1024 bytes double-buffered, EP6: 512 bytes quad-buffered**
- **EP2: 512 bytes double-buffered, EP4: 512 bytes double-buffered, EP6: 512 bytes double-buffered, EP8: 512 bytes double buffered**
- **EP2: 1024 bytes quad-buffered**

3.8 External Interface

3.8.1 Architecture

The SX2 slave FIFO architecture has eight 512-byte blocks in the endpoint RAM which directly serve as FIFO memories, and are controlled by FIFO control signals (such as IFCLK, CS, SLRD, SLWR, SLOE, PKTEND, and flags).

The SX2 command interface is used to setup the SX2 endpoints, control endpoint 0, interrupts, DMA, and descriptor data. The command interface has its own READY signal for gating writes, and an INTERRUPT signal to indicate that it has data to be read. It uses the same control signals (IFCLK, SLRD, SLWR, SLOE, CS).

In operation, some of the eight RAM blocks fill or empty from the SIE, while the others are connected to the I/O transfer logic. The transfer logic takes the form of a slave FIFO interface for externally controlled transfers. The SX2 endpoint FIFOs are implemented as eight physically distinct 256x16 RAM blocks. At any given time, some RAM blocks are filling/emptying with USB data under SIE control, while other RAM blocks are available to the external processor. The blocks can be configured as double, triple, or quad buffered as previously shown.

3.8.2 Control Signals

The SX2 has three address pins which are used to select either the FIFOs or the command interface. The addresses correspond to the following table. CS#, if enabled, overides any address selection.

Table 3-1. Address Truth-table

Address/ Selection	CS#	FIFOADR2	FIFOADR1	FIFOADR0
FIFO2	0	0	0	0
FIFO4	0	0	0	1
FIFO6	0	0	1	0
FIFO8	0	0	1	1
COMMAND	0	1	X	X
DISABLED	1	X	X	X

The SX2 accepts either an internally derived clock or externally supplied clock (IFCLK, max. frequency 48 MHz), and SLRD, SLWR, SLOE, PKTEND, CS signals from external logic. The interface can be selected for 8 or 16 bit operation by an internal configuration bit, and an Output Enable signal SLOE enables data of the selected width. External logic must insure that the output enable signal is inactive when writing data to the SX2. The interface can also operate asynchronously, where the SLRD and SLWR signals act directly as strobes, rather than as clock qualifiers in synchronous mode. An optional CS signal will three-state the data bus and ignore SLRD, SLWR, and address pin strobes.

3.8.3 FIFO Clock Rates

An SX2 register bit selects one of two frequencies for the internally supplied interface clock: 30 MHz or 48 MHz. Alternatively, an externally supplied clock, of up to 48 MHz, feeding the IFCLK pin can be used as the interface clock. IFCLK can be configured to function as an output clock when the FIFOs are internally clocked. An output enable bit in the IFCONFIG register turns this clock output off, if desired.

3.9 I^2C-Compatible Bus Controller

SX2 has one I^2C-compatible port that is driven by two internal controllers, one that automatically operates at boot time to load VID/PID/DID and configuration information, and another that the external processor can use to control external I^2C-compatible devices. The I^2C-compatible port operates in master mode only.

3.9.1 I^2C-Compatible Bus Pins

The I^2C-compatible Bus pins SCL and SDA must have external X.X-KΩ pull-up resistors.

3.9.2 EEPROM Boot Load Access

At power-on reset the I^2C-compatible Bus loader will load up to 1 KB of descriptor data. Boot loads only occur after power-on reset.

3.9.3 I^2C-Compatible Bus General Purpose Access

The external processor can control peripherals connected to the I^2C-compatible bus using the I2CTL and I2DAT registers. SX2 provides master control only; it is never an I^2C-compatible slave.

4.0 DMA

The SX2 has a DMACOUNTER register used for automatic multiple packet transfers. The external processor can set-up its DMA engine and write the number of packets to the DMACOUNTER register. When the SX2 has counted the appropriate number of packets transferred, it will notify the external processor with an interrupt. This allows automatic unattended data transfers of up to 256 kbytes.

5.0 Enumeration

The SX2 has 1 KB of descriptor RAM into which the external processor writes its descriptor. The SX2 will automatically respond to chapter 9 USB requests without any external processor intervention. Once enumeration is complete, the SX2 will notify the

external processor with an interrupt. If the SX2 receives a request from the host that it does not know how to respond to, it will interrupt the external processor. The external processor then has the choice of responding to the request or stalling.

6.0 Pin Assignments

Figure 6-1 identifies all signals for the 56-pin package. The following page illustrates the pin diagram.

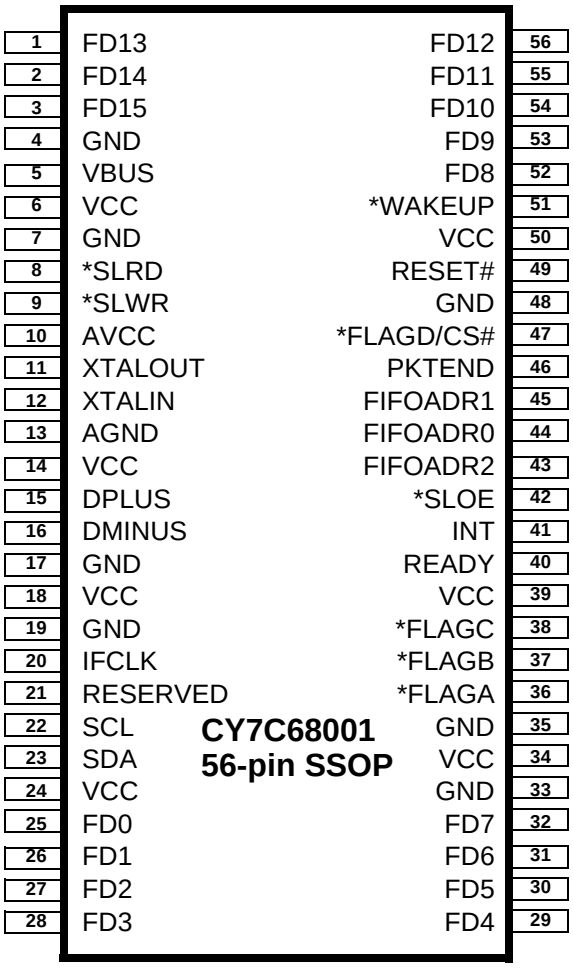

Figure 6-1. CY7C68001 56-pin SSOP Pin Assignment

* denotes programmable polarity.

