



CY7C68001

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



TABLE OF CONTENTS

1.0 EZ-USB SX2™ FEATURES	7
1.1 Introduction	7
1.2 Features	7
1.3 Block Diagram	7
2.0 APPLICATIONS	8
2.1 System Diagram	8
3.0 FUNCTIONAL OVERVIEW	9
3.1 USB Signaling Speed	9
3.2 Buses	9
3.3 Boot Methods	9
3.3.1 EEPROM Organization	9
3.3.2 Default Enumeration	10
3.4 Interrupt System	10
3.4.1 Architecture	10
3.4.2 ITENABLE Register Bit Definition	10
3.5 Resets and Wakeup	11
3.5.1 Reset	11
3.5.2 USB Reset	11
3.5.3 Wakeup	11
3.6 Endpoint RAM	11
3.6.1 Size	11
3.6.2 Organization	11
3.6.3 Endpoint Configurations (High-speed Mode)	12
3.6.4 Default Endpoint Memory Configuration	12
3.7 External Interface	12
3.7.1 Architecture	12
3.7.2 Control Signals	13
3.7.3 IFCLK	13
3.7.4 FIFO Access	13
3.7.5 FIFO Flag Pins Configuration	14
3.7.6 Default FIFO Programmable Flag Set-up	14
3.7.7 FIFO Programmable Flag (PF) Set-up	14
3.7.8 Command Protocol	14
4.0 ENUMERATION	15
4.1 Standard Enumeration	15
4.2 Default Enumeration	16
5.0 ENDPOINT 0	16
6.0 PIN ASSIGNMENTS	17
6.1 56-pin SSOP	17
6.2 56-pin QFN	18
6.3 CY7C68001 Pin Descriptions	19
7.0 REGISTER SUMMARY	21
7.1 IFCONFIG Register 0x01	22
7.1.1 Bit 7: IFCLKSRC	22
7.1.2 Bit 6: 3048MHZ	22
7.1.3 Bit 5: IFCLKOE	22

TABLE OF CONTENTS (continued)

7.1.4 Bit 4: IFCLKPOL	22
7.1.5 Bit 3: ASYNC	22
7.1.6 Bit 2: STANDBY	22
7.1.7 Bit 1: FLAGD/CS#	22
7.1.8 Bit 0: DISCON	22
7.2 FLAGSAB/FLAGSCD Registers 0x02/0x03	22
7.3 POLAR Register 0x04	23
7.3.1 Bit 7: WUPOL	23
7.3.2 Bit 5: PKTEND	23
7.3.3 Bit 4: SLOE	24
7.3.4 Bit 3: SLRD	24
7.3.5 SLWR Bit 2	24
7.3.6 EF Bit 1	24
7.3.7 FF Bit 0	24
7.4 REVID Register 0x05	24
7.5 EPxCFG Register 0x06–0x09	24
7.5.1 Bit 7: VALID	24
7.5.2 Bit 6: DIR	24
7.5.3 Bit [5,4]: TYPE1, TYPE0	24
7.5.4 Bit 3: SIZE	25
7.5.5 Bit 2: STALL	25
7.5.6 Bit [1,0]: BUF1, BUF0	25
7.6 EPxPKTLENH/L Registers 0x0A–0x11	25
7.6.1 Bit 7: INFM1 EPxPKTLENH.7	26
7.6.2 Bit 6: OEP1 EPxPKTLENH.6	26
7.6.3 Bit 5: ZEROLEN EPxPKTLENH.5	26
7.6.4 Bit 4: WORDWIDE EPxPKTLENH.4	26
7.6.5 Bit [2..0]: PL[X:0] Packet Length Bits	26
7.7 EPxPFH/L Registers 0x12–0x19	26
7.7.1 DECIS EPxPFH.7	26
7.7.2 PKSTAT EPxPFH.6	26
7.7.3 IN:PKTS(2:0)/OUT:PFC[12:10] EPxPFH[5:3]	27
7.8 EPxISOINPKTS Registers 0x1A–0x1D	27
7.9 EPxxFLAGS Registers 0x1E–0x1F	27
7.9.1 EPxPF Bit 6, Bit 2	28
7.9.2 EPxEF Bit 5, Bit 1	28
7.9.3 EPxFF Bit 4, Bit 0	28
7.10 INPKTEND/FLUSH Register 0x20	28
7.11 USBFRAMEH/L Registers 0x2A, 0x2B	28
7.12 MICROFRAME Registers 0x2C	29
7.13 FNADDR Register 0x2D	29
7.14 INTENABLE Register 0x2E	29
7.14.1 SETUP Bit 7	29
7.14.2 EP0BUF Bit 6	29
7.14.3 FLAGS Bit 5	29
7.14.4 ENUMOK Bit 2	29
7.14.5 BUSACTIVITY Bit 1	29
7.14.6 READY Bit 0	29

TABLE OF CONTENTS (continued)

7.15 DESC Register 0x30	30
7.16 EP0BUF Register 0x31	30
7.17 SETUP Register 0x32	30
7.18 EP0BC Register 0x33	30
8.0 ABSOLUTE MAXIMUM RATINGS	31
9.0 OPERATING CONDITIONS	31
10.0 DC ELECTRICAL CHARACTERISTICS	31
11.0 AC ELECTRICAL CHARACTERISTICS	32
11.1 USB Transceiver	32
11.2 Command Interface	32
11.2.1 Command Synchronous Read	32
11.2.2 Command Synchronous Write	33
11.2.3 Command Asynchronous Read	34
11.2.4 Command Asynchronous Write	34
11.3 FIFO Interface	35
11.3.1 Slave FIFO Synchronous Read	35
11.3.2 Slave FIFO Synchronous Write	36
11.3.3 Slave FIFO Synchronous Packet End Strobe	37
11.3.4 Slave FIFO Synchronous Address	37
11.3.5 Slave FIFO Asynchronous Read	38
11.3.6 Slave FIFO Asynchronous Write	38
11.3.7 Slave FIFO Asynchronous Packet End Strobe	39
11.3.8 Slave FIFO Asynchronous Address	39
11.4 Slave FIFO Address to Flags/Data	39
11.5 Slave FIFO Output Enable	40
11.6 Sequence Diagram	40
11.6.1 Single and Burst Synchronous Read Example	40
11.6.2 Single and Burst Synchronous Write	41
11.6.3 Sequence Diagram of a Single and Burst Asynchronous Read	42
11.6.4 Sequence Diagram of a Single and Burst Asynchronous Write	43
12.0 DEFAULT DESCRIPTOR	44
13.0 GENERAL PCB LAYOUT GUIDELINES	47
14.0 QUAD FLAT PACKAGE NO LEADS (QFN) PACKAGE DESIGN NOTES	47
15.0 ORDERING INFORMATION	48
16.0 PACKAGE DIAGRAMS	48
16.1 56-pin SSOP Package	48
16.2 56-pin QFN Package	49
17.0 DOCUMENT REVISION HISTORY	50

LIST OF FIGURES

FIGURE 1-1. BLOCK DIAGRAM	7
FIGURE 2-1. EXAMPLE USB SYSTEM DIAGRAM	8
FIGURE 3-1. ENDPOINT CONFIGURATION	12
FIGURE 6-1. CY7C68001 56-PIN SSOP PIN ASSIGNMENT	17
FIGURE 6-2. CY7C68001 56-PIN QFN ASSIGNMENT	18
FIGURE 11-1. COMMAND SYNCHRONOUS READ TIMING DIAGRAM.....	32
FIGURE 11-2. COMMAND SYNCHRONOUS WRITE TIMING DIAGRAM	33
FIGURE 11-3. COMMAND ASYNCHRONOUS READ TIMING DIAGRAM	34
FIGURE 11-4. COMMAND ASYNCHRONOUS WRITE TIMING DIAGRAM.....	34
FIGURE 11-5. SLAVE FIFO SYNCHRONOUS READ TIMING DIAGRAM.....	35
FIGURE 11-6. SLAVE FIFO SYNCHRONOUS WRITE TIMING DIAGRAM	36
FIGURE 11-7. SLAVE FIFO SYNCHRONOUS PACKET END STROBE TIMING DIAGRAM	37
FIGURE 11-8. SLAVE FIFO SYNCHRONOUS ADDRESS TIMING DIAGRAM	37
FIGURE 11-9. SLAVE FIFO ASYNCHRONOUS READ TIMING DIAGRAM	38
FIGURE 11-10. SLAVE FIFO ASYNCHRONOUS WRITE TIMING DIAGRAM.....	38
FIGURE 11-11. SLAVE FIFO ASYNCHRONOUS PACKET END STROBE TIMING DIAGRAM.....	39
FIGURE 11-12. SLAVE FIFO ASYNCHRONOUS ADDRESS TIMING DIAGRAM.....	39
FIGURE 11-13. SLAVE FIFO ADDRESS TO FLAGS/DATA TIMING DIAGRAM	39
FIGURE 11-14. SLAVE FIFO OUTPUT ENABLE TIMING DIAGRAM	40
FIGURE 11-15. SLAVE FIFO SYNCHRONOUS READ SEQUENCE AND TIMING DIAGRAM.....	40
FIGURE 11-16. SLAVE FIFO SYNCHRONOUS SEQUENCE OF EVENTS DIAGRAM.....	40
FIGURE 11-17. SLAVE FIFO SYNCHRONOUS WRITE SEQUENCE AND TIMING DIAGRAM	41
FIGURE 11-18. SLAVE FIFO ASYNCHRONOUS READ SEQUENCE AND TIMING DIAGRAM.....	42
FIGURE 11-19. SLAVE FIFO ASYNCHRONOUS READ SEQUENCE OF EVENTS DIAGRAM.....	42
FIGURE 11-20. SLAVE FIFO ASYNCHRONOUS WRITE SEQUENCE AND TIMING DIAGRAM.....	43
FIGURE 14-1. CROSSSECTION OF THE AREA UNDERNEATH THE QFN PACKAGE	47
FIGURE 14-2. (A) PLOT OF THE SOLDER MASK (WHITE AREA)	47
FIGURE 14-2. (B) X-RAY IMAGE OF THE ASSEMBLY	47
FIGURE 16-1. 56-LEAD SHRUNK SMALL OUTLINE PACKAGE	48
FIGURE 16-2. LF56A 56-PIN QFN PACKAGE	49

LIST OF TABLES

TABLE 3-1. DESCRIPTOR LENGTH SET TO 0X06: DEFAULT ENUMERATION	9
TABLE 3-2. DESCRIPTOR LENGTH NOT SET TO 0X06	9
TABLE 3-3. FIFO ADDRESS LINES SETTING	13
TABLE 3-4. COMMAND ADDRESS BYTE	14
TABLE 3-5. COMMAND DATA BYTE ONE	14
TABLE 3-6. COMMAND DATA BYTE TWO	14
TABLE 3-7. COMMAND ADDRESS WRITE BYTE	15
TABLE 3-8. COMMAND DATA WRITE BYTE ONE	15
TABLE 3-9. COMMAND DATA WRITE BYTE TWO	15
TABLE 3-10. COMMAND ADDRESS READ BYTE	15
TABLE 6-1. SX2 PIN DESCRIPTIONS	19
TABLE 7-1. SX2 REGISTER SUMMARY	21
TABLE 7-2. FIFO FLAG 4-BIT CODING	23
TABLE 7-3. ENDPOINT TYPE	25
TABLE 7-4. ENDPOINT BUFFERING	25
TABLE 7-5. PKTS BITS	27
TABLE 7-6. EPXISOINPKTS	27
TABLE 10-1. DC CHARACTERISTICS	31
TABLE 11-1. COMMAND SYNCHRONOUS READ PARAMETERS WITH INTERNALLY SOURCED IFCLK	32
TABLE 11-2. COMMAND SYNCHRONOUS READ WITH EXTERNALLY SOURCED IFCLK	32
TABLE 11-3. COMMAND SYNCHRONOUS WRITE PARAMETERS WITH INTERNALLY SOURCED IFCLK	33
TABLE 11-4. COMMAND SYNCHRONOUS WRITE PARAMETERS WITH EXTERNALLY SOURCED IFCLK	33
TABLE 11-5. COMMAND READ PARAMETERS	34
TABLE 11-6. COMMAND WRITE PARAMETERS	34
TABLE 11-7. SLAVE FIFO SYNCHRONOUS READ WITH INTERNALLY SOURCED IFCLK	35
TABLE 11-8. SLAVE FIFO SYNCHRONOUS READ WITH EXTERNALLY SOURCED IFCLK	35
TABLE 11-9. SLAVE FIFO SYNCHRONOUS WRITE PARAMETERS WITH INTERNALLY SOURCED IFCLK	36
TABLE 11-10. SLAVE FIFO SYNCHRONOUS WRITE PARAMETERS WITH EXTERNALLY SOURCED IFCLK	36
TABLE 11-11. SLAVE FIFO SYNCHRONOUS PACKET END STROBE PARAMETERS, INTERNALLY SOURCED IFCLK	37
TABLE 11-12. SLAVE FIFO SYNCHRONOUS PACKET END STROBE PARAMETERS, EXTERNALLY SOURCED IFCLK	37
TABLE 11-13. SLAVE FIFO SYNCHRONOUS ADDRESS PARAMETERS	37
TABLE 11-14. SLAVE FIFO ASYNCHRONOUS READ PARAMETERS	38
TABLE 11-15. SLAVE FIFO ASYNCHRONOUS WRITE PARAMETERS WITH INTERNALLY SOURCED IFCLK	38
TABLE 11-16. SLAVE FIFO ASYNCHRONOUS PACKET END STROBE PARAMETERS	39
TABLE 11-17. SLAVE FIFO ASYNCHRONOUS ADDRESS PARAMETERS	39
TABLE 11-18. SLAVE FIFO ADDRESS TO FLAGS/DATA PARAMETERS	39
TABLE 11-19. SLAVE FIFO OUTPUT ENABLE PARAMETERS	40
TABLE 15-1. ORDERING INFORMATION	48

1.0 EZ-USB SX2™ Features

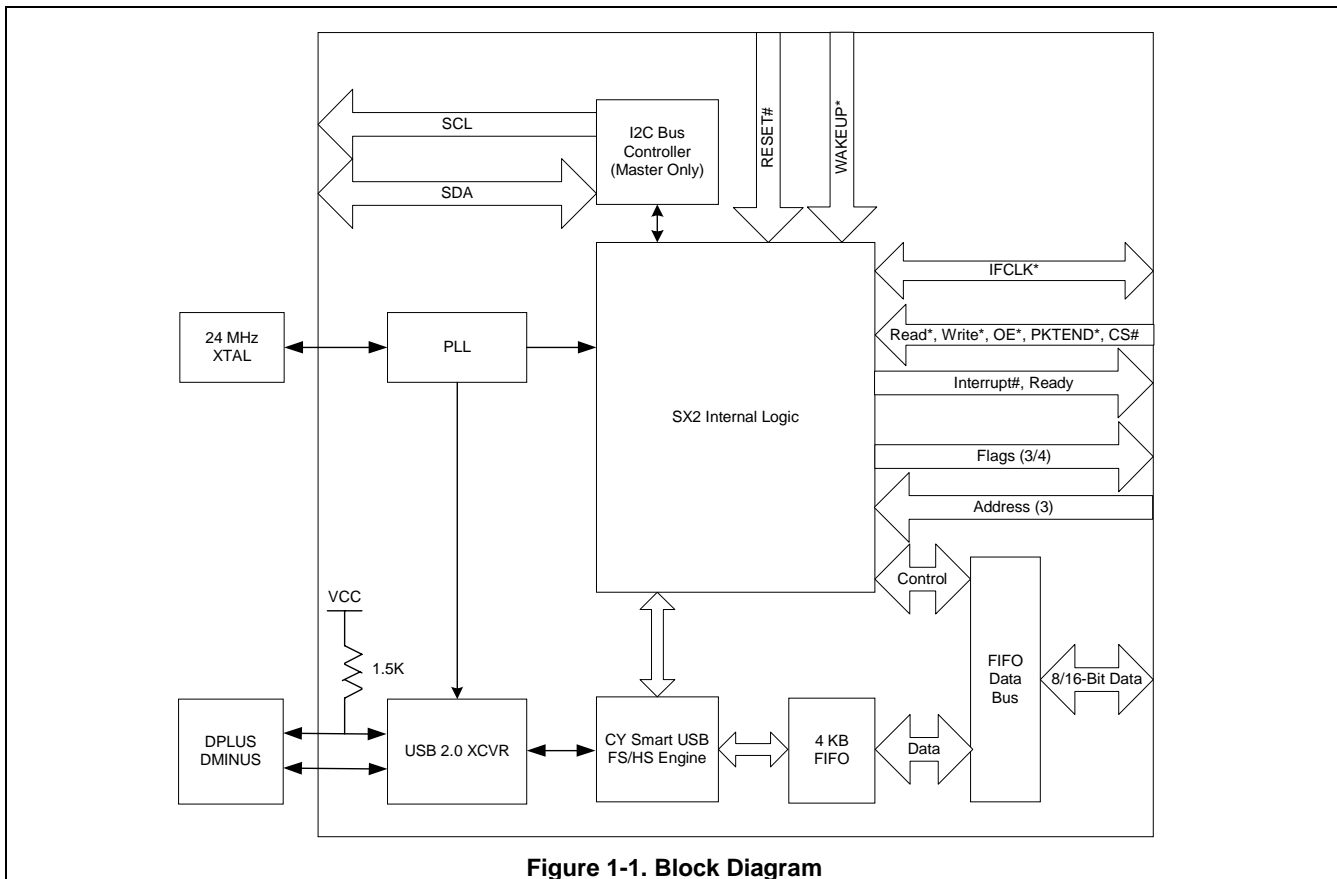
1.1 Introduction

The EZ-USB SX2™ USB interface device is designed to work with any external master, such as standard microprocessors, DSPs, ASICs, and FPGAs to enable USB 2.0 support for any peripheral design. SX2 has a built-in USB transceiver and Serial Interface Engine (SIE), along with a command decoder for sending and receiving USB data. The controller has four endpoints that share a 4-KB FIFO space for maximum flexibility and throughput, as well as Control Endpoint 0. SX2 has three address pins and a selectable 8- or 16- bit data bus for command and data input or output.

1.2 Features

- USB 2.0-certified compliant
- Operates at high (480 Mbps) or full (12 Mbps) speed
- Supports Control Endpoint 0:
 - Used for handling USB device requests
- Supports four configurable endpoints that share a 4-KB FIFO space
 - Endpoints 2, 4, 6, 8 for application-specific control and data
- Standard 8- or 16-bit external master interface
 - Glueless interface to most standard microprocessors DSPs, ASICs, and FPGAs
 - Synchronous or Asynchronous interface
- Integrated phase-locked loop (PLL)
- 3.3V operation, 5V tolerant I/Os
- 56-pin SSOP and QFN package
- Complies with most device class specifications

1.3 Block Diagram



2.0 Applications

- DSL modems
- ATA interface
- Memory card readers
- Legacy conversion devices
- Cameras
- Scanners
- Home PNA
- Wireless LAN
- MP3 players
- Networking
- Printers

The "Reference Designs" section of the Cypress web site provides additional tools for typical USB applications. Each reference design comes complete with firmware source code and object code, schematics, and documentation. Please see the Cypress web site at www.cypress.com.

2.1 System Diagram

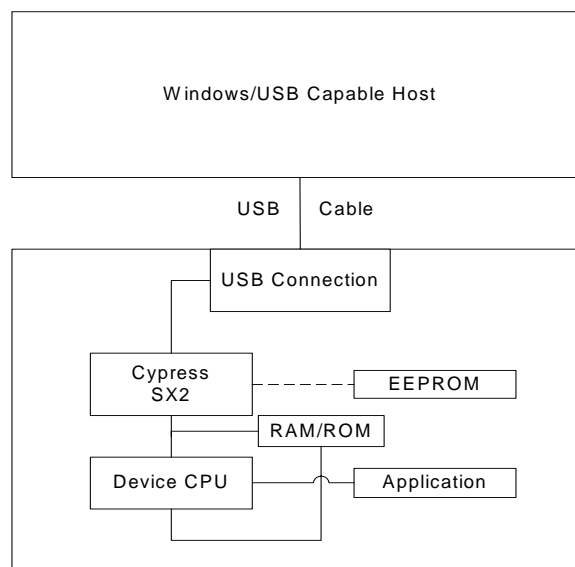


Figure 2-1. Example USB System Diagram

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/s
- High-speed, with a signaling bit rate of 480 Mbits/s.

SX2 does not support the low-speed signaling rate of 1.5 Mbits/s.

3.2 Buses

SX2 features:

- A selectable 8- or 16-bit bidirectional data bus
- An address bus for selecting the FIFO or Command Interface.

3.3 Boot Methods

During the power-up sequence, internal logic of the SX2 checks for the presence of an I²C-compatible EEPROM.^[1,2] If it finds an EEPROM, it will boot off the EEPROM. When the presence of an EEPROM is detected, the SX2 checks the value of first byte. If the first byte is found to be a 0xC4, the SX2 loads the next two bytes into the IFCONFIG and POLAR registers, respectively. If the fourth byte is also 0xC4, the SX2 enumerates using the descriptor in the EEPROM, then signals to the external master when enumeration is complete via an ENUMOK interrupt (Section 3.4). If no EEPROM is detected, the SX2 relies on the external master for the descriptors. Once this descriptor information is received from the external master, the SX2 will connect to the USB bus and enumerate.

3.3.1 EEPROM Organization

The valid sequence of bytes in the EEPROM are displayed below.

Table 3-1. Descriptor Length Set to 0x06: Default Enumeration

Byte Index	Description
0	0xC4
1	IFCONFIG
2	POLAR
3	0xC4
4	Descriptor Length (LSB):0x06
5	Descriptor Length (MSB): 0x00
6	VID (LSB)
7	VID (MSB)
8	PID (LSB)
9	PID (MSB)
10	DID (LSB)
11	DID (MSB)

Table 3-2. Descriptor Length Not Set to 0x06

Byte Index	Description
0	0xC4
1	IFCONFIG
2	POLAR
3	0xC4
4	Descriptor Length (LSB)
5	Descriptor Length (MSB)
6	Descriptor[0]
7	Descriptor[1]
8	Descriptor[2]
..
..
..

- **0xC4:** This initial byte tells the SX2 that this is a valid EEPROM with configuration information.
- **IFCONFIG:** This byte contains the settings for the IFCONFIG register. The IFCONFIG register bits are defined in Section 7.1. If the external master requires an interface configuration different from the default, that interface can be specified in this byte.
- **POLAR:** This byte contains the polarities of the interface signals. The POLAR register bits are defined in Section 7.3. If the external master requires signal polarities different from the default, those polarities can be specified in this byte.
- **Descriptor:** This next byte determines whether or not the SX2 loads the descriptor from the EEPROM. If this byte = 0xC4, the SX2 will load the descriptor starting with the next byte. If this byte does not equal 0xC4, the SX2 will wait for descriptor information from the external master.

Notes:

1. Because there is no direct way to detect which EEPROM type (single or double address) is connected, SX2 uses the EEPROM address pins A2, A1, and A0 to determine whether to send out one or two bytes of address. Single-byte address EEPROMs (24LC01, etc.) should be strapped to address 000 and double-byte EEPROMs (24LC64, etc.) should be strapped to address 001.
2. The SCL and SDA pins must be pulled up for this detection method to work properly, even if an EEPROM is not connected. Typical pull-up values are 2.2K – 10K Ohms.

- **Descriptor Length:** The next two bytes indicate the length of the descriptor contained in the EEPROM. The length is loaded least significant byte (LSB) first, then most significant byte (MSB).
- **Byte 7 Starts Descriptor Information:** The descriptor can be a maximum of 500 bytes.

3.3.2 Default Enumeration

An optional default descriptor can be used to simplify enumeration. Only the Vendor ID (VID), Product ID (PID), and Device ID (DID) need to be loaded by the SX2 for it to enumerate with this default set-up. This information is either loaded from an EEPROM in the case when the presence of an EEPROM (*Table 3-1*) is detected, or the external master may simply load a VID, PID, and DID when no EEPROM is present. In this default enumeration, the SX2 uses the in-built default descriptor (refer to Section 12.0).

If the descriptor length loaded from the EEPROM is 6, SX2 will load a VID, PID, and DID from the EEPROM and enumerate. The VID, PID, and DID are loaded LSB, then MSB. For example, if the VID, PID, and DID are 0x0547, 0x1002, and 0x0001, respectively, then the bytes should be stored as:

- 0x47, 0x05, 0x02, 0x10, 0x01, 0x00.

If there is no EEPROM, SX2 will wait for the external master to provide the descriptor information. To use the default descriptor, the external master must write to the appropriate register (0x30) with descriptor length equal to 6 followed by the VID, PID, and DID. Refer to Section 4.2 for further information on how the external master may load the values.

The default descriptor enumerates four endpoints as listed in the following page:

- Endpoint 2: Bulk out, 512 bytes in high-speed mode, 64 bytes in full-speed mode
- Endpoint 4: Bulk out, 512 bytes in high-speed mode, 64 bytes in full-speed mode
- Endpoint 6: Bulk in, 512 bytes in high-speed mode, 64 bytes in full-speed mode
- Endpoint 8: Bulk in, 512 bytes in high-speed mode, 64 bytes in full-speed mode.

The entire default descriptor is listed in Section 12.0 of this data sheet.

3.4 Interrupt System

3.4.1 Architecture

The SX2 provides an output signal that indicates to the external master that the SX2 has an interrupt condition, or that the data from a register read request is available. The SX2 has six interrupt sources: SETUP, EP0BUF, FLAGS, ENUMOK, BUSACTIVITY, and READY. Each interrupt can be enabled or disabled by setting or clearing the corresponding bit in the INTENABLE register.

When an interrupt occurs, the INT# pin will be asserted, and the corresponding bit will be set in the Interrupt Status Byte. The external master reads the Interrupt Status Byte by strobing SLRD/SLOE. This presents the Interrupt Status Byte on the lower portion of the data bus (FD[7:0]). Reading the Interrupt Status Byte automatically clears the interrupt. Only one interrupt request will occur at a time; the SX2 buffers multiple pending interrupts.

If the external master has initiated a register read request, the SX2 will buffer interrupts until the external master 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. Following is a description of this INTENABLE register.

3.4.2 INTENABLE Register Bit Definition

Bit 7: SETUP

If this interrupt is enabled, and the SX2 receives a set-up packet from the USB host, the SX2 asserts the INT# pin and sets bit 7 in the Interrupt Status Byte. This interrupt only occurs if the set-up request is not one that the SX2 automatically handles. For complete details on how to handle the SETUP interrupt, refer to Section 5.0 of this data sheet.

Bit 6: EP0BUF

If this interrupt is enabled, and the Endpoint 0 buffer becomes available to the external master for read or write operations, the SX2 asserts the INT# pin and sets bit 6 in the Interrupt Status Byte. This interrupt is used for handling the data phase of a set-up request. For complete details on how to handle the EP0BUF interrupt, refer to Section 5.0 of this data sheet.