6.1 CY7C68001 Pin Descriptions

Table 6-1. SX2 Pin Descriptions

Pin	Name	Type	Default	Description
10	AVCC	Power	N/A	**Analog V_{CC}**. This signal provides power to the analog section of the chip.
13	AGND	Power	N/A	**Analog Ground**. Connect to ground with as short a path as possible.
16	USBD–	I/O/Z	Z	**USB D– Signal**. Connect to the USB D– signal.
15	USBD+	I/O/Z	Z	**USB D+ Signal**. Connect to the USB D+ signal.
49	RESET#	Input	N/A	**Active LOW Reset**. Resets the entire chip. This pin is normally tied to V_{CC} through a 100K resistor, and to GND through a 0.1-µF capacitor.
12	XIN	Input	N/A	**Crystal Input**. Connect this signal to a 24-MHz parallel-resonant, fundamental mode crystal and 20-pF capacitor to GND.
11	XOUT	Output	N/A	**Crystal Output**. Connect this signal to a 24-MHz parallel-resonant, fundamental mode crystal and 20-pF capacitor to GND.
5	VBUS	Power	N/A	**Vbus**. Connect to Vbus or +5V if self powered.
40	READY	Output	H	**READY** is an output-only ready that gates external command reads and writes.
41	INTERRUPT	Output	L	**INTERRUPT** is an output-only external interrupt signal.
42	SLOE	Input	I	**SLOE** is an input-only output enable with programmable polarity (FIFOPOLAR.4) for the slave FIFOs connected to FD[0..7] or FD[0..15].
43	FIFOADR2	Input	I	**FIFOADR2** is an input-only address select for the slave FIFOs connected to FD[0..7] or FD[0..15].
44	FIFOADR0	Input	I	**FIFOADR0** is an input-only address select for the slave FIFOs connected to FD[0..7] or FD[0..15].
45	FIFOADR1	Input	I	**FIFOADR1** is an input-only address select for the slave FIFOs connected to FD[0..7] or FD[0..15].
46	PKTEND	Input	I	**PKTEND** is an input-only packet end with programmable polarity (FIFOPOLAR.5) for the slave FIFOs connected to FD[0..7] or FD[0..15].
47	FLAGD/CS#	I/O	L	**FLAGD** is a programmable slave-FIFO output status flag signal. **CS#** is a master chip select.
25	FD[0]	I/O/Z	I	**FD[0]** is the bidirectional FIFO data bus.
26	FD[1]	I/O/Z	I	**FD[1]** is the bidirectional FIFO data bus.
27	FD[2]	I/O/Z	I	**FD[2]** is the bidirectional FIFO data bus.
28	FD[3]	I/O/Z	I	**FD[3]** is the bidirectional FIFO data bus.
29	FD[4]	I/O/Z	I	**FD[4]** is the bidirectional FIFO data bus.
30	FD[5]	I/O/Z	I	**FD[5]** is the bidirectional FIFO data bus.
31	FD[6]	I/O/Z	I	**FD[6]** is the bidirectional FIFO data bus.
32	FD[7]	I/O/Z	I	**FD[7]** is the bidirectional FIFO data bus.
52	FD[8]	I/O/Z	I	**FD[8]** is the bidirectional FIFO data bus.
53	FD[9]	I/O/Z	I	**FD[9]** is the bidirectional FIFO data bus.
54	FD[10]	I/O/Z	I	**FD[10]** is the bidirectional FIFO data bus.
55	FD[11]	I/O/Z	I	**FD[11]** is the bidirectional FIFO data bus.
56	FD[12]	I/O/Z	I	**FD[12]** is the bidirectional FIFO data bus.
1	FD[13]	I/O/Z	I	**FD[13]** is the bidirectional FIFO data bus.
2	FD[14]	I/O/Z	I	**FD[14]** is the bidirectional FIFO data bus.
3	FD[15]	I/O/Z	I	**FD[15]** is the bidirectional FIFO data bus.

Table 6-1. SX2 Pin Descriptions (continued)

Pin	Name	Type	Default	Description
8	SLRD	Input	N/A	**SLRD** is the input-only read strobe with programmable polarity (FIFOPOLAR.3) for the slave FIFOs connected to FDI[0..7] or FDI[0..15].
9	SLWR	Input	N/A	**SLWR** is the input-only write strobe with programmable polarity (FIFOPOLAR.2) for the slave FIFOs connected to FDI[0..7] or FDI[0..15].
36	FLAGA	Output	H	**FLAGA** is a programmable slave-FIFO output status flag signal. Defaults to PRGFLAG for the FIFO selected by the FIFOADR[1:0] pins.
37	FLAGB	Output	H	**FLAGB** is a programmable slave-FIFO output status flag signal. Defaults to FULL for the FIFO selected by the FIFOADR[1:0] pins.
38	FLAGC	Output	H	**FLAGC** is a programmable slave-FIFO output status flag signal. Defaults to EMPTY for the FIFO selected by the FIFOADR[1:0] pins.
20	IFCLK	I/O/Z	Z	Interface Clock, used for synchronously clocking data into or out of the slave FIFOs. IFCLK also serves as a timing reference for all slave FIFO control signals and GPIF. When internal clocking, IFCONFIG.7 = 1, is used the IFCLK pin can be configured to output 30/48 MHz by bits IFCONFIG.5 and IFCONFIG.6 and may be inverted by setting the bit IFCONFIG.4 = 1.
21	Reserved	Input	N/A	**Reserved**. Connect to ground.
51	WAKEUP	Input	N/A	**USB Wakeup**. If the SX2 is in suspend, asserting this pin starts up the oscillator and interrupts the SX2 to allow it to exit the suspend mode. Holding WAKEUP asserted inhibits the SX2 chip from suspending. This pin has programmable polarity (WAKEUP.4).
22	SCL	OD	Z	**I^2C-compatible Clock**. Connect to V_{CC} with a X.X K resistor, **even if no I^2C-compatible peripheral is attached.**
23	SDA	OD	Z	**I^2C-Compatible Data**. Connect to V_{CC} with a X.X K resistor, **even if no I^2C-compatible peripheral is attached.**
6	V_{CC}	Power	N/A	**V_{CC}**. Connect to 3.3 V power source.
14	V_{CC}	Power	N/A	**V_{CC}**. Connect to 3.3 V power source.
18	V_{CC}	Power	N/A	**V_{CC}**. Connect to 3.3 V power source.
24	V_{CC}	Power	N/A	**V_{CC}**. Connect to 3.3 V power source.
34	V_{CC}	Power	N/A	**V_{CC}**. Connect to 3.3 V power source.
39	V_{CC}	Power	N/A	**V_{CC}**. Connect to 3.3 V power source.
50	V_{CC}	Power	N/A	**V_{CC}**. Connect to 3.3 V power source.
4	GND	Ground	N/A	**Ground**.
7	GND	Ground	N/A	**Ground**.
17	GND	Ground	N/A	**Ground**.
19	GND	Ground	N/A	**Ground**.
33	GND	Ground	N/A	**Ground**.
35	GND	Ground	N/A	**Ground**.
48	GND	Ground	N/A	**Ground**.