Bit 5: FLAGS

If this interrupt is enabled, and any OUT endpoint FIFO's state changes from empty to not-empty, the SX2 asserts the INT# pin and sets bit 5 in the Interrupt Status Byte. This is an alternate way to monitor the status of OUT endpoint FIFOs instead of using the FLAGA-FLAGD pins, and can be used to indicate when an OUT packet has been received from the host.

Bit 4: ENUMOK

If this interrupt is enabled and the SX2 receives a SET_CONFIGURATION request from the USB host, the SX2 asserts the INT# pin and sets bit 2 in the Interrupt Status Byte. This event signals the completion of the SX2 enumeration process.

Bit 1: BUSACTIVITY

If this interrupt is enabled, and the SX2 detects either an absence or resumption of activity on the USB bus, the SX2 asserts the INT# pin and sets bit 1 in the Interrupt Status Byte. This usually indicates that the USB host is either suspending or resuming or that a self-powered device has been plugged in or unplugged. If the SX2 is bus-powered, the external master must put the SX2 into a low-power mode after detecting a USB suspend condition to be USB-compliant.

Bit 0: READY

If this interrupt is enabled, bit 0 in the Interrupt Status Byte is set when the SX2 has powered up and performed a self-test. The external master should always wait for this interrupt before trying to read or write to the SX2, unless an external EEPROM with a valid descriptor is present. If an external EEPROM with a valid descriptor is present, the ENUMOK interrupt will occur instead of the READY interrupt after power up. A READY interrupt will also occur if the SX2 is awakened from a low-power mode via the WAKEUP pin. This READY interrupt indicates that the SX2 is ready for commands or data.

3.5 Resets and Wakeup**3.5.1 Reset**

An input pin (RESET#) resets the chip. The internal PLL stabilizes approximately 7.6 ms after V_{CC} has reached 3.3V. Typically, an external RC network ($R = 100\text{ K Ohms}$, $C = 0.1\text{ }\mu\text{f}$) is used to provide the RESET# signal.

3.5.2 USB Reset

When the SX2 detects a USB Reset condition on the USB bus, SX2 handles it like any other enumeration sequence. This means that SX2 will enumerate again and assert the ENUMOK interrupt to let the external master know that it has enumerated. The external master will then be responsible for configuring the SX2 for the application. The external master should also check whether SX2 enumerated at High or Full speed in order to adjust the EPxPKTLENH/L register values accordingly. The last initialization task is for the external master to flush all of the SX2 FIFOs.

3.5.3 Wakeup

The SX2 exits its low-power state when one of the following events occur:

- USB bus signals a resume. The SX2 will assert a BUSACTIVITY interrupt.
- The external master asserts the WAKEUP pin. The SX2 will assert a READY interrupt.

3.6 Endpoint RAM**3.6.1 Size**

- Control endpoint: 64 Bytes: 1 × 64 bytes (Endpoint 0).
- FIFO Endpoints: 4096 Bytes: 8 × 512 bytes (Endpoint 2, 4, 6, 8).

3.6.2 Organization

- EP0—Bidirectional Endpoint 0, 64-byte buffer.
- EP2, 4, 6, 8—Eight 512-byte buffers, bulk, interrupt, or isochronous. EP2 and EP6 can be either double-, triple-, or quad-buffered. EP4 and EP8 can only be double-buffered. For high-speed endpoint configuration options, see *Figure 3-1*.

3.6.3 Endpoint Configurations (High-speed Mode)

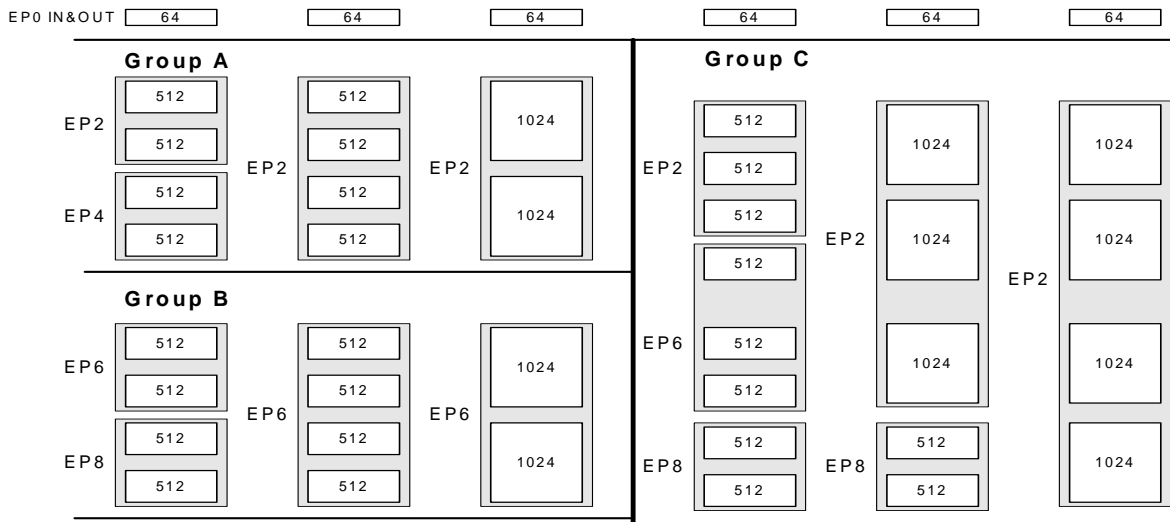


Figure 3-1. Endpoint Configuration

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

- One configuration from Group A and one from Group B
- One configuration from Group C.

Some example endpoint configurations are as follows.

- 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.

3.6.4 Default Endpoint Memory Configuration

At power-on-reset, the endpoint memories are configured as follows:

- EP2: Bulk OUT, 512 bytes/packet, 2x buffered.
- EP4: Bulk OUT, 512 bytes/packet, 2x buffered.
- EP6: Bulk IN, 512 bytes/packet, 2x buffered.
- EP8: Bulk IN, 512 bytes/packet, 2x buffered.

3.7 External Interface

The SX2 presents two interfaces to the external master.

1. A FIFO interface through which EP2, 4, 6, and 8 data flows.
2. A command interface, which is used to set up the SX2, read status, load descriptors, and access Endpoint 0.

3.7.1 Architecture

The SX2 slave FIFO architecture has eight 512-byte blocks in the endpoint RAM that directly serve as FIFO memories and are controlled by FIFO control signals (IFCLK, CS#, SLRD, SLWR, SLOE, PKTEND, and FIFOADR[2:0]).

The SX2 command interface is used to set up the SX2, read status, load descriptors, and access Endpoint 0. The command interface has its own READY signal for gating writes, and an INT# signal to indicate that the SX2 has data to be read, or that an interrupt event has occurred. The command interface uses the same control signals (IFCLK, CS#, SLRD, SLWR, SLOE, and FIFOADR[2:0]) as the FIFO interface, except for PKTEND.

3.7.2 Control Signals

3.7.2.1 FIFOADDR Lines

The SX2 has three address pins that are used to select either the FIFOs or the command interface. The addresses correspond to the following table.

Table 3-3. FIFO Address Lines Setting

Address/Selection	FIFOADR2	FIFOADR1	FIFOADR0
FIFO2	0	0	0
FIFO4	0	0	1
FIFO6	0	1	0
FIFO8	0	1	1
COMMAND	1	0	0
RESERVED	1	0	1
RESERVED	1	1	0
RESERVED	1	1	1

The SX2 accepts either an internally derived clock (30 or 48 MHz) or externally supplied clock (IFCLK, 5-50 MHz), and SLRD, SLWR, SLOE, PKTEND, CS#, FIFOADR[2:0] signals from an external master. The interface can be selected for 8- or 16-bit operation by an internal configuration bit, and an Output Enable signal SLOE enables the data bus driver of the selected width. The external master must ensure that the output enable signal is inactive when writing data to the SX2. The interface can operate either asynchronously where the SLRD and SLWR signals act directly as strobes, or synchronously where the SLRD and SLWR act as clock qualifiers. The optional CS# signal will tristate the data bus and ignore SLRD, SLWR, PKTEND.

The external master reads from OUT endpoints and writes to IN endpoints, and reads from or writes to the command interface.

3.7.2.2 Read: SLOE and SLRD

In synchronous mode, the FIFO pointer is incremented on each rising edge of IFCLK while SLRD is asserted. In asynchronous mode, the FIFO pointer is incremented on each asserted-to-deasserted transition of SLRD.

SLOE is a data bus driver enable. When SLOE is asserted, the data bus is driven by the SX2.

3.7.2.3 Write: SLWR

In synchronous mode, data on the FD bus is written to the FIFO (and the FIFO pointer is incremented) on each rising edge of IFCLK while SLWR is asserted. In asynchronous mode, data on the FD bus is written to the FIFO (and the FIFO pointer is incremented) on each asserted-to-deasserted transition of SLWR.

3.7.2.4 PKTEND

PKTEND commits the current buffer to USB. To send a short IN packet (one which has not been filled to max packet size determined by the value of PL[X:0] in EPxPKTLENH/L), the external master strobes the PKTEND pin.

3.7.3 IFCLK

The IFCLK pin can be configured to be either an input (default) or an output interface clock. Bits IFCONFIG[7:4] define the behavior of the interface clock. To use the SX2's internally-derived 30- or 48-MHz clock, set IFCONFIG.7 to 1 and set IFCONFIG.6 to 0 (30 MHz) or to 1 (48 MHz). To use an externally supplied clock, set IFCONFIG.7=0 and drive the IFCLK pin (5 MHz – 50 MHz). The input or output IFCLK signal can be inverted by setting IFCONFIG.4=1.

3.7.4 FIFO Access

An external master can access the slave FIFOs either asynchronously or synchronously:

- Asynchronous—SLRD, SLWR, and PKTEND pins are strobes.
- Synchronous—SLRD, SLWR, and PKTEND pins are enables for the IFCLK clock pin.

An external master accesses the FIFOs through the data bus, FD [15:0]. This bus can be either 8- or 16-bits wide; the width is selected via the WORDWIDE bit in the EPxPKTLENH/L registers. The data bus is bidirectional, with its output drivers controlled by the SLOE pin. The FIFOADR[2:0] pins select which of the four FIFOs is connected to the FD [15:0] bus, or if the command interface is selected.

3.7.5 FIFO Flag Pins Configuration

The FIFO flags are FLAGA, FLAGB, FLAGC, and FLAGD. These FLAGx pins report the status of the FIFO selected by the FIFOADR[2:0] pins. At reset, these pins are configured to report the status of the following:

- FLAGA reports the status of the programmable flag.
- FLAGB reports the status of the full flag.
- FLAGC reports the status of the empty flag.
- FLAGD defaults to the CS# function.

The FIFO flags can either be indexed or fixed. Fixed flags report the status of a particular FIFO regardless of the value on the FIFOADR [2:0] pins. Indexed flags report the status of the FIFO selected by the FIFOADR [2:0] pins.^[3]

3.7.6 Default FIFO Programmable Flag Set-up

By default, FLAGA is the Programmable Flag (PF) for the endpoint being pointed to by the FIFOADR[2:0] pins. For EP2 and EP4, the default endpoint configuration is BULK, OUT, 512, 2x, and the PF pin asserts when the entire FIFO has greater than/equal to 512 bytes. For EP6 and EP8, the default endpoint configuration is BULK, IN, 512, 2x, and the PF pin asserts when the entire FIFO has less than/equal to 512 bytes. In other words, EP6/8 report a half-empty state, and EP2/4 report a half-full state. The polarity of the programmable flag is set to active low and cannot be altered.

3.7.7 FIFO Programmable Flag (PF) Set-up

Each FIFO's programmable-level flag (PF) asserts when the FIFO reaches a user-defined fullness threshold. That threshold is configured as follows:

1. For OUT packets: The threshold is stored in PFC12:0. The PF is asserted when the number of bytes *in the entire FIFO* is less than/equal to (DECIS = 0) or greater than/equal to (DECIS = 1) the threshold.
2. For IN packets, with PKTSTAT = 1: The threshold is stored in PFC9:0. The PF is asserted when the number of bytes written into *the current packet in the FIFO* is less than/equal to (DECIS = 0) or greater than/equal to (DECIS = 1) the threshold.
3. For IN packets, with PKTSTAT = 0: The threshold is stored in two parts: PKTS2:0 holds the number of committed packets, and PFC9:0 holds the number of bytes in the current packet. The PF is asserted when the FIFO is at or less full than (DECIS = 0), or at or more full than (DECIS = 1), the threshold.

3.7.8 Command Protocol

An address of [1 0 0] on FIFOADR [2:0] will select the command interface. The command interface is used to write to and read from the SX2 registers and the Endpoint 0 buffer, as well as the descriptor RAM. Command read and write transactions occur over FD[7:0] only. Each byte written to the SX2 is either an address or a data byte, as determined by bit7. If bit7 = 1, then the byte is considered an address byte. If bit7 = 0, then the byte is considered a data byte. If bit7 = 1, then bit6 determines whether the address byte is a read request or a write request. If bit6 = 1, then the byte is considered a read request. If bit6 = 0 then the byte is considered a write request. Bits [5:0] hold the register address of the request. The format of the command address byte is shown in *Table 3-4*.

Table 3-4. Command Address Byte

Address/Data#	Read/Write#	A5	A4	A3	A2	A1	A0
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Each Write request is followed by two or more data bytes. If another address byte is received before both data bytes are received, the SX2 ignores the first address and any incomplete data transfers. The format for the data bytes is shown in *Table 3-5* and *Table 3-6*. Some registers take a series of bytes. Each byte is transferred using the same protocol.

Table 3-5. Command Data Byte One

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	X	X	X	D7	D6	D5	D4

Table 3-6. Command Data Byte Two

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	X	X	X	D3	D2	D1	D0

The first command data byte contains the upper nibble of data, and the second command byte contains the lower nibble of data.