7.0 Register Summary

Table 7-1. SX2 Register Summary

Hex	Size	Name	Description	D7	D6	D5	D4	D3	D2	D1	D0	Default	Access
		General Configuration											
01	1	IFCONFIG	Interface Configuration	IFCLKSRC	3048MHZ	IFCLKOE	IFCLKPOL	ASYNC			WORDWIDE	11000000	RW
02	1	FLAGSAB	FIFO FLAGA and FLAGB Assignments	FLAGB3	FLAGB2	FLAGB1	FLAGB0	FLAGA3	FLAGA2	FLAGA1	FLAGA0		
03	1	FLAGSCD	FIFO FLAGC and FLAGD Assignments	FLAGD3	FLAGD2	FLAGD1	FLAGD0	FLAGC3	FLAGC2	FLAGC1	FLAGC0		
09	1	FIFOPOLAR	FIFO polarities	WUPOL	0	PKTEND	OE	RD	WR	EF	FF	00000000	RW
0A	1	REVID	Chip Revision	rv7	rv6	rv5	rv4	rv3	rv2	rv1	rv0	00000000	R
		Endpoint Configuration											
12	1	EP2CFG	Endpoint 2 Configuration	VALID	dir	TYPE1	TYPE0	SIZE	STALL	BUF1	BUF0	10100010	RW
13	1	EP4CFG	Endpoint 4 Configuration	VALID	dir	TYPE1	TYPE0	0	STALL	0	0	10100000	RW
14	1	EP6CFG	Endpoint 6 Configuration	VALID	dir	TYPE1	TYPE0	SIZE	STALL	BUF1	BUF0	11100010	RW
15	1	EP8CFG	Endpoint 8 Configuration	VALID	dir	TYPE1	TYPE0	0	STALL	0	0	11100000	RW
20	1	EP2PKTLENH	Endpoint 2 Packet Length H (IN only)	0	0	0	0	0	PL10	PL9	PL8	00000010	RW
21	1	EP2PKTLENL	Endpoint 2 Packet Length L (IN only)	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
22	1	EP4PKTLENH	Endpoint 4 Packet Length H (IN only)	0	0	0	0	0	0	PL9	PL8	00000010	RW
23	1	EP4PKTLENL	Endpoint 4 Packet Length L (IN only)	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
24	1	EP6PKTLENH	Endpoint 6 Packet Length H (IN only)	0	0	0	0	0	PL10	PL9	PL8	00000010	RW
25	1	EP6PKTLENL	Endpoint 6 Packet Length L (IN only)	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
26	1	EP8PKTLENH	Endpoint 8 Packet Length H (IN only)	0	0	0	0	0	0	PL9	PL8	00000010	RW
27	1	EP8PKTLENL	Endpoint 8 Packet Length L (IN only)	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
30	1	EP2PFH	EP2 Programmable Flag H	DECIS	PKTSTAT	IN:PKTS[2] OUT:PFC12	IN:PKTS[1] OUT:PFC11	IN:PKTS[0] OUT:PFC10	0	PFC9	PFC8	10000010	RW
31	1	EP2PFL	EP2 Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
32	1	EP4PFH	EP4 Programmable Flag H	DECIS	PKTSTAT	0	IN: PKTS[1] OUT:PFC10	IN: PKTS[0] OUT:PFC9	0	0	PFC8	10000010	RW
33	1	EP4PFL	EP4 Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
34	1	EP6PFH	EP6 Programmable Flag H	DECIS	PKTSTAT	IN:PKTS[2] OUT:PFC12	IN:PKTS[1] OUT:PFC11	IN:PKTS[0] OUT:PFC10	0	PFC9	PFC8	10000010	RW
35	1	EP6PFL	EP6 Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
36	1	EP8PFH	EP8 Programmable Flag H	DECIS	PKTSTAT	0	IN: PKTS[1] OUT:PFC10	IN: PKTS[0] OUT:PFC9	0	0	PFC8	10000010	RW
37	1	EP8PFL	EP8 Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
40	1	EP2ISOINPKTS	EP2 (if ISO) IN Packets per frame (1-3)	0	0	0	0	0	0	INPPF1	INPPF0	00000001	RW
41	1	EP4ISOINPKTS	EP4 (if ISO) IN Packets per frame (1-3)	0	0	0	0	0	0	INPPF1	INPPF0	00000001	RW
42	1	EP6ISOINPKTS	EP6 (if ISO) IN Packets per frame (1-3)	0	0	0	0	0	0	INPPF1	INPPF0	00000001	RW
43	1	EP8ISOINPKTS	EP8 (if ISO) IN Packets per frame (1-3)	0	0	0	0	0	0	INPPF1	INPPF0	00000001	RW
		I2C											
78	1	I2CS	Control & Status	START	STOP	LASTRD			BERR	ACK	DONE	000xx000	bbbrrrrr
79	1	I2DAT	Data	d7	d6	d5	d4	d3	d2	d1	d0	xxxxxxxx	RW
7A	1	I2CTL	I2C Control	0	0	0	0	0	0	STOPIE	400kHz	00000000	RW
		USB Configuration											
84	1	USBFRAMEH	USB Frame count H	0	0	0	0	0	FC10	FC9	FC8	xxxxxxxx	R
85	1	USBFRAMEL	USB Frame count L	FC7	FC6	FC5	FC4	FC3	FC2	FC1	FC0	xxxxxxxx	R
86	1	MICROFRAME	Microframe count, 0-7	0	0	0	0	0	MF2	MF1	MF0	xxxxxxxx	R
87	1	FNADDR	USB Function address	HSGRANT	FA6	FA5	FA4	FA3	FA2	FA1	FA0	xxxxxxxx	R
		Interrupts											
	1	INTENABLE	Interrupt Enable	EP0TXD	EP0RXD			I2CDONE	ENUMOK	SUSP/RES	DMAEOT	11111111	RW
	1	IRQ	Interrupt Request Flags	EP0TXD	EP0RXD			I2CDONE	ENUMOK	SUSP/RES	DMAEOT	00000000	R
		DMA											
	1	DMACOUNTER	DMA Operation Counter	c7	c6	c5	c4	c3	c2	c1	c0	00000000	RW
		Descriptor											
	1024	DESC	Descriptor RAM	d7	d6	d5	d4	d3	d2	d1	d0	0000000	RW
		Endpoint Buffers											
	64	EP0BUF	Endpoint 0 Buffer	d7	d6	d5	d4	d3	d2	d1	d0	xxxxxxxx	RW

7.1 IFCONFIG.

7.1.1 IFCLKSRC

This bit selects the clock source for the FIFOs. If IFCLKSRC = 0, the external clock on the IFCLK pin is selected. If IFCLKSRC = 1, an internal 30/48 MHz clock is used. Internal logic sets this bit to 1 at power-on.

7.1.2 3048MHZ

This bit selects the internal FIFO clock frequency. If 3048MHZ = 0, the internal clock frequency is 30 MHz. If 3048MHZ = 1, the internal clock frequency is 48 MHz. Internal logic sets this bit to 1 at power-on.

7.1.3 IFCLKOE

This bit selects if the IFCLK pin is driven. If IFCLKOE = 0, the IFCLK pin is floated. If IFCLKOE = 1, the IFCLK pin is driven. Internal logic sets this bit to 0 at power up.

7.1.4 IFCLKPOL

This bit controls the polarity of the IFCLK signal.

When IFCLKPOL = 0, the clock has normal polarity.

When IFCLKPOL = 1, the clock is inverted.

7.1.5 ASYNC

This bit controls whether the FIFO interface is synchronous or asynchronous. When ASYNC = 0, the FIFOs operate synchronously. In synchronous mode, a clock is supplied either internally or externally on the IFCLK pin. The FIFO control signals function as read and write enable signals for the clock signal.

When ASYNC = 1, the FIFOs operate asynchronously. No clock signal input to IFCLK is required, and the FIFO control signals function directly as read and write strobes.

7.1.6 WORDWIDE

This bit controls whether the data interface is 8 or 16 bits wide. If WORDWIDE = 0, then the data interface is 8 bits wide, and FD[15:8] have no function. If WORDWIDE = 1, then the data interface is 16 bits wide.

7.2 FLAGSAB/CD.

The SX2 has four FIFO flags output pins, FLAGA, FLAGB, FLAGC, FLAGD. These flags can be programmed to represent various FIFO flags using four select bits for each FIFO. The 4-bit coding for all four flags is the same, as shown in the following table.

FLAGx3	FLAGx2	FLAGx1	FLAGx0	Pin Function
0	0	0	0	FLAGA=PF, FLAGB=FF, FLAGC=EF, FLAGD=?? (Actual FIFO is selected by FIFOADR[0:1] pins)
0	0	0	1	reserved
0	0	1	0	reserved
0	0	1	1	reserved
0	1	0	0	EP2 PF
0	1	0	1	EP4 PF
0	1	1	0	EP6 PF
0	1	1	1	EP8 PF
1	0	0	0	EP2 EF
1	0	0	1	EP4 EF
1	0	1	0	EP6 EF
1	0	1	1	EP8 EF
1	1	0	0	EP2 FF
1	1	0	1	EP4 FF
1	1	1	0	EP6 FF

FLAGx3	FLAGx2	FLAGx1	FLAGx0	Pin Function
1	1	1	1	EP8 FF

For the default (0000) selection, the four FIFO flags are fixed-function as shown in the first table entry. The input pins FIFOADR1 and FIFOADR0 select to which of the four FIFOs the flags correspond. These pins are decoded as shown in *Table 3-1*

The other (non-zero) values of FLAGx[3:0] allow the designer to independently configure the four flag outputs FLAGA-FLAGD to correspond to any flag-Programmable, Full, or Empty-from any of the four endpoint FIFOs. This allows each flag to be assigned to any of the four FIFOs, including those not currently selected by the FIFOADDR pins. For example, the external logic could be filling the EP2IN FIFO with data while also checking the full flag for the EP4OUT FIFO.

7.3 FIFOPOLAR.

This register controls the polarities of some FIFO signals and also the wakeup pin.

7.3.1 PKTEND

This flag selects the polarity of the PKTEND FIFO input pin. 0 selects the polarity shown in the data sheet (active LOW). 1 selects active HIGH.

7.3.2 SLOE

This flag selects the polarity of the SLOE FIFO input pin. 0 selects the polarity shown in the data sheet (active LOW). 1 selects active HIGH.

7.3.3 SLRD

This flag selects the polarity of the SLRD FIFO input pin. 0 selects the polarity shown in the data sheet (active LOW). 1 selects active HIGH.

7.3.4 SLWR

This flag selects the polarity of the SLWR FIFO input pin. 0 selects the polarity shown in the data sheet (active LOW). 1 selects active HIGH.

7.3.5 EF

This flag selects the empty-flag polarity.

7.3.6 FF

This flag selects the full-flag polarity.

7.4 REVID.

These register bits define the silicon revision. Consult individual Cypress Semiconductor datasheets for values.

7.5 EPxCFG.

These registers configure the large, data-handling SX2 endpoints. *Figure 3-1* shows the configuration choices for these endpoints. Shaded blocks group endpoint buffers for double-, triple-, or quad-buffering. The endpoint direction is set independently — any shaded block can have any direction.

To the left of the vertical line, independent columns can be chosen above and below the line. For example, you can configure EP2 as a quad-buffered 512-byte IN endpoint (middle column), and EP6 as a double-buffered 512-byte OUT endpoint plus EP8 as a double-buffered 512-byte OUT endpoint (left column).

To the right of the vertical line, only one of the columns can be chosen. For example, if you want EP2 as triple-buffered IN, EP6 as triple-buffered OUT, and EP8 as double buffered 512 OUT, choose the middle column. Configuration choices can be made only on the left or right side of the diagram, so in this example none of the left side choices are available.

7.5.1 VALID

The external processor sets VALID = 1 to activate an endpoint, and VALID = 0 to de-activate it. All SX2 end-points default to valid. An endpoint whose VALID bit is 0 does not respond to any USB traffic.

7.5.2 DIR

0 = OUT, 1 = IN

7.5.3 TYPE

These bits define the endpoint type, as shown in the table below. The TYPE bits apply to all of the endpoint configuration registers. All SX2 endpoints except EP0 default to BULK.

TYPE1	TYPE0	Endpoint Type
0	0	Invalid
0	1	Isochronous
1	0	Bulk (Default)
1	1	Interrupt

7.5.4 SIZE

0 = 512 bytes, 1 = 1024 bytes

Endpoints 4 and 8 can only be 512 bytes. Endpoints 2 and 6 are selectable.

7.5.5 STALL

Each bulk endpoint (IN or OUT) has a STALL Bit in its Control and Status Register (bit 0). If the external processor sets this bit, any requests to the endpoint return a STALL handshake rather than ACK or NAK. The Get Status-Endpoint Request returns the STALL state for the endpoint indicated in byte 4 of the request. Note that bit 7 of the endpoint number EP (byte 4) specifies direction.

7.5.6 Bit 1–0 BUF

The amount of endpoint buffering is presented in table below.

BUF1	BUF0	Buffering
0	0	Quad
0	1	Invalid
1	0	Double
1	1	Triple

7.6 EPxPKTLENH/L

The external processors can use these registers to set smaller packet sizes than the physical buffer size (refer to the previously described EPxCFG registers). The default packet size is 512 bytes for all endpoints. Note that EP2 and EP6 can have maximum sizes of 1024 bytes, and EP4 and EP8 can have maximum sizes of 512 bytes, to be consistent with the endpoint structure.

7.7 EPxPFH/L

The Programmable Flag registers control when the programmable flag (PF) goes active for each of the four endpoint FIFOS: EP2, EP4, EP6 and EP8. The E*P*nFIFOH/L fields are interpreted differently for OUT and IN endpoints.

7.7.1 7.7.1 DECIS

If DECIS is set to 0, then PF = 1 when BC (Byte Count) <= PF. If DECIS is set to 1, then PF = 1 when BC >= PF.

7.7.2 7.7.2PKSTAT

If PKTSTAT=0: The PF (Programmable Flag) and BC (Byte Counts) refer to the entire full end-point FIFO. PKTSTAT = 1: The PF and BCs refer to the current packet.

For IN endpoints, the trigger registers can apply to either the full FIFO, comprising multiple packets, or only to the current packet being filled. The PKTSTAT bit controls this choice:

PKTSTAT	PF applies to	EPnPFH:L format
0	PKTS + Current packet bytes	PKTS[ ] PFC[ ]
1	Current Packet bytes only	PFC[ ]

7.7.3 7.7.3IN:PKTS[2..0]/OUT:PFC[12..10]

Writing the byte count with AUTOOUT or AUTOIN set is not recommended.

This three bits have a different meaning, depending on whether PKTSTAT equals “1” or “0” and whether the transfer is over an OUT or IN endpoint.

IN endpoints

When PKTSTAT = 0, the PF becomes active when there are PKTS packets plus PFC bytes in the FIFO.

When PKTSTAT = 1, the PF becomes active when there are PFC bytes in the FIFO, no matter how many packets are in the FIFO.

OUT endpoints

The PF becomes active when there are PFC bytes in the FIFO.

In the first example below, bits 5–3 have data that is required to complete the transfer. In the second example, bits 5–3 do not matter — those bits are Don’t Cares because PKTSTAT = 1.