Note:

3. In indexed mode, the value of the FLAGx pins is indeterminate except when addressing a FIFO (FIFOADR[2:0]={000,001,010,011}).

3.7.8.1 Write Request Example

Prior to writing to a register, two conditions must be met: FIFOADR[2:0] must hold [1 0 0], and the Ready line must be HIGH. The external master should not initiate a command if the READY pin is not in a High state.

Example: to write the byte <10110000> into the IFCONFIG register (0x01), first send a command address byte as follows.

Table 3-7. Command Address Write Byte

Address/Data#	Read/Write#	A5	A4	A3	A2	A1	A0
1	0	0	0	0	0	0	1

- The first bit signifies an address transfer.
- The second bit signifies that this is a write command.
- The next six bits represent the register address (000001 binary = 0x01 hex).

Once the byte has been received the SX2 pulls the READY pin low to inform the external master not to send any more information. When the SX2 is ready to receive the next byte, the SX2 pulls the READY pin high again. Next, the upper nibble of the data byte is written to the SX2 as follows.

Table 3-8. Command Data Write Byte One

Address/Data#	Don't Care	Don't Care	Don't Care	D7	D6	D5	D4
0	X	X	X	1	0	1	1

- The first bit signifies that this is a data transfer.
- The next three are don't care bits.
- The next four bits hold the upper nibble of the transferred byte.

Next, the lower nibble of the data byte is written to the SX2:

Table 3-9. Command Data Write Byte Two

Address/Data#	Don't Care	Don't Care	Don't Care	D3	D2	D1	D0
0	X	X	X	0	0	0	0

At this point the entire byte <10110000> has been transferred to register 0x01 and the write sequence is complete.

3.7.8.2 Read Request Example

The Read cycle is simpler than the write cycle. The Read cycle consists of a read request from the external master to the SX2. For example, to read the contents of register 0x01, a command address byte is written to the SX2 as follows.

Table 3-10. Command Address Read Byte

Address/Data#	Read/Write#	A5	A4	A3	A2	A1	A0
1	1	0	0	0	0	0	1

When the data is ready to be read, the SX2 asserts the INT# pin to tell the external master that the data it requested is waiting on FD[7:0].^[4]

4.0 Enumeration

The SX2 has two modes of enumeration. The first mode is automatic through EEPROM boot load, as described in section 3.3. The second method is a manual load of the descriptor or VID, PID, and DID as described below.

4.1 Standard Enumeration

The SX2 has 500 bytes of descriptor RAM into which the external master may write its descriptor. The descriptor RAM is accessed through register 0x30. To load a descriptor, the external master does the following:

- Initiate a Write Request to register 0x30.
- Write two bytes (four command data transfers) that define the length of the entire descriptor about to be transferred. The LSB is written first, followed by the MSB.^[5]
- Write the descriptor, one byte at a time until complete.^[5] Note: the register address is only written once.

Notes:

4. An important note: Once the SX2 receives a Read request, the SX2 allocates the interrupt line solely for the read request. If one of the six interrupt sources described in Section 3.4 is asserted, the SX2 will buffer that interrupt until the read request completes.
5. These and all other data bytes must conform to the command protocol.

After the entire descriptor has been transferred, the SX2 will float the pull-up resistor connected to D+, and parse through the descriptor to locate the individual descriptors. After the SX2 has parsed the entire descriptor, the SX2 will connect the pull-up resistor and enumerate automatically. When enumeration is complete, the SX2 will notify the external master with an ENUMOK interrupt.

The format and order of the descriptor should be as follows (see Section 12.0 for an example):

- Device.
- Device qualifier.
- High-speed configuration, high-speed interface, high-speed endpoints.
- Full-speed configuration, full-speed interface, full-speed endpoints.
- Strings.

4.2 Default Enumeration

The external master may simply load a VID, PID, and DID and use the default descriptor built into the SX2. To use the default descriptor, the descriptor length described above must equal 6. After the external master has written the length, the VID, PID, and DID must be written LSB, then MSB. For example, if the VID, PID, and DID are 0x04B4, 0x1002, and 0x0001 respectively, then the external master does the following:

- Initiates a Write Request to register 0x30.
- Writes two bytes (four command data transfers) that define the length of the entire descriptor about to be transferred. In this case, the length is always six.
- Writes the VID, PID, and DID bytes: 0xB4, 0x04, 0x02, 0x10, 0x01, 0x00 (in nibble format per the command protocol).

The default descriptor is listed in Section 12.0. The default descriptor can be used as a starting point for a custom descriptor.

5.0 Endpoint 0

The SX2 will automatically respond to USB chapter 9 requests without any external master intervention. If the SX2 receives a request to which it cannot respond automatically, the SX2 will notify the external master. The external master then has the choice of responding to the request or stalling.

After the SX2 receives a set-up packet to which it cannot respond automatically, the SX2 will assert a SETUP interrupt. After the external master reads the Interrupt Status Byte to determine that the interrupt source was the SETUP interrupt, it can initiate a read request to the SETUP register, 0x32. When the SX2 sees a read request for the SETUP register, it will present the first byte of set-up data to the external master. Each additional read request will present the next byte of set-up data, until all eight bytes have been read.

The external master can stall this request at this or any other time. To stall a request, the external master initiates a write request for the SETUP register, 0x32, and writes any non-zero value to the register.

If this set-up request has a data phase, the SX2 will then interrupt the external master with an EP0BUF interrupt when the buffer becomes available. The SX2 determines the direction of the set-up request and interrupts when either:

- IN: the Endpoint 0 buffer becomes available to write to, or
- OUT: the Endpoint 0 buffer receives a packet from the USB host.

For an IN set-up transaction, the external master can write up to 64 bytes at a time for the data phase. The steps to write a packet are as follows:

1. Wait for an EP0BUF interrupt, indicating that the buffer is available.
2. Initiate a write request for register 0x31.
3. Write one data byte.
4. Repeat steps 2 and 3 until either all the data or 64 bytes have been written, whichever is less.
5. Write the number of bytes in this packet to the byte count register, 0x33.

To send more than 64 bytes, the process is repeated. The SX2 internally stores the length of the data phase that was specified in the wLength field (bytes 6,7) of the set-up packet. To send less than the requested amount of data, the external master writes a packet that is less than 64 bytes, or if a multiple of 64, the external master follows the data with a zero-length packet. When the SX2 sees a short or zero-length packet, it will complete the set-up transfer by automatically completing the handshake phase. The SX2 will not allow more data than the wLength field specified in the set-up packet. Note: the PKTEND pin does not apply to Endpoint 0. The only way to send a short or zero length packet is by writing to the byte count register with the appropriate value.

For an OUT set-up transaction, the external master can read each packet received from the USB host during the data phase. The steps to read a packet are as follows:

1. Wait for an EPOBUF interrupt, indicating that a packet was received from the USB host into the buffer.
2. Initiate a read request for the byte count register, 0x33. This indicates the amount of data received from the host.
3. Initiate a read request for register 0x31.
4. Read one byte.
5. Repeat steps 3 and 4 until the number of bytes specified in the byte count register has been read.

To receive more than 64 bytes, the process is repeated. The SX2 internally stores the length of the data phase that was specified in the wLength field of the set-up packet (bytes 6,7). When the SX2 sees that the specified number of bytes have been received, it will complete the set-up transfer by automatically completing the handshake phase. If the external master does not wish to receive the entire transfer, it can stall the transfer.

If the SX2 receives another set-up packet before the current transfer has completed, it will interrupt the external master with another SETUP interrupt.

If the SX2 receives a set-up packet with no data phase, the external master can accept the packet and complete the handshake phase by writing zero to the byte count register.

6.0 Pin Assignments

6.1 56-pin SSOP

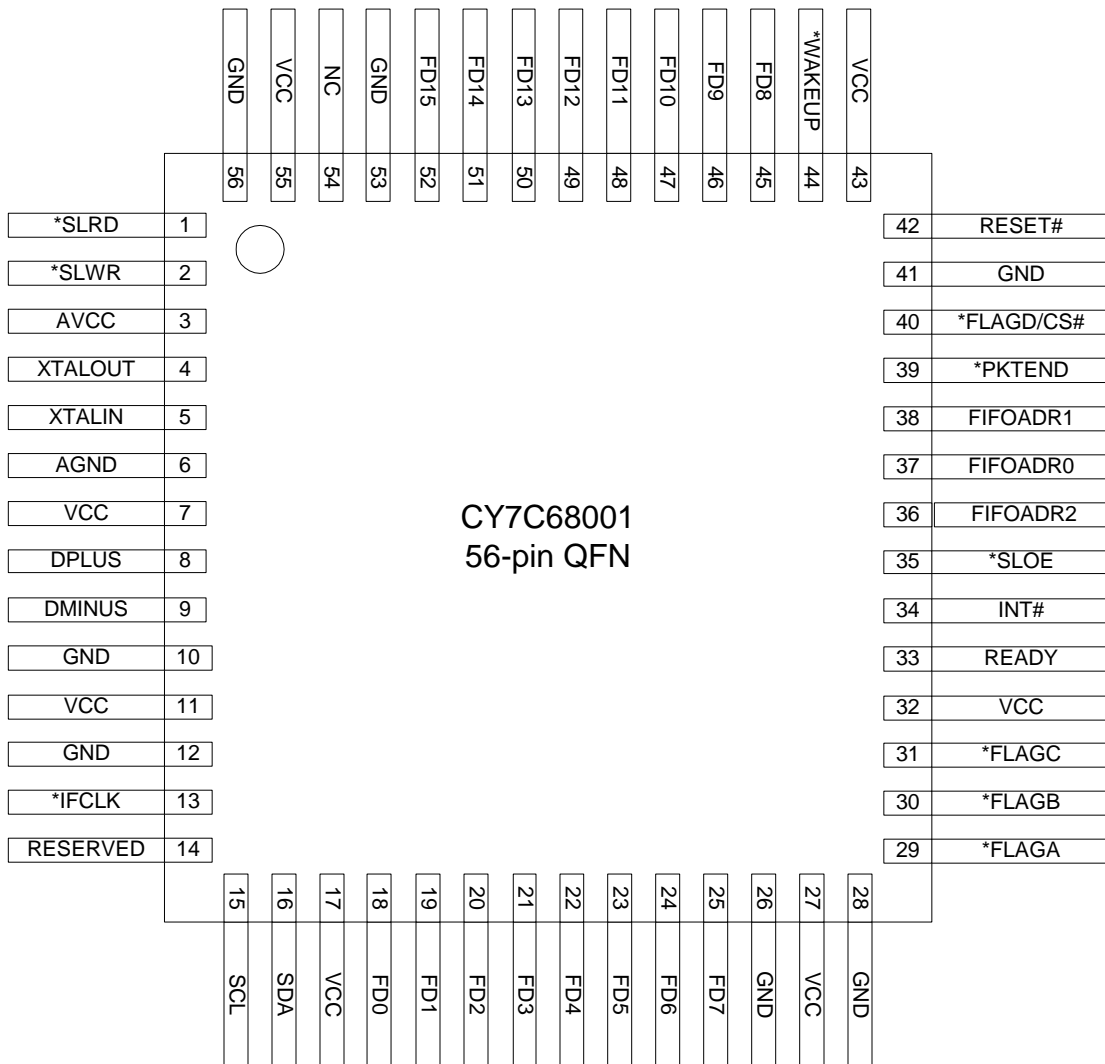
1	FD13	FD12	56
2	FD14	FD11	55
3	FD15	FD10	54
4	GND	FD9	53
5	NC	FD8	52
6	VCC	*WAKEUP	51
7	GND	VCC	50
8	*SLRD	RESET#	49
9	*SLWR	GND	48
10	AVCC	*FLAGD/CS#	47
11	XTALOUT	*PKTEND	46
12	XTALIN	FIFOADR1	45
13	AGND	FIFOADR0	44
14	VCC	FIFOADR2	43
15	DPLUS	*SLOE	42
16	DMINUS	INT#	41
17	GND	READY	40
18	VCC	VCC	39
19	GND	*FLAGC	38
20	*IFCLK	*FLAGB	37
21	RESERVED	*FLAGA	36
22	SCL	GND	35
23	SDA	VCC	34
24	VCC	GND	33
25	FD0	FD7	32
26	FD1	FD6	31
27	FD2	FD5	30
28	FD3	FD4	29

CY7C68001
56-pin SSOP

Figure 6-1. CY7C68001 56-pin SSOP Pin Assignment^[6]

Note:

6. A * denotes programmable polarity.

6.2 56-pin QFN

Figure 6-2. CY7C68001 56-pin QFN Assignment^[6]