Example 1:

Assume a Bulk IN transfer over Endpoint 2 and PKTSTAT = 0:

EP2FIFOPFH = 0001 0000

- b6 = 0 (or PKTSTAT = 0): this indicates that the transfer will include packets (as defined by bits 5, 4, and 3) plus bytes (the sum in the flag low register)
- b5b4b3 = 010 binary (or 2 decimal): this indicates the number of packets to expect during the transfer (in this case, two packets)

EP2FIFOPFL = 0011 0010

- plus 50 bytes in the currently filling packet
  (the sum of the binary bits in the EP2FIFOPFL register is 2 +16 + 32 = 50 decimal)

Example 2:

To perform an IN transfer of a number over the same endpoint, set PKTSTAT = 1 and write a value into the EP2FIFOPFL register:

EP2FIFOPFH = 01xxx000

EP2FIFOPFHL = 75

Setting PKTSTAT = 1 causes the PF decision to be based on the byte count alone, ignoring the packet count. This mode is valuable for double-buffered endpoints, where only the byte count of the currently-filling packet is important.

Example 3 of OUT Endpoint transfers: For OUT endpoints, the PF flag applies to the total number of bytes in the multi-packet FIFO, with no packet count field. Instead of representing byte counts in two segments, a packet count and a byte count for the currently emptying packet, the byte Trigger values indicate total bytes available in the FIFO.

7.7.4 7.7.4PFC[9..8]

Bits 1–0 of EP2FIFOPFH are bits 9–8 of the byte count register.

IN endpoints

When PKTSTAT = 0, the PF becomes active when there are PKTS packets plus PFC bytes in the FIFO.
When PKTSTAT = 1, the PF becomes active when there are PFC bytes in the FIFO, no matter how many packets are in the FIFO.

OUT endpoints

The PF becomes active when there are PFC bytes in the FIFO.

7.8 EPxISOINPKTS

For ISOCHRONOUS IN endpoints only, these registers determine the number of packets per frame (full-speed mode) or micro-frame (high speed mode) according to the following table.

INPPF1*	INPPF0	Packets
0	0	Invalid
0	1	1
1	0	2
1	1	3

***Must be 1 for full-speed mode**

7.9 I2CS

7.9.1 START

The external processor sets the START Bit to “1” to prepare a bus transfer. If START=1, the next load to I2DAT will generate the start condition followed by the serialized byte of data in I2DAT. The external processor loads byte data into I2DAT after setting the START Bit. The bus controller clears the START Bit during the ACK interval.

7.9.2 STOP

The external processor sets STOP = 1 to terminate a bus transfer. The bus controller clears the STOP Bit after completing the STOP condition. If the external processor sets the STOP Bit during a byte transfer, the STOP condition will be generated immediately following the ACK phase of the byte transfer. If no byte transfer is occurring when the STOP Bit is set, the STOP condition will be carried out immediately on the bus. Data should not be written to I2CS or I2DAT until the STOP Bit returns LOW.

7.9.3 LASTRD

To read data over the I2C compatible bus, a bus master floats the SDA line and issues clock pulses on the SCL line. After every eight bits, the master drives SDA low for one clock to indicate ACK. To signal the last byte of the read transfer, the master floats SDA at ACK time to instruct the slave to stop sending. This is controlled by the external processor by setting LastRD = 1 before reading the last byte of a read transfer. The bus controller clears the LastRD Bit at the end of the transfer (at ACK time).

7.9.4 BERR

This bit indicates a bus error. BERR = 1 indicates that there was bus contention, which results when an outside device drives the bus LOW when it should not, or when another bus master wins arbitration, taking control of the bus. BERR is cleared when external processor reads or writes the IDATA Register.

7.9.5 ACK

Every ninth SCL or a write transfer the slave indicates reception of the byte by asserting ACK. The bus controller floats SDA during this time, samples the SDA line, and updates the ACK Bit with the complement of the detected value. ACK = 1 indicates acknowledge, and ACK = 0 indicates not-acknowledge. The USB core updates the ACK Bit at the same time it sets DONE = 1. The ACK Bit should be ignored for read transfers on the bus.

7.9.6 DONE

The bus controller sets this bit whenever it completes a byte transfer, right after the ACK stage. The controller also generates an Interrupt Request when it sets the DONE Bit. The bus controller automatically clears the DONE Bit and the Interrupt Request bit whenever the external processor reads or writes the I2DAT Register.

7.10 I2DAT

This register is the I^2C-compatible Bus data register.

7.11 I2CTL

7.11.1 400KHZ

For I2C-compatible peripherals that support it, the I2C-compatible bus can run at 400 KHz. For compatibility, the bus powers-up at the 100-KHz frequency. If 400KHZ=0, the I2C-compatible bus operates at approximately 100 KHz. If 400KHZ=1, the I 2 C compatible bus operates at approximately 400 KHz. This bit is copied to the I2CCTL register bit 0, which is read-write to the external processor. Thus the I2C compatible bus speed is initially set by the EEPROM bit, and may be changed subsequently by the external processor.

7.12 USBFRAMEH/L

Every millisecond the host sends a SOF token indicating “Start Of Frame,” along with an 11-bit incrementing frame count. The SX2 copies the frame count into these registers at every SOF. One use of the frame count is to respond to the USB SYNC_FRAME Request. If the USB core detects a missing or garbled SOF, it generates an internal SOF and increments USBFRAMEL-USBRAMEH.

7.13 MICROFRAME

MICROFRAME contains a count 0–7 which indicates which of the 125 µs microframes last occurred. This register is active only when SX2 is operating at high speed (480 Mbits).

7.14 FNADDR

During the USB enumeration process, the host sends a device a unique 7-bit address, which the USB core copies into this register. There is normally no reason for the CPU to know its USB device address because the USB Core automatically responds only to its assigned address.

7.15 INTENABLE.

This register is used to enable various interrupts.

7.15.1 EP0TXD enables an interrupt when a packet is transferred from endpoint 0.

7.15.2 EP0RXD enables an interrupt when a packet is transferred to endpoint 0.

7.15.3 I2CDONE enables an interrupt when an I^2C-Compatible Bus transfer is completed.

7.15.4 ENUMOK enables an interrupt when the enumeration is completed.

7.15.5 SUSP/RES enables an interrupt when the SX2 detects a suspend or resume condition.

7.15.6 DMAEOT enables an interrupt when the multiple packet DMA is completed.

7.16 IRQ

This register contains the interrupt request flags. Only one of these bits will be set at any given time. Reading this register automatically clears the interrupt.

7.17 DMACOUNTER

This register is used to setup the multiple packet DMA feature. When the external processor writes a packet count number to this register, the SX2 will automatically count the number of packets transferred through USB. When the number of packets transferred equals the number in this register, the SX2 will interrupt the external processor with a DMAEOT. Note that only the number of

packets is counted, not the number of bytes transferred. If the packet size is 1024, then up to 256 kbytes can be transferred without processor intervention.

7.18 DESCR

This register address is used to access the 1024 byte descriptor FIFO. The external processor writes a byte to this address corresponding to the number of bytes of descriptor data to be written. The external processor then consecutively writes that number of bytes into the descriptor FIFO.