6.3 CY7C68001 Pin Descriptions
Table 6-1. SX2 Pin Descriptions

QFN Pin	SSOP Pin	Name	Type	Default	Description
3	10	AVCC	Power	N/A	Analog V_{CC} . This signal provides power to the analog section of the chip.
6	13	AGND	Power	N/A	Analog Ground . Connect to ground with as short a path as possible.
9	16	DMINUS	I/O/Z	Z	USB D– Signal . Connect to the USB D– signal.
8	15	DPLUS	I/O/Z	Z	USB D+ Signal . Connect to the USB D+ signal.
42	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.
5	12	XTALIN	Input	N/A	Crystal Input . Connect this signal to a 24-MHz parallel-resonant, fundamental mode crystal and 20-pF capacitor to GND. It is also correct to drive XTALIN with an external 24-MHz square wave derived from another clock source.
4	11	XTALOUT	Output	N/A	Crystal Output . Connect this signal to a 24-MHz parallel-resonant, fundamental mode crystal and 20-pF capacitor to GND. If an external clock is used to drive XTALIN, leave this pin open.
54	5	NC	Output	O	No Connect . This pin must be left unconnected.
33	40	READY	Output	L	READY is an output-only ready that gates external command reads and writes. Active High.
34	41	INT#	Output	H	INT# is an output-only external interrupt signal. Active Low.
35	42	SLOE	Input	I	SLOE is an input-only output enable with programmable polarity (POLAR.4) for the slave FIFOs connected to FD[7:0] or FD[15:0].
36	43	FIFOADR2	Input	I	FIFOADR2 is an input-only address select for the slave FIFOs connected to FD[7:0] or FD[15:0].
37	44	FIFOADR0	Input	I	FIFOADR0 is an input-only address select for the slave FIFOs connected to FD[7:0] or FD[15:0].
38	45	FIFOADR1	Input	I	FIFOADR1 is an input-only address select for the slave FIFOs connected to FD[7:0] or FD[15:0].
39	46	PKTEND	Input	I	PKTEND is an input-only packet end with programmable polarity (POLAR.5) for the slave FIFOs connected to FD[7:0] or FD[15:0].
40	47	FLAGD/C S#	CS#:I FLAGD:O	I	FLAGD is a programmable slave-FIFO output status flag signal. CS# is a master chip select (default).
18	25	FD[0]	I/O/Z	I	FD[0] is the bidirectional FIFO/Command data bus.
19	26	FD[1]	I/O/Z	I	FD[1] is the bidirectional FIFO/Command data bus.
20	27	FD[2]	I/O/Z	I	FD[2] is the bidirectional FIFO/Command data bus.
21	28	FD[3]	I/O/Z	I	FD[3] is the bidirectional FIFO/Command data bus.
22	29	FD[4]	I/O/Z	I	FD[4] is the bidirectional FIFO/Command data bus.
23	30	FD[5]	I/O/Z	I	FD[5] is the bidirectional FIFO/Command data bus.
24	31	FD[6]	I/O/Z	I	FD[6] is the bidirectional FIFO/Command data bus.
25	32	FD[7]	I/O/Z	I	FD[7] is the bidirectional FIFO/Command data bus.
45	52	FD[8]	I/O/Z	I	FD[8] is the bidirectional FIFO data bus.
46	53	FD[9]	I/O/Z	I	FD[9] is the bidirectional FIFO data bus.
47	54	FD[10]	I/O/Z	I	FD[10] is the bidirectional FIFO data bus.
48	55	FD[11]	I/O/Z	I	FD[11] is the bidirectional FIFO data bus.
49	56	FD[12]	I/O/Z	I	FD[12] is the bidirectional FIFO data bus.
50	1	FD[13]	I/O/Z	I	FD[13] is the bidirectional FIFO data bus.
51	2	FD[14]	I/O/Z	I	FD[14] is the bidirectional FIFO data bus.
52	3	FD[15]	I/O/Z	I	FD[15] is the bidirectional FIFO data bus.

Table 6-1. SX2 Pin Descriptions (continued)

QFN Pin	SSOP Pin	Name	Type	Default	Description
1	8	SLRD	Input	N/A	SLRD is the input-only read strobe with programmable polarity (POLAR.3) for the slave FIFOs connected to FD[7:0] or FD[15:0].
2	9	SLWR	Input	N/A	SLWR is the input-only write strobe with programmable polarity (POLAR.2) for the slave FIFOs connected to FD[7:0] or FD[15:0].
29	36	FLAGA	Output	H	FLAGA is a programmable slave-FIFO output status flag signal. Defaults to PF for the FIFO selected by the FIFOADR[2:0] pins.
30	37	FLAGB	Output	H	FLAGB is a programmable slave-FIFO output status flag signal. Defaults to FULL for the FIFO selected by the FIFOADR[2:0] pins.
31	38	FLAGC	Output	H	FLAGC is a programmable slave-FIFO output status flag signal. Defaults to EMPTY for the FIFO selected by the FIFOADR[2:0] pins.
13	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. When using the internal clock reference (IFCONFIG.7=1) the IFCLK pin can be configured to output 30/48 MHz by setting bits IFCONFIG.5 and IFCONFIG.6. IFCLK may be inverted by setting the bit IFCONFIG.4=1. Programmable polarity.
14	21	Reserved	Input	N/A	Reserved. Must be connected to ground.
44	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. During normal operation, holding WAKEUP asserted inhibits the SX2 chip from suspending. This pin has programmable polarity (POLAR.7).
15	22	SCL	OD	Z	I²C Clock. Connect to V _{CC} with a 2.2K-10 K Ohms resistor, even if no I ² C EEPROM is attached.
16	23	SDA	OD	Z	I²C Data. Connect to V _{CC} with a 2.2K-10 K Ohms resistor, even if no I ² C EEPROM is attached.
55	6	V _{CC}	Power	N/A	V_{CC}. Connect to 3.3V power source.
7	14	V _{CC}	Power	N/A	V_{CC}. Connect to 3.3V power source.
11	18	V _{CC}	Power	N/A	V_{CC}. Connect to 3.3V power source.
17	24	V _{CC}	Power	N/A	V_{CC}. Connect to 3.3V power source.
27	34	V _{CC}	Power	N/A	V_{CC}. Connect to 3.3V power source.
32	39	V _{CC}	Power	N/A	V_{CC}. Connect to 3.3V power source.
43	50	V _{CC}	Power	N/A	V_{CC}. Connect to 3.3V power source.
53	4	GND	Ground	N/A	Connect to ground.
56	7	GND	Ground	N/A	Connect to ground.
10	17	GND	Ground	N/A	Connect to ground.
12	19	GND	Ground	N/A	Connect to ground.
26	33	GND	Ground	N/A	Connect to ground.
28	35	GND	Ground	N/A	Connect to ground.
41	48	GND	Ground	N/A	Connect to 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	STANDBY	FLAGD/CS#	DISCON	11001001	RW
02	1	FLAGSAB	FIFO FLAGA and FLAGB Assignments	FLAGB3	FLAGB2	FLAGB1	FLAGB0	FLAGA3	FLAGA2	FLAGA1	FLAGA0	00000000	RW
03	1	FLAGSCD	FIFO FLAGC and FLAGD Assignments	FLAGD3	FLAGD2	FLAGD1	FLAGD0	FLAGC3	FLAGC2	FLAGC1	FLAGC0	00000000	RW
04	1	POLAR	FIFO polarities	WUPOL	0	PKTEND	SLOE	SLRD	SLWR	EF	FF	00000000	RW
05	1	REVID	Chip Revision	Major	Major	Major	Major	minor	minor	minor	minor	xxxxxxx	R
*Endpoint Configuration													
06	1	EP2CFG	Endpoint 2 Configuration	VALID	dir	TYPE1	TYPE0	SIZE	STALL	BUF1	BUF0	10100010	RW
07	1	EP4CFG	Endpoint 4 Configuration	VALID	dir	TYPE1	TYPE0	0	STALL	0	0	10100000	RW
08	1	EP6CFG	Endpoint 6 Configuration	VALID	dir	TYPE1	TYPE0	SIZE	STALL	BUF1	BUF0	11000010	RW
09	1	EP8CFG	Endpoint 8 Configuration	VALID	dir	TYPE1	TYPE0	0	STALL	0	0	11000000	RW
0A	1	EP2PKTLENH	Endpoint 2 Packet Length H	INFM1	OEP1	ZEROLEN	WORD-WIDE	0	PL10	PL9	PL8	00110010	RW
0B	1	EP2PKTLENL	Endpoint 2 Packet Length L (IN only)	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
0C	1	EP4PKTLENH	Endpoint 4 Packet Length H	INFM1	OEP1	ZEROLEN	WORD-WIDE	0	0	PL9	PL8	00110010	RW
0D	1	EP4PKTLENL	Endpoint 4 Packet Length L (IN only)	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
0E	1	EP6PKTLENH	Endpoint 6 Packet Length H	INFM1	OEP1	ZEROLEN	WORD-WIDE	0	PL10	PL9	PL8	00110010	RW
0F	1	EP6PKTLENL	Endpoint 6 Packet Length L (IN only)	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
10	1	EP8PKTLENH	Endpoint 8 Packet Length H	INFM1	OEP1	ZEROLEN	WORD-WIDE	0	0	PL9	PL8	00110010	RW
11	1	EP8PKTLENL	Endpoint 8 Packet Length L (IN only)	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
12	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	10001000	RW
13	1	EP2PFL	EP2 Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
14	1	EP4PFH	EP4 Programmable Flag H	DECIS	PKTSTAT	0	IN:PKTS[1] OUT:PFC11	IN:PKTS[0] OUT:PFC9	0	0	PFC8	10001000	RW
15	1	EP4PFL	EP4 Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
16	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	00001000	RW
17	1	EP6PFL	EP6 Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
18	1	EP8PFH	EP8 Programmable Flag H	DECIS	PKTSTAT	0	IN:PKTS[1] OUT:PFC11	IN:PKTS[0] OUT:PFC9	0	0	PFC8	00001000	RW
19	1	EP8PFL	EP8 Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
1A	1	EP2ISOINPKTS	EP2 (if ISO) IN Packets per frame (1-3)	0	0	0	0	0	0	INPPF1	INPPF0	00000001	RW
1B	1	EP4ISOINPKTS	EP4 (if ISO) IN Packets per frame (1-3)	0	0	0	0	0	0	INPPF1	INPPF0	00000001	RW
1C	1	EP6ISOINPKTS	EP6 (if ISO) IN Packets per frame (1-3)	0	0	0	0	0	0	INPPF1	INPPF0	00000001	RW
1D	1	EP8ISOINPKTS	EP8 (if ISO) IN Packets per frame (1-3)	0	0	0	0	0	0	INPPF1	INPPF0	00000001	RW
FLAGS													
1E	1	EP24FLAGS	Endpoints 2,4 FIFO Flags	0	EP4PF	EP4EF	EP4FF	0	EP2PF	EP2EF	EP2FF	00100010	R
1F	1	EP68FLAGS	Endpoints 6,8 FIFO Flags	0	EP8PF	EP8EF	EP8FF	0	EP6PF	EP6EF	EP6FF	01100110	R
INPKTEND/FLUSH													
20	1	INPKTEND/FLUSH	Force Packet End / Flush FIFOs	FIFO8	FIFO6	FIFO4	FIFO2	EP3	EP2	EP1	EP0	00000000	W
USB Configuration													
2A	1	USBFRAMEH	USB Frame count H	0	0	0	0	0	FC10	FC9	FC8	xxxxxxx	R
2B	1	USBFRAMEL	USB Frame count L	FC7	FC6	FC5	FC4	FC3	FC2	FC1	FC0	xxxxxxx	R
2C	1	MICROFRAME	Microframe count, 0-7	0	0	0	0	0	MF2	MF1	MF0	xxxxxxx	R
2D	1	FNADDR	USB Function address	HSGRANT	FA6	FA5	FA4	FA3	FA2	FA1	FA0	xxxxxxx	R
Interrupts													
2E	1	INTENABLE	Interrupt Enable	SETUP	EP0BUF	FLAGS	1	1	ENUMOK	BUSACTIVITY	READY	11111111	RW
Descriptor													
30	500	DESC	Descriptor RAM	d7	d6	d5	d4	d3	d2	d1	d0	xxxxxxx	W
Endpoint 0													
31	64	EP0BUF	Endpoint 0 Buffer	d7	d6	d5	d4	d3	d2	d1	d0	xxxxxxx	RW
32	8/1	SETUP	Endpoint 0 Set-up Data / Stall	d7	d6	d5	d4	d3	d2	d1	d0	xxxxxxx	RW
33	1	EP0BC	Endpoint 0 Byte Count	d7	d6	d5	d4	d3	d2	d1	d0	xxxxxxx	RW

*Please note that the SX2 was not designed to support dynamic modification of these endpoint configuration registers. If your applications need the ability to change endpoint configurations after the device has already enumerate with a specific configuration, please expect some delay in being able to access the FIFOs after changing the configuration. For example, after writing to EP2PKTLENL, you must wait for at least 35 us measured from the time the READY signal is asserted before writing to the FIFO. This delay time varies for different registers and is not characterized, because the SX2 was not designed for this dynamic change of endpoint configuration registers.

7.1 IFCONFIG Register 0x01

IFCONFIG								0x01
Bit #	7	6	5	4	3	2	1	0
Bit Name	IFCLKSRC	3048MHZ	IFCLKOE	IFCLKPOL	ASYNC	STANDBY	FLAGD/CS#	DISCON
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Default	1	1	0	0	1	0	0	1

7.1.1 Bit 7: 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 (default), an internal 30 or 48 MHz clock is used.

7.1.2 Bit 6: 3048MHZ

This bit selects the internal FIFO clock frequency. If 3048MHZ = 0, the internal clock frequency is 30 MHz. If 3048MHZ = 1 (default), the internal clock frequency is 48 MHz.

7.1.3 Bit 5: IFCLKOE

This bit selects if the IFCLK pin is driven. If IFCLKOE = 0 (default), the IFCLK pin is floated. If IFCLKOE = 1, the IFCLK pin is driven.

7.1.4 Bit 4: IFCLKPOL

This bit controls the polarity of the IFCLK signal.

- When IFCLKPOL=0, the clock has the polarity shown in all the timing diagrams in this data sheet (rising edge is the activating edge).
- When IFCLKPOL=1, the clock is inverted (in some cases may help with satisfying data set-up times).

7.1.5 Bit 3: 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, and the FIFO control signals function as read and write enable signals for the clock signal.

When ASYNC = 1 (default), 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 Bit 2: STANDBY

This bit instructs the SX2 to enter a low-power mode. When STANDBY=1, the SX2 will enter a low-power mode by turning off its oscillator. The external master should write this bit after it receives a bus activity interrupt (indicating that the host has signaled a USB suspend condition). If SX2 is disconnected from the USB bus, the external master can write this bit at any time to save power. Once suspended, the SX2 is awakened either by resumption of USB bus activity or by assertion of its WAKEUP pin.

7.1.7 Bit 1: FLAGD/CS#

This bit controls the function of the FLAGD/CS# pin. When FLAGD/CS# = 0 (default), the pin operates as a slave chip select. If FLAGD/CS# = 1, the pin operates as FLAGD.

7.1.8 Bit 0: DISCON

This bit controls whether the internal pull-up resistor connected to D+ is pulled high or floating. When DISCON = 1 (default), the pull-up resistor is floating simulating a USB unplug. When DISCON=0, the pull-up resistor is pulled high signaling a USB connection.

7.2 FLAGSAB/FLAGSCD Registers 0x02/0x03

The SX2 has four FIFO flags output pins: FLAGA, FLAGB, FLAGC, FLAGD.

FLAGSAB								0x02
Bit #	7	6	5	4	3	2	1	0
Bit Name	FLAGB3	FLAGB2	FLAGB1	FLAGB0	FLAGA3	FLAGA2	FLAGA1	FLAGA0
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

FLAGSCD

0x03

Bit #	7	6	5	4	3	2	1	0
Bit Name	FLAGD3	FLAGD2	FLAGD1	FLAGD0	FLAGC3	FLAGC2	FLAGC1	FLAGC0
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

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.

Table 7-2. FIFO Flag 4-bit Coding

FLAGx3	FLAGx2	FLAGx1	FLAGx0	Pin Function
0	0	0	0	FLAGA = PF, FLAGB = FF, FLAGC = EF, FLAGD = CS# (actual FIFO is selected by FIFOADR[2:0] 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
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 FIFOADR[2:0] select to which of the four FIFOs the flags correspond. These pins are decoded as shown in *Table 3-3*.

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 FIFOADR [2:0] pins. For example, the external master could be filling the EP2IN FIFO with data while also checking the empty flag for the EP4OUT FIFO.

7.3 POLAR Register 0x04

This register controls the polarities of FIFO pin signals and the WAKEUP pin.

POLAR

0x04

Bit #	7	6	5	4	3	2	1	0
Bit Name	WUPOL	0	PKTEND	SLOE	SLRD	SLWR	EF	FF
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

7.3.1 Bit 7: WUPOL

This flag sets the polarity of the WAKEUP pin. If WUPOL = 0 (default), the polarity is active LOW. If WUPOL=1, the polarity is active HIGH.

7.3.2 Bit 5: PKTEND

This flag selects the polarity of the PKTEND pin. If PKTEND = 0 (default), the polarity is active LOW. If PKTEND = 1, the polarity is active HIGH.

7.3.3 Bit 4: SLOE

This flag selects the polarity of the SLOE pin. If SLOE = 0 (default), the polarity is active LOW. If SLOE = 1, the polarity is active HIGH. This bit can only be changed by using the EEPROM configuration load.

7.3.4 Bit 3: SLRD

This flag selects the polarity of the SLRD pin. If SLRD = 0 (default), the polarity is active LOW. If SLRD = 1, the polarity is active HIGH. This bit can only be changed by using the EEPROM configuration load.

7.3.5 SLWR Bit 2

This flag selects the polarity of the SLWR pin. If SLWR = 0 (default), the polarity is active LOW. If SLWR = 1, the polarity is active HIGH. This bit can only be changed by using the EEPROM configuration load.

7.3.6 EF Bit 1

This flag selects the polarity of the EF pin (FLAGA/B/C/D). If EF = 0 (default), the EF pin is pulled low when the FIFO is empty. If EF = 1, the EF pin is pulled HIGH when the FIFO is empty.

7.3.7 FF Bit 0

This flag selects the polarity of the FF pin (FLAGA/B/C/D). If FF = 0 (default), the FF pin is pulled low when the FIFO is full. If FF = 1, the FF pin is pulled HIGH when the FIFO is full.

7.4 REVID Register 0x05

These register bits define the silicon revision.

REVID								0x05
Bit #	7	6	5	4	3	2	1	0
Bit Name	Major	Major	Major	Major	Minor	Minor	Minor	Minor
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

The upper nibble is the major revision. The lower nibble is the minor revision. For example: if REVID = 0x11, then the silicon revision is 1.1.

7.5 EPxCFG Register 0x06–0x09

These registers configure the large, data-handling SX2 endpoints, EP2, 4, 6, and 8. *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.

EPxCFG								0x06, 0x07, 0x08, 0x09
Bit #	7	6	5	4	3	2	1	0
Bit Name	VALID	DIR	TYPE1	TYPE0	SIZE	STALL	BUF1	BUF0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	1	0	0	0	1	0

7.5.1 Bit 7: VALID

The external master sets VALID = 1 to activate an endpoint, and VALID = 0 to deactivate it. All SX2 endpoints default to valid. An endpoint whose VALID bit is 0 does not respond to any USB traffic.

7.5.2 Bit 6: DIR

0 = OUT, 1 = IN. Defaults for EP2/4 are DIR = 0, OUT, and for EP6/8 are DIR = 1, IN.

7.5.3 Bit [5,4]: TYPE1, TYPE0

These bits define the endpoint type, as shown in *Table 7-3*. The TYPE bits apply to all of the endpoint configuration registers. All SX2 endpoints except EP0 default to BULK.

Table 7-3. Endpoint Type

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

7.5.4 Bit 3: SIZE

0 = 512 bytes (default), 1 = 1024 bytes.

Endpoints 4 and 8 can only be 512 bytes. The size of endpoints 2 and 6 is selectable.

7.5.5 Bit 2: STALL

Each bulk endpoint (IN or OUT) has a STALL bit (bit 2). If the external master 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]: BUF1, BUF0

The depth of endpoint buffering is selected via BUF1:0, as shown in *Table 7-4*.

Table 7-4. Endpoint Buffering

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

7.6 EPxPKTLENH/L Registers 0x0A–0x11

The external master 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.

In addition, the EPxPKTLENH register has four other endpoint configuration bits.

EPxPKTLENL 0x0B, 0x0D, 0x0F, 0x11

Bit #	7	6	5	4	3	2	1	0
Bit Name	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
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

EP2PKTLENH, EP6PKTLENH 0x0A, 0x0E

Bit #	7	6	5	4	3	2	1	0
Bit Name	INFM1	OEP1	ZEROLEN	WORDWIDE	0	PL10	PL9	PL8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	1	0

EP4PKTLENH, EP8PKTLENH 0x0C, 0x10

Bit #	7	6	5	4	3	2	1	0
Bit Name	INFM1	OEP1	ZEROLEN	WORDWIDE	0	0	PL9	PL8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	1	0	0	1	0

7.6.1 Bit 7: INFM1 EPxPKTLENH.7

When the external master sets INFM = 1 in an endpoint configuration register, the FIFO flags for that endpoint become valid one sample earlier than when the full condition occurs. These bits take effect only when the FIFOs are operating synchronously according to an internally or externally supplied clock. Having the FIFO flag indications one sample early simplifies some synchronous interfaces. This applies only to IN endpoints. Default is INFM1 = 0.

7.6.2 Bit 6: OEP1 EPxPKTLENH.6

When the external master sets an OEP = 1 in an endpoint configuration register, the FIFO flags for that endpoint become valid one sample earlier than when the empty condition occurs. These bits take effect only when the FIFOs are operating synchronously according to an internally or externally supplied clock. Having the FIFO flag indications one sample early simplifies some synchronous interfaces. This applies only to OUT endpoints. Default is OEP1 = 0.

7.6.3 Bit 5: ZEROLEN EPxPKTLENH.5

When ZEROLEN = 1 (default), a zero length packet will be sent when the PKTEND pin is asserted and there are no bytes in the current packet. If ZEROLEN = 0, then a zero length packet will not be sent under these conditions.

7.6.4 Bit 4: WORDWIDE EPxPKTLENH.4

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

7.6.5 Bit [2..0]: PL[X:0] Packet Length Bits

The default packet size is 512 bytes for all endpoints.

7.7 EPxPFH/L Registers 0x12–0x19

The Programmable Flag registers control when the PF goes active for each of the four endpoint FIFOs: EP2, EP4, EP6, and EP8. The EPxPFH/L fields are interpreted differently for OUT and IN endpoints.

EP2PFL, EP4PFL, EP6PFL, EP8PFL

0x13, 0x15, 0x17, 0x19

Bit #	7	6	5	4	3	2	1	0
Bit Name	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0
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

EP2PFH, EP6PFH

0x12, 0x16

Bit #	7	6	5	4	3	2	1	0
Bit Name	DECIS	PKTSTAT	INPKTS[2] OUT:PFC12	INPKTS[1] OUT:PFC11	INPKTS[0] OUT:PFC10	0	0	PFC8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	1	0	0	0

EP4PFH, EP8PFH

0x14, 0x18

Bit #	7	6	5	4	3	2	1	0
Bit Name	DECIS	PKTSTAT	INPKTS[2] OUT:PFC12	INPKTS[1] OUT:PFC11	INPKTS[0] OUT:PFC10	0	0	PFC8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	0	0	0

7.7.1 DECIS EPxPFH.7

If DECIS = 0, then PF = 1 when the byte count <= PF. If DECIS = 1 (default), then PF = 1 when the byte count >= PF. For OUT endpoints, the byte count is the total number of bytes in the FIFO that are available to the external master. For IN endpoints, the byte count is determined by the PKSTAT bit.

7.7.2 PKSTAT EPxPFH.6

If PKTSTAT = 0 (default), the PF refers to the entire IN endpoint FIFO. If PKTSTAT = 1, the PF refers to the current packet. For IN endpoints, the PF can apply to either the entire FIFO, comprising multiple packets, or only to the current packet being filled.

PKTSTAT	PF applies to	EPnPFH:L format
0	Number of committed packets + current packet bytes	PKTS[] PFC[]
1	Current packet bytes only	PFC[]

7.7.3 IN:PKTS(2:0)/OUT:PFC[12:10] EPxPFH[5:3]

These three bits have a different meaning, depending on whether this is an IN or OUT endpoint, and if IN endpoint, the value of the PKTSTAT bit.

7.7.3.1 IN Endpoints

When PKTSTAT = 0 (default), the PF considers when there are PKTS packets plus PFC bytes in the FIFO. PKTS(2:0) determines how many packets are considered, according to the following table.

Table 7-5. PKTS Bits

PKTS2	PKTS1	PKTS0	Number of packets
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4

When PKTSTAT = 1, the PF considers when there are PFC bytes in the FIFO, no matter how many packets are in the FIFO. The PKTS(2:0) bits are ignored.

7.7.3.2 OUT Endpoints

The PF considers when there are PFC bytes in the FIFO.

7.8 EPxISOINPKTS Registers 0x1A–0x1D

EP2ISOINPKTS, EP4ISOINPKTS, EP6ISOINPKTS, EP8ISOINPKTS

0x1A, 0x1B, 0x1C, 0x1D

Bit #	7	6	5	4	3	2	1	0
Bit Name	0	0	0	0	0	INPPF2	INPPF1	INPPF0
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	1

For ISOCRONOUS IN endpoints only, these registers determine the number of packets per frame (only one per frame for full-speed mode) or microframe (up to three per microframe for high-speed mode), according to the following table.

Table 7-6. EPxISOINPKTS

INPPF1	INPPF0	Packets
0	0	Invalid
0	1	1 (default)
1	0	2
1	1	3

7.9 EPxxFLAGS Registers 0x1E–0x1F

The EPxxFLAGS provide an alternate way of checking the status of the endpoint FIFO flags. If enabled, the SX2 can interrupt the external master when a flag is asserted, and the external master can read these two registers to determine the state of the FIFO flags. If the INFM1 and/or OEP1 bits are set, then the EPxEF and EPxFF bits are actually empty +1 and full -1.

EP24FLAGS

0x1E

Bit #	7	6	5	4	3	2	1	0
Bit Name	0	EP4PF	EP4EF	EP4FF	0	EP2PF	EP4EF	EP4FF
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	1

EP68FLAGS

0x1F

Bit #	7	6	5	4	3	2	1	0
Bit Name	0	EP8PF	EP8EF	EP8FF	0	EP6PF	EP6EF	EP6FF
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	1

7.9.1 EPxPF Bit 6, Bit 2

This bit is the current state of endpoint x's programmable flag.

7.9.2 EPxEF Bit 5, Bit 1

This bit is the current state of endpoint x's empty flag. EPxEF = 1 if the endpoint is empty.

7.9.3 EPxFF Bit 4, Bit 0

This bit is the current state of endpoint x's full flag. EPxFF = 1 if the endpoint is full.

7.10 INPKTEND/FLUSH Register 0x20

This register allows the external master to duplicate the function of the PKTEND pin. The register also allows the external master to selectively flush endpoint FIFO buffers..

INPKTEND/FLUSH

0x20

Bit #	7	6	5	4	3	2	1	0
Bit Name	FIFO8	FIFO6	FIFO4	FIFO2	EP3	EP2	EP1	EP0
Read/Write	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bit [4..7]: These bits allows the external master to selectively flush any or all of the endpoint FIFOs. By writing the desired endpoint FIFO bit, SX2 logic flushes the selected FIFO. For example setting bit 7 flushes endpoint 8 FIFO.

Bit [3..0]: These bits are is used only for IN transfers. By writing the desired endpoint number (2,4,6 or 8), SX2 logic automatically commits an IN buffer to the USB host. For example, for committing a packet through endpoint 6, set the lower nibble to 6: set bits 1 and 2 high.

7.11 USBFRAMEH/L Registers 0x2A, 0x2B

Every millisecond, the USB host sends an 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.