7.19 EP0BUF

This register address is used to access the 64 byte endpoint 0 FIFO. The external processor writes a byte to this address corresponding to the number of bytes of endpoint 0 data to be written. The external processor then consecutively writes that number of bytes into the endpoint FIFO.

8.0 Absolute Maximum Ratings

Storage Temperature..... –65°C to +150°C
Ambient Temperature with Power Supplied.....0°C to +70°C
Supply Voltage to Ground Potential.....–0.5V to +4.0V
DC Input Voltage to Any Pin.....TBD
DC Voltage Applied to Outputs in High Z State..... –0.5V to V_{CC}+0.5V
Power Dissipation..... TBD mW
Static Discharge Voltage..... >2000V
Latch-up Current..... >TBD mA

9.0 Operating Conditions

T_A (Ambient Temperature Under Bias).....0°C to +70°C
Supply Voltage.....+3.0V to +3.6V
Ground Voltage.....0V
F_{OSC} (Oscillator or Crystal Frequency).....24 MHz ± 100 ppm

10.0 DC Characteristics

Table 10-1. DC Characteristics

Parameter	Description	Conditions	Min.	Typ.	Max.	Unit
V_{CC}	Supply Voltage		3.0		3.6	V
V_{IH}	Input High Voltage		2		5.25	V
V_{IL}	Input Low Voltage		–0.5		0.8	V
I_I	Input Leakage Current	$0< V_{IN} < V_{CC}$			±10	µA
V_{OH}	Output Voltage High	I_{OUT} = 4 mA	2.4			V
V_{OL}	Output Low Voltage	I_{OUT} = –4 mA			0.4	V
I_{OH}	Output Current High				4	mA
I_{OL}	Output Current Low				4	mA
C_{IN}	Input Pin Capacitance				10	pF
I_{SUSP}	Suspend Current	Includes 1.5k internal pull-up		250	400	µA
I_{CC}	Supply Current	8051 running, connected to USB		128	TBD	mA

Table 10-2. USB Transceiver

USB 1.1/2.0 compliant.

11.0 AC Electrical Characteristics

11.1 USB Transceiver

USB 2.0 compliant in full and high speed.

11.2 Slave FIFO Synchronous Read

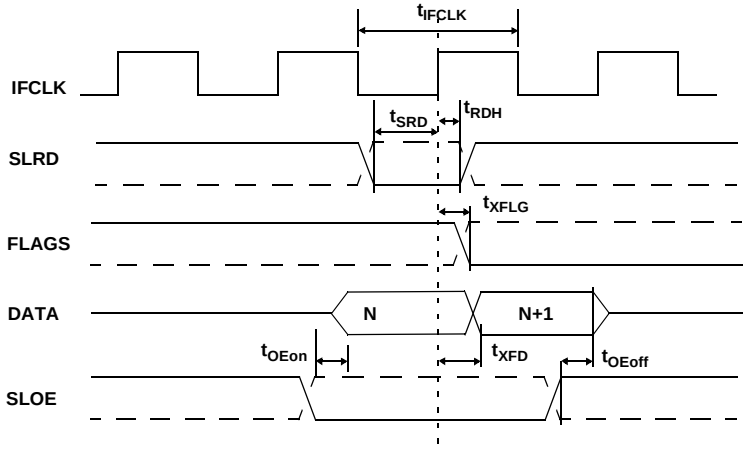

Figure 11-1. Slave FIFO Synchronous Read Timing Diagram

* dashed lines denote signals with programmable polarity

Table 11-1. Slave FIFO Synchronous Read Parameters[1]

Parameter	Description	Min.	Max.	Unit
t_{SRD}	SLRD to Clock Set-up Time	17.2		ns
t_{RDH}	Clock to SLRD Hold Time	0		ns
t_{OEon}	SLOE Turn-on to FIFO Data Valid		7.5	ns
t_{OEoff}	SLOE Turn-off to FIFO Data Hold		7.5	ns
t_{XFLG}	Clock to FLAGS Output Propagation Delay		5.3	ns
t_{XFD}	Clock to FIFO Data Output Propagation Delay		9.9	ns

Note:

1. IFCLK must not exceed 48 MHz

11.3 Slave FIFO Asynchronous Read

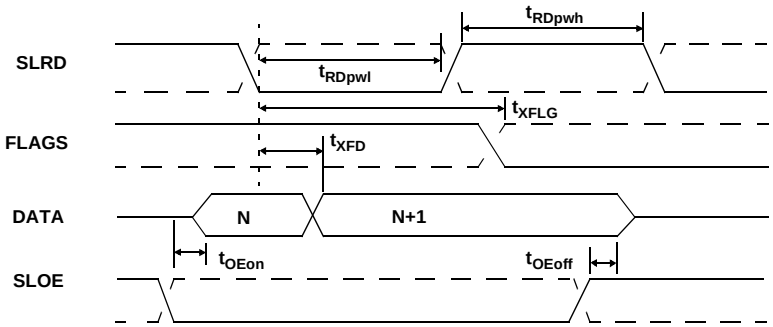

Figure 11-2. Slave FIFO Asynchronous Read Timing Diagram

* dashed lines denote signals with programmable polarity

Table 11-2. Slave FIFO Asynchronous Read Parameters[2]

Parameter	Description	Min.	Max.	Unit
t_{RDpwl}	SLRD Pulse Width Low	50		ns
t_{RDpwh}	SLRD Pulse Width High	50		ns
t_{XFLG}	SLRD to FLAGS Output Propagation Delay		70	ns
t_{XFD}	SLRD to FIFO Data Output Propagation Delay		11.2	ns
t_{OEon}	SLOE Turn-on to FIFO Data Valid		7.5	ns
t_{OEoff}	SLOE Turn-off to FIFO Data Hold		7.5	ns

Note:

2. Slave FIFO asynchronous parameter values are using internal IFCLK setting at 48MHz.

11.4 Command Read

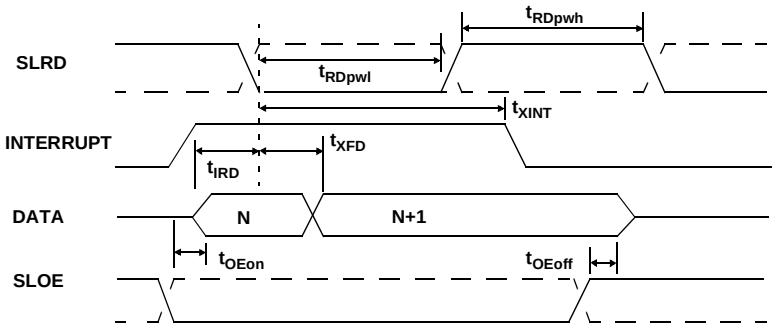

Figure 11-3. Command Read Timing Diagram

* dashed lines denote signals with programmable polarity

Table 11-3. Command Read Parameters

Parameter	Description	Min.	Max.	Unit
t_{RDpwl}	SLRD Pulse Width Low	50		ns
t_{RDpwh}	SLRD Pulse Width High	50		ns
t_{IRD}	INTERRUPT to SLRD	TBD		ns
t_{XINT}	SLRD to INTERRUPT		TBD	ns
t_{XFD}	SLRD to FIFO Data Output Propagation Delay		11.2	ns
t_{OEon}	SLOE Turn-on to FIFO Data Valid		7.5	ns
t_{OEoff}	SLOE Turn-off to FIFO Data Hold		7.5	ns