USBFRAMEH

0x2A

Bit #	7	6	5	4	3	2	1	0
Bit Name	0	0	0	0	0	FC10	FC9	FC8
Read/Write	R	R	R	R	R	R	R	R
Reset	X	X	X	X	X	X	X	x

USBFRAMEL

0x2B

Bit #	7	6	5	4	3	2	1	0
Bit Name	FC7	FC6	FC5	FC4	FC3	FC2	FC1	FC0
Read/Write	R	R	R	R	R	R	R	R
Reset	X	X	X	X	X	X	X	X

One use of the frame count is to respond to the USB SYNC_FRAME Request. If the SX2 detects a missing or garbled SOF, the SX2 generates an internal SOF and increments USBFRAMEL–USBFRAMEH.

7.12 MICROFRAME Registers 0x2C

MICROFRAME								0x2C
Bit #	7	6	5	4	3	2	1	0
Bit Name	0	0	0	0	0	MF2	MF1	MF0
Read/Write	R	R	R	R	R	R	R	R
Reset	X	X	X	X	X	X	X	x

MICROFRAME contains a count 0–7 that indicates which of the 125 microsecond microframes last occurred.

This register is active only when SX2 is operating in high-speed mode (480 Mbits/sec).

7.13 FNADDR Register 0x2D

During the USB enumeration process, the host sends a device a unique 7-bit address that the SX2 copies into this register. There is normally no reason for the external master to know its USB device address because the SX2 automatically responds only to its assigned address.

FNADDR								0x2D
Bit #	7	6	5	4	3	2	1	0
Bit Name	HSGRANT	FA6	FA5	FA4	FA3	FA2	FA1	FA0
Read/Write	R	R	R	R	R	R	R	R
Reset	X	X	X	X	X	X	X	X

Bit 7: HSGRANT, Set to 1 if the SX2 enumerated at high speed. Set to 0 if the SX2 enumerated at full speed.

Bit[6..0]: Address set by the host.

7.14 INTENABLE Register 0x2E

This register is used to enable/disable the various interrupt sources, and by default all interrupts are enabled.

ITENABLE								0x2E
Bit #	7	6	5	4	3	2	1	0
Bit Name	SETUP	EP0BUF	FLAGS	ENUMOK	1	1	BUSACTIVITY	READY
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

7.14.1 SETUP Bit 7

Setting this bit to a 1 enables an interrupt when a set-up packet is received from the USB host.

7.14.2 EP0BUF Bit 6

Setting this bit to a 1 enables an interrupt when the Endpoint 0 buffer becomes available.

7.14.3 FLAGS Bit 5

Setting this bit to a 1 enables an interrupt when an OUT endpoint FIFO's state transitions from empty to not-empty.

7.14.4 ENUMOK Bit 2

Setting this bit to a 1 enables an interrupt when SX2 enumeration is complete.

7.14.5 BUSACTIVITY Bit 1

Setting this bit to a 1 enables an interrupt when the SX2 detects an absence or presence of bus activity.

7.14.6 READY Bit 0

Setting this bit to a 1 enables an interrupt when the SX2 has powered on and performed an internal self-test.

7.15 DESC Register 0x30

This register address is used to write the 500-byte descriptor RAM. The external master writes two bytes (four command data transfers) to this address corresponding to the length of the descriptor or VID/PID/DID data to be written. The external master then consecutively writes that number of bytes into the descriptor RAM in nibble format. For complete details, refer to Section 4.0.

7.16 EP0BUF Register 0x31

This register address is used to access the 64-byte Endpoint 0 buffer. The external master can read or write to this register to complete Endpoint 0 data transfers. For complete details, refer to Section 5.0.

7.17 SETUP Register 0x32

This register address is used to access the 8-byte set-up packet received from the USB host. If the external master writes to this register, it can stall Endpoint 0. For complete details, refer to Section 5.0.

7.18 EP0BC Register 0x33

This register address is used to access the byte count of Endpoint 0. For Endpoint 0 OUT transfers, the external master can read this register to get the number of bytes transferred from the USB host. For Endpoint 0 IN transfers, the external master writes the number of bytes in the Endpoint 0 buffer to transfer the bytes to the USB host. For complete details, refer to Section 5.0.

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 5.25V
 DC Voltage Applied to
 Outputs in High-Z State -0.5V to $V_{CC} + 0.5V$
 Power Dissipation 936 mW
 Static Discharge Voltage > 2000V

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 Parallel Resonant

10.0 DC Electrical Characteristics

Table 10-1. DC Characteristics

Parameter	Description	Conditions ^[7]	Min.	Typ.	Max.	Unit
V_{CC}	Supply Voltage		3.0	3.3	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	$I_{OUT} = 4 \text{ mA}$	2.4			V
V_{OL}	Output Voltage Low	$I_{OUT} = -4 \text{ mA}$			0.4	V
I_{OH}	Output Current High				4	mA
I_{OL}	Output Current Low				4	mA
C_{IN}	Input Pin Capacitance	Except D+/D-			10	pF
		D+/D-			15	pF
I_{SUSP}	Suspend Current	includes 1.5k integrated pull-up		250		μA
I_{SUSP}	Suspend Current	excluding 1.5k integrated pull-up		30		μA
I_{CC}	Supply Current	Connected to USB at high speed		180		mA
		Connected to USB at full speed		110		mA
T_{RESET}	RESET Time after valid power	$V_{CC} \text{ min} = 3.0V$	1.91			mS

Note:

7. Specific conditions for I_{CC} measurements: HS typical 3.3V, 25°C, 48 MHz; FS typical 3.3V, 25°C, 48 MHz.

11.0 AC Electrical Characteristics

11.1 USB Transceiver

USB 2.0-certified compliant in full and high speed.

11.2 Command Interface

11.2.1 Command Synchronous Read

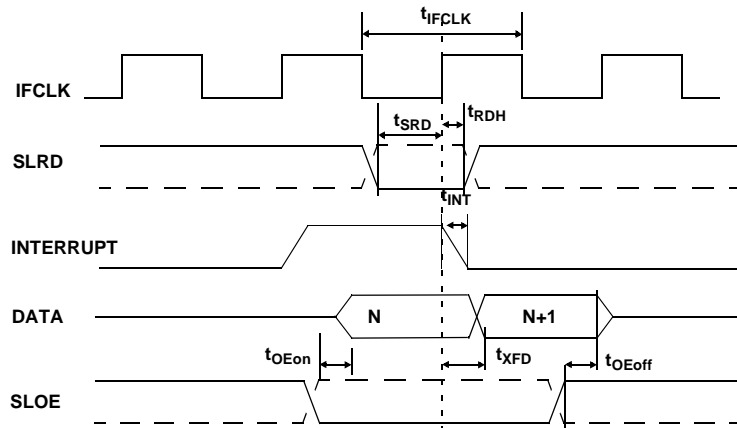


Figure 11-1. Command Synchronous Read Timing Diagram^[8]

Table 11-1. Command Synchronous Read Parameters with Internally Sourced IFCLK

Parameter	Description	Min.	Max.	Unit
t_{IFCLK}	IFCLK period	20.83		ns
t_{SRD}	SLRD to Clock Set-up Time	18.7		ns
t_{RDH}	Clock to SLRD Hold Time	0		ns
t_{OEon}	SLOE Turn-on to FIFO Data Valid		10.5	ns
t_{OEoff}	SLOE Turn-off to FIFO Data Hold		10.5	ns
t_{INT}	Clock to INTERRUPT Output Propagation Delay		9.5	ns
t_{XFD}	Clock to Command Data Output Propagation Delay		11	ns

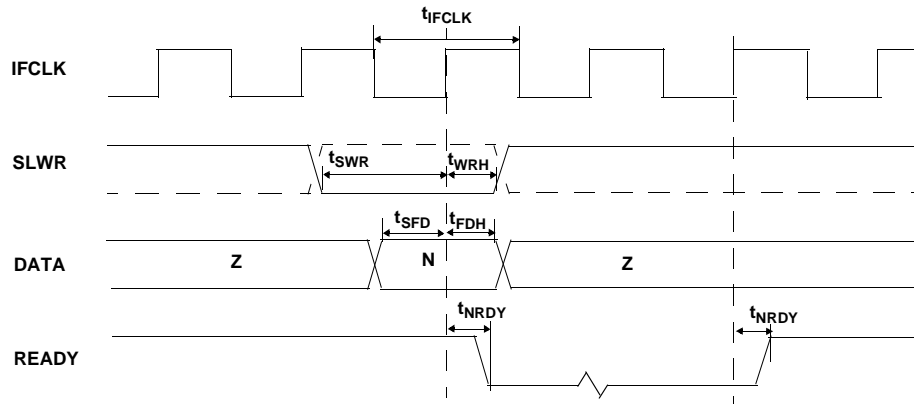
Table 11-2. Command Synchronous Read with Externally Sourced IFCLK^[9]

Parameter	Description	Min.	Max.	Unit
t_{IFCLK}	IFCLK Period	20	200	ns
t_{SRD}	SLRD to Clock Set-up Time	12.7		ns
t_{RDH}	Clock to SLRD Hold Time	3.7		ns
t_{OEon}	SLOE Turn-on to FIFO Data Valid		10.5	ns
t_{OEoff}	SLOE Turn-off to FIFO Data Hold		10.5	ns
t_{INT}	Clock to INTERRUPT Output Propagation Delay		13.5	ns
t_{XFD}	Clock to Command Data Output Propagation Delay		15	ns

Notes:

8. Dashed lines denote signals with programmable polarity.
9. Externally sourced IFCLK must not exceed 50 MHz.

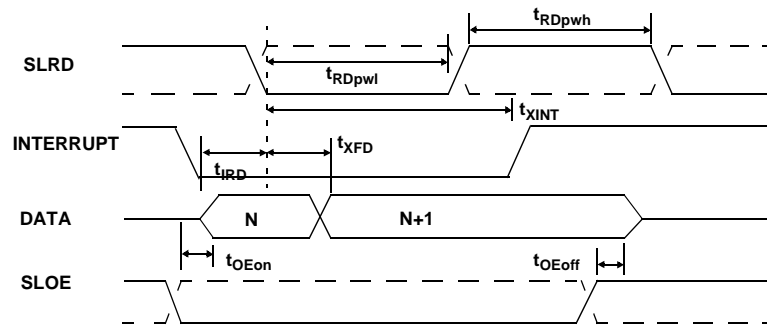
11.2.2 Command Synchronous Write


Figure 11-2. Command Synchronous Write Timing Diagram^[8]
Table 11-3. Command Synchronous Write Parameters with Internally Sourced IFCLK

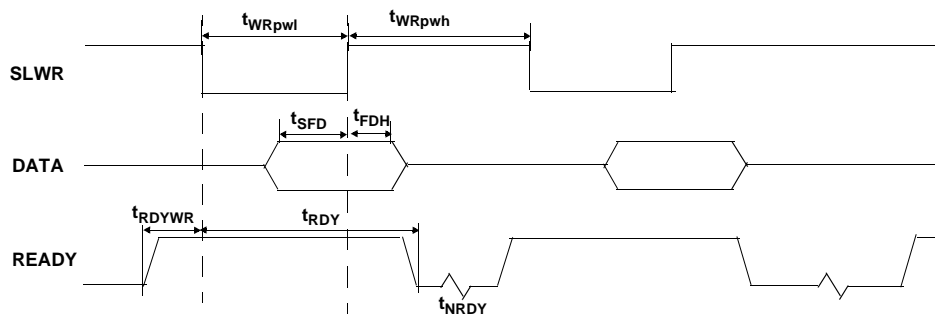
Parameter	Description	Min.	Max.	Unit
t_{IFCLK}	IFCLK Period	20.83		ns
t_{SWR}	SLWR to Clock Set-up Time	18.1		ns
t_{WRH}	Clock to SLWR Hold Time	0		ns
t_{SFD}	Command Data to Clock Set-up Time	9.2		ns
t_{FDH}	Clock to Command Data Hold Time	0		ns
t_{NRDY}	Clock to READY Output Propagation Time		9.5	ns

Table 11-4. Command Synchronous Write Parameters with Externally Sourced IFCLK^[9]

Parameter	Description	Min.	Max.	Unit
t_{IFCLK}	IFCLK Period	20	200	ns
t_{SWR}	SLWR to Clock Set-up Time	12.1		ns
t_{WRH}	Clock to SLWR Hold Time	3.6		ns
t_{SFD}	Command Data to Clock Set-up Time	3.2		ns
t_{FDH}	Clock to Command Data Hold Time	4.5		ns
t_{NRDY}	Clock to READY Output Propagation Time		13.5	ns

11.2.3 Command Asynchronous Read

Figure 11-3. Command Asynchronous Read Timing Diagram^[8]
Table 11-5. 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	0		ns
t_{XINT}	SLRD to INTERRUPT		70	ns
t_{XFD}	SLRD to Command Data Output Propagation Delay		15	ns
t_{OEon}	SLOE Turn-on to FIFO Data Valid		10.5	ns
t_{OEoff}	SLOE Turn-off to FIFO Data Hold		10.5	ns

11.2.4 Command Asynchronous Write

Figure 11-4. Command Asynchronous Write Timing Diagram^[8]
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 Command DATA Set-up Time	10		ns
t_{FDH}	Command DATA to SLWR Hold Time	10		ns
t_{RDYWR}	READY to SLWR Time	0		ns
t_{RDY}	SLWR to READY		70	ns