11.5 Slave FIFO Synchronous Write

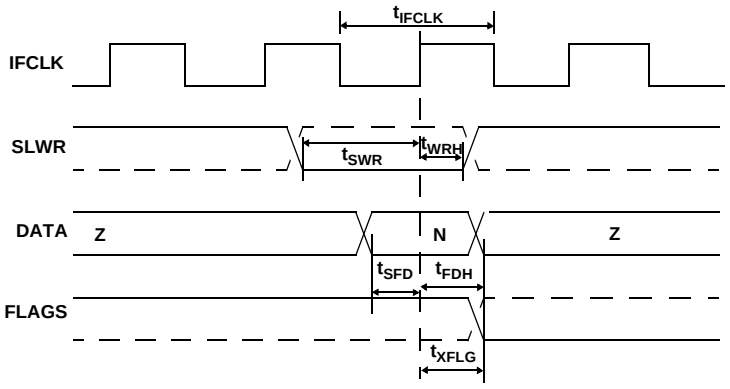

Figure 11-4. Slave FIFO Synchronous Write Timing Diagram

* dashed lines denote signals with programmable polarity

Table 11-4. Slave FIFO Synchronous Write Parameters[1]

Parameter	Description	Min.	Max.	Unit
t_{SWR}	SLWR to Clock Setup Time	10.4		ns
t_{WRH}	Clock to SLWR Hold Time	0		ns
t_{SFD}	FIFO Data to Clock Setup Time	5.1		ns
t_{FDH}	Clock to FIFO Data Hold Time	0		ns
t_{XFLG}	Clock to FLAGS Output Propagation Time		5.3	ns

11.6 Slave FIFO Asynchronous Write

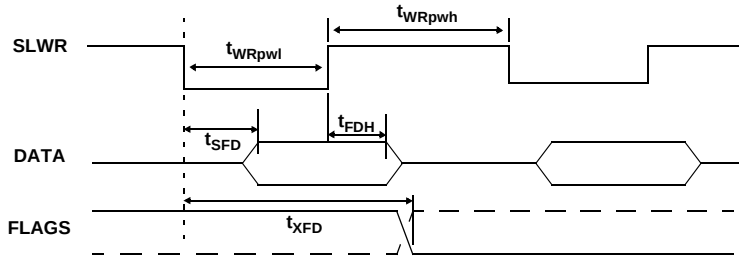

Figure 11-5. Slave FIFO Asynchronous Write Timing Diagram

* dashed lines denote signals with programmable polarity

Table 11-5. Slave FIFO Asynchronous Write Parameters[2]

Parameter	Description	Min.	Max.	Unit
t_{WRpwl}	SLWR Pulse Low	50		ns
t_{WRpwh}	SLWR Pulse High	70		ns
t_{SFD}	SLWR to FIFO DATA Setup Time	2.7		ns
t_{FDH}	FIFO DATA to SLWR Hold Time	2.7		ns
t_{XFD}	SLWR to FLAGS Output Propagation Delay		70	ns

11.7 Command Write

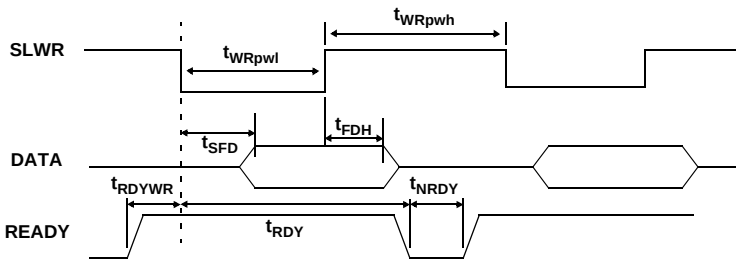

Figure 11-6. Command Write Timing Diagram

* dashed lines denote signals with programmable polarity

Table 11-6. Command Write Parameters

Parameter	Description	Min.	Max.	Unit
t_{WRpwl}	SLWR Pulse Low	50		ns
t_{WRpwh}	SLWR Pulse High	70		ns
t_{SFD}	SLWR to FIFO DATA Setup Time	2.7		ns
t_{FDH}	FIFO DATA to SLWR Hold Time	2.7		ns
t_{RDYRD}	READY to SLWR Time	TBD		ns
t_{RDY}	READY Pulse		TBD	ns
t_{NRDY}	READY to next READY		TBD	ns

11.8 Slave FIFO Synchronous Packet End Strobe

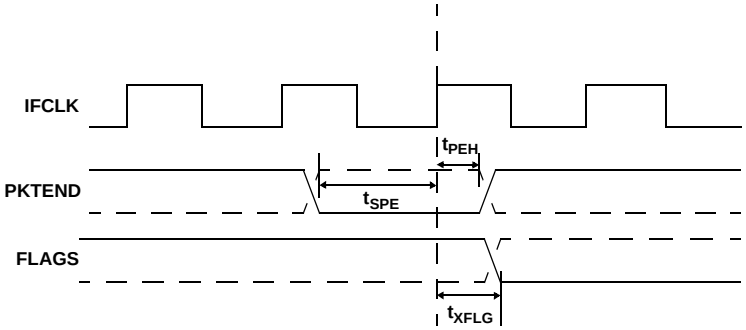

Figure 11-7. Slave FIFO Synchronous Packet End Strobe Timing Diagram

* dashed lines denote signals with programmable polarity

Table 11-7. Slave FIFO Synchronous Packet End Strobe Parameters[1]

Parameter	Description	Min.	Max.	Unit
t_{SPE}	PKTEND to Clock Setup Time	10.4		ns
t_{PEH}	Clock to PKTEND Hold Time	0		ns
t_{XFLG}	Clock to FLAGS Output Propagation Delay		5.3	ns

11.9 Slave FIFO Asynchronous Packet End Strobe

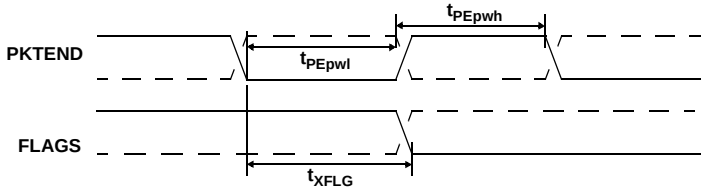

Figure 11-8. Slave FIFO Asynchronous Packet End Strobe Timing Diagram

* dashed lines denote signals with programmable polarity

Table 11-8. Slave FIFO Asynchronous Packet End Strobe Parameters[3]

Parameter	Description	Min.	Max.	Unit
t_{PEpwl}	PKTEND Pulse Width Low	50		ns
t_{PWpwh}	PKTEND Pulse Width High	50		ns
t_{XFLG}	PKTEND to FLAGS Output Propagation Delay		70	ns

Note:

3. Slave FIFO asynchronous parameter values are using internal IFCLK setting at 48MHz.Slave FIFO Output Enable.

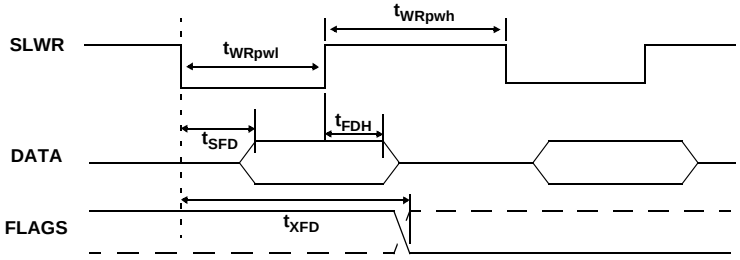

Figure 11-9. Slave FIFO Output Enable Timing Diagram

* dashed lines denote signals with programmable polarity

Table 11-9. Slave FIFO Output Enable Parameters

Parameter	Description	Min.	Max.	Unit
t_{OEon}	SLOE assert to FIFO DATA Output		7.5	ns
t_{OEoff}	SLOE de-assert to FIFO DATA Hold		7.5	ns

11.10 Slave FIFO Address to Flags/Data

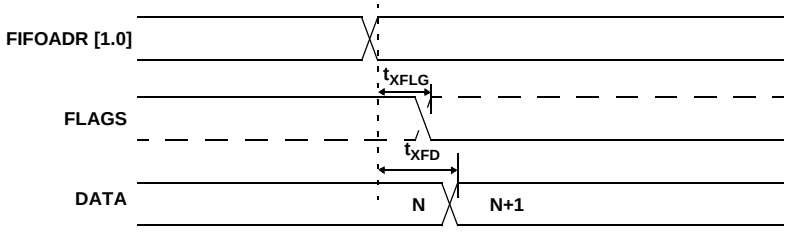

Figure 11-10. Slave FIFO Address to Flags/Data Timing Diagram

* dashed lines denote signals with programmable polarity

Table 11-10. Slave FIFO Address to Flags/Data Parameters

Parameter	Description	Min.	Max.	Unit
t_{XFLG}	FIFOADR[1:0] to FLAGS Output Propagation Delay		9.1	ns
t_{XFD}	FIFOADR[1:0] to FIFODATA Output Propagation Delay		13.9	ns

11.11 Slave FIFO Synchronous Address

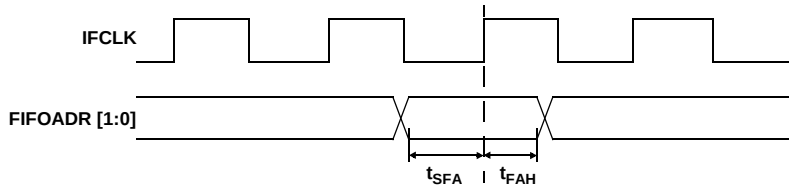

Figure 11-11. Slave FIFO Synchronous Address Timing Diagram

Table 11-11. Slave FIFO Synchronous Address Parameters[1]

Parameter	Description	Min.	Max.	Unit
t_{SFA}	FIFOADR[1:0] to Clock Setup Time	19.2		ns
t_{FAH}	Clock to FIFOADR[1:0] Hold Time	0		ns

11.12 Slave FIFO Asynchronous Address

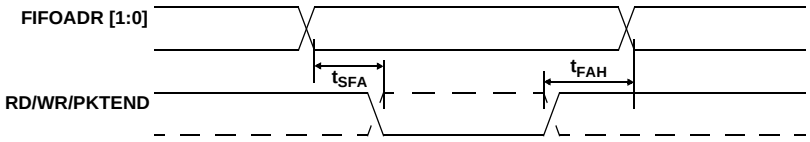

Figure 11-12. Slave FIFO Asynchronous Address Timing Diagram

* dashed lines denote signals with programmable polarity

Table 11-12. Slave FIFO Asynchronous Address Parameters[2]

Parameter	Description	Min.	Max.	Unit
t_{SFA}	FIFOADR[1:0] to RD/WR/PKTEND Setup Time	0		ns
t_{FAH}	RD/WR/PKTEND to FIFOADR[1:0] Hold Time	0		ns

12.0 Ordering Information

Table 12-1. Ordering Information

Ordering Code	Package Type			
CY7C68001-56PVC	56 SSOP			
CY36XX	EZ-USB SX2 Xcelerator Development Kit			

Document #: 38-01110-**

13.0 Package Diagrams

The SX2 is available in the 56-pin SSOP package.

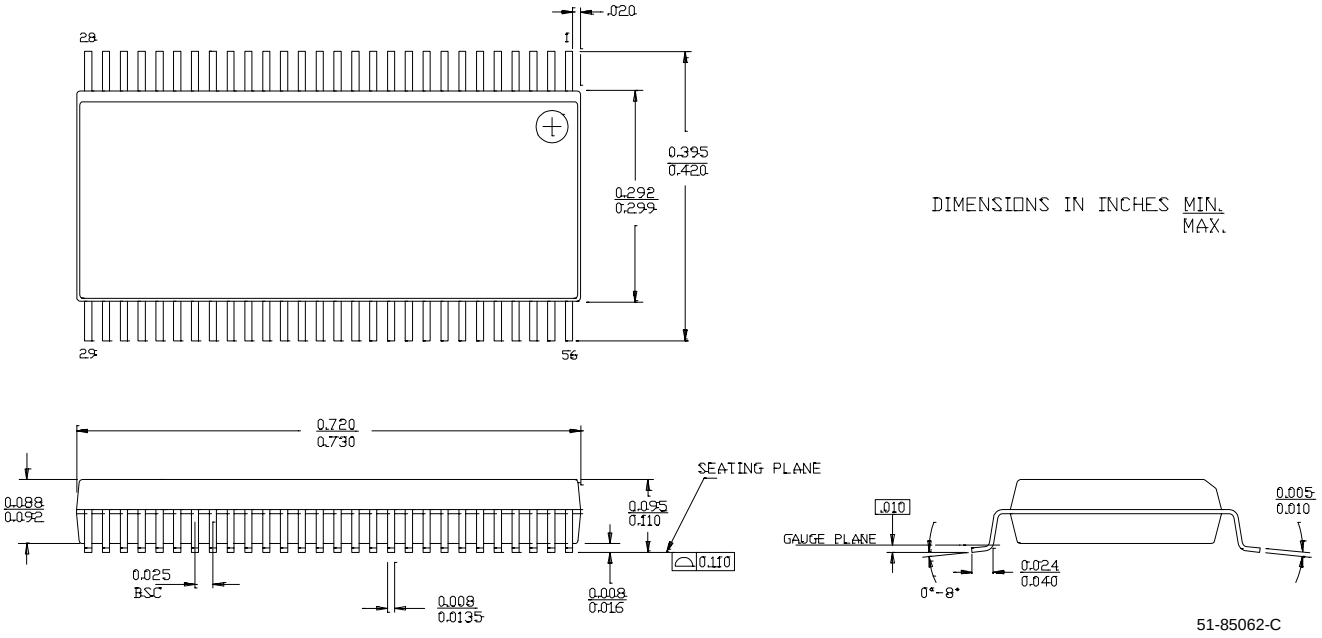

Figure 13-1. 56-Lead Shrunk Small Outline Package

14.0 Document Revision History

Table 14-1. Revision History

Document Title: CY7C68001 EZ-USB SX2 Document Number: 38-01110				
REV.	**ECN NUMBER**	**ISSUE DATE**	**ORIG. OF CHANGE**	**DESCRIPTION OF CHANGE**
**	3422	12/05/00	OTT	1. New Data Sheet