11.3 FIFO Interface

11.3.1 Slave FIFO Synchronous Read

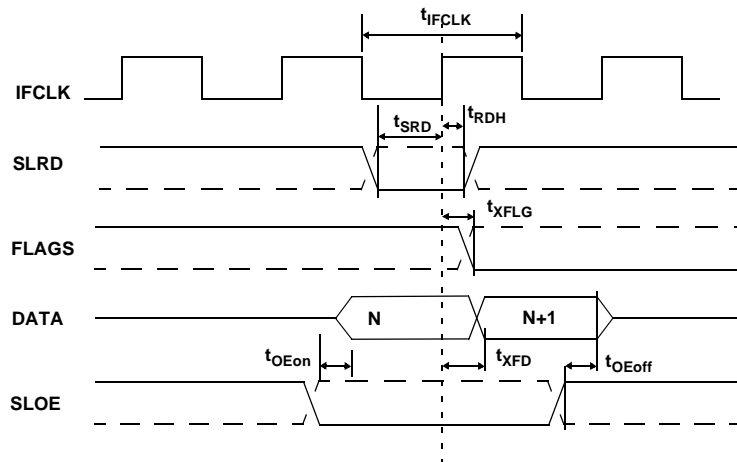


Figure 11-5. Slave FIFO Synchronous Read Timing Diagram^[8]

Table 11-7. Slave FIFO Synchronous Read with Internally Sourced IFCLK^[9]

Parameter	Description	Min.	Max.	Unit
t_{IFCLK}	IFCLK Period	20.83		ns
t_{SRD}	SLRD to Clock Set-up Time	18.7		ns
t_{RDH}	Clock to SLRD Hold Time	0		ns
t_{OEon}	SLOE Turn-on to FIFO Data Valid		10.5	ns
t_{OEoff}	SLOE Turn-off to FIFO Data Hold		10.5	ns
t_{XFLG}	Clock to FLAGS Output Propagation Delay		9.5	ns
t_{XFD}	Clock to FIFO Data Output Propagation Delay		11	ns

Table 11-8. Slave FIFO Synchronous Read with Externally Sourced IFCLK^[9]

Parameter	Description	Min.	Max.	Unit
t_{IFCLK}	IFCLK Period	20	200	ns
t_{SRD}	SLRD to Clock Set-up Time	12.7		ns
t_{RDH}	Clock to SLRD Hold Time	3.7		ns
t_{OEon}	SLOE Turn-on to FIFO Data Valid		10.5	ns
t_{OEoff}	SLOE Turn-off to FIFO Data Hold		10.5	ns
t_{XFLG}	Clock to FLAGS Output Propagation Delay		13.5	ns
t_{XFD}	Clock to FIFO Data Output Propagation Delay		15	ns

11.3.2 Slave FIFO Synchronous Write

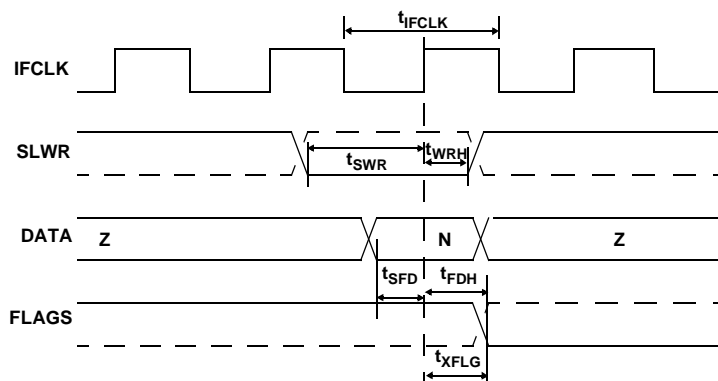

 Figure 11-6. Slave FIFO Synchronous Write Timing Diagram^[8]

 Table 11-9. Slave FIFO Synchronous Write Parameters with Internally Sourced IFCLK^[9]

Parameter	Description	Min.	Max.	Unit
t_{IFCLK}	IFCLK Period	20.83		ns
t_{SWR}	SLWR to Clock Set-up Time	18.1		ns
t_{WRH}	Clock to SLWR Hold Time	0		ns
t_{SFD}	FIFO Data to Clock Set-up Time	9.2		ns
t_{FDH}	Clock to FIFO Data Hold Time	0		ns
t_{XFLG}	Clock to FLAGS Output Propagation Time		9.5	ns

 Table 11-10. Slave FIFO Synchronous Write Parameters with Externally Sourced IFCLK^[9]

Parameter	Description	Min.	Max.	Unit
t_{IFCLK}	IFCLK Period	20		ns
t_{SWR}	SLWR to Clock Set-up Time	12.1		ns
t_{WRH}	Clock to SLWR Hold Time	3.6		ns
t_{SFD}	FIFO Data to Clock Set-up Time	3.2		ns
t_{FDH}	Clock to FIFO Data Hold Time	4.5		ns
t_{XFLG}	Clock to FLAGS Output Propagation Time		13.5	ns

11.3.3 Slave FIFO Synchronous Packet End Strobe

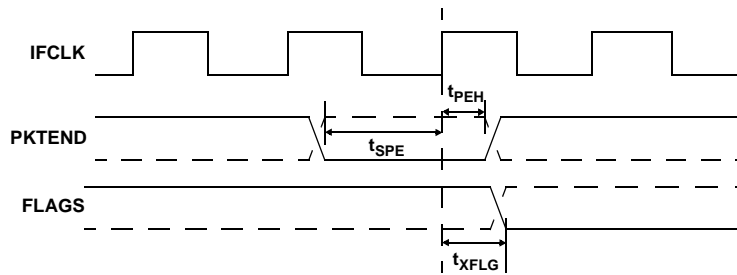


Figure 11-7. Slave FIFO Synchronous Packet End Strobe Timing Diagram^[8]

Table 11-11. Slave FIFO Synchronous Packet End Strobe Parameters, Internally Sourced IFCLK^[9]

Parameter	Description	Min.	Max.	Unit
t_{IFCLK}	IFCLK Period	20.83		ns
t_{SPE}	PKTEND to Clock Set-up Time	14.6		ns
t_{PEH}	Clock to PKTEND Hold Time	0		ns
t_{XFLG}	Clock to FLAGS Output Propagation Delay		9.5	ns

Table 11-12. Slave FIFO Synchronous Packet End Strobe Parameters, Externally Sourced IFCLK^[9]

Parameter	Description	Min.	Max.	Unit
t_{IFCLK}	IFCLK Period	20	200	ns
t_{SPE}	PKTEND to Clock Set-up Time	8.6		ns
t_{PEH}	Clock to PKTEND Hold Time	2.5		ns
t_{XFLG}	Clock to FLAGS Output Propagation Delay		13.5	ns

11.3.4 Slave FIFO Synchronous Address

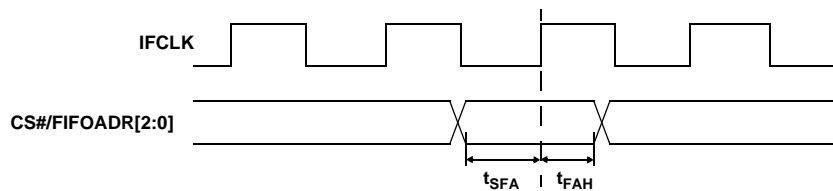
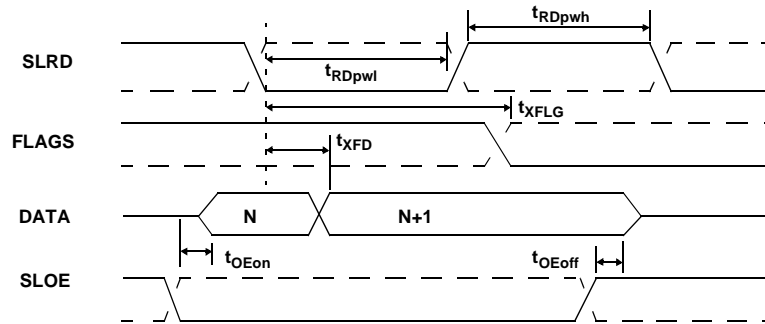


Figure 11-8. Slave FIFO Synchronous Address Timing Diagram

Table 11-13. Slave FIFO Synchronous Address Parameters^[9]

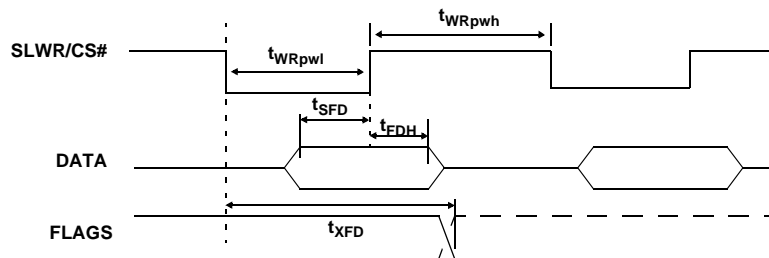
Parameter	Description	Min.	Max.	Unit
t_{IFCLK}	Interface Clock Period	20	200	ns
t_{SFA}	FIFOADR[2:0] to Clock Set-up Time	25		ns
t_{FAH}	Clock to FIFOADR[2:0] Hold Time	10		ns

11.3.5 Slave FIFO Asynchronous Read


Figure 11-9. Slave FIFO Asynchronous Read Timing Diagram^[8]
Table 11-14. Slave FIFO Asynchronous Read Parameters^[10]

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		15	ns
t_{OEon}	SLOE Turn-on to FIFO Data Valid		10.5	ns
t_{OEoff}	SLOE Turn-off to FIFO Data Hold		10.5	ns

11.3.6 Slave FIFO Asynchronous Write


Figure 11-10. Slave FIFO Asynchronous Write Timing Diagram^[8]
Table 11-15. Slave FIFO Asynchronous Write Parameters with Internally Sourced IFCLK^[10]

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 Set-up Time	10		ns
t_{FDH}	FIFO DATA to SLWR Hold Time	10		ns
t_{XFD}	SLWR to FLAGS Output Propagation Delay		70	ns

Note:

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

11.3.7 Slave FIFO Asynchronous Packet End Strobe

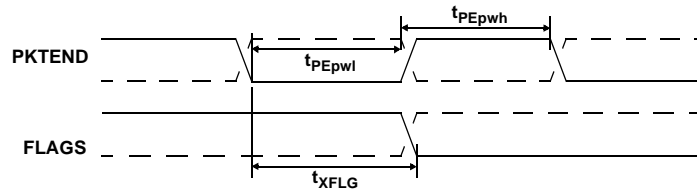


Figure 11-11. Slave FIFO Asynchronous Packet End Strobe Timing Diagram

Table 11-16. Slave FIFO Asynchronous Packet End Strobe Parameters^[10]

Parameter	Description	Min.	Max.	Unit
t_{PEpwl}	PKTEND Pulse Width LOW	50		ns
t_{PEpwh}	PKTEND Pulse Width HIGH	50		ns
t_{XFLG}	PKTEND to FLAGS Output Propagation Delay		70	ns

11.3.8 Slave FIFO Asynchronous Address

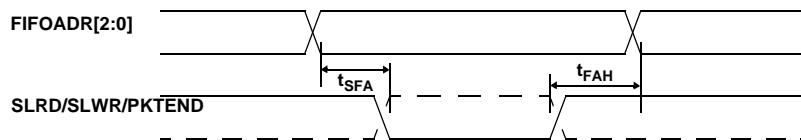


Figure 11-12. Slave FIFO Asynchronous Address Timing Diagram^[8]

Table 11-17. Slave FIFO Asynchronous Address Parameters^[10]

Parameter	Description	Min.	Max.	Unit
t_{SFA}	FIFOADR[2:0] to RD/WR/PKTEND Set-up Time	10		ns
t_{FAH}	SLRD/PKTEND to FIFOADR[2:0] Hold Time	20		ns
t_{FAH}	SLWR to FIFOADR[2:0] Hold Time	TBD ^[11]		ns

11.4 Slave FIFO Address to Flags/Data

Following timings is applicable to synchronous and asynchronous interface.

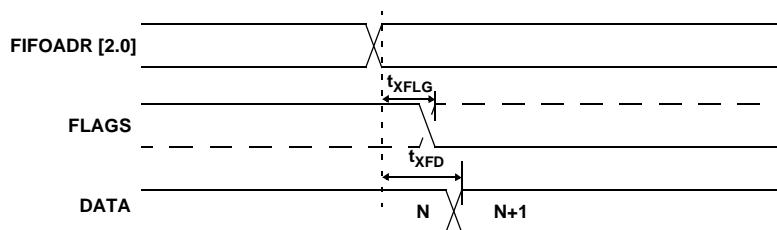


Figure 11-13. Slave FIFO Address to Flags/Data Timing Diagram^[8]

Table 11-18. Slave FIFO Address to Flags/Data Parameters

Parameter	Description	Min.	Max.	Unit
t_{XFLG}	FIFOADR[2:0] to FLAGS Output Propagation Delay		10.7	ns
t_{XFD}	FIFOADR[2:0] to FIFODATA Output Propagation Delay		14.3	ns

Note:

11. t_{FAH} for a write function has been experimentally found to be 70 ns. This is yet to be characterized. We recommend that you use a 70 ns.

11.5 Slave FIFO Output Enable

Following timings is applicable to synchronous and asynchronous interface.

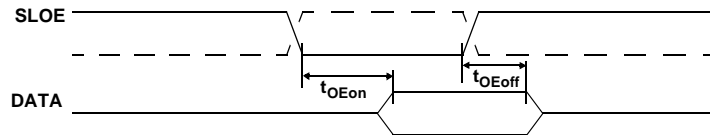


Figure 11-14. Slave FIFO Output Enable Timing Diagram^[8]

Table 11-19. Slave FIFO Output Enable Parameters

Parameter	Description	Min.	Max.	Unit
t_{OEon}	SLOE assert to FIFO DATA Output		10.5	ns
t_{OEoff}	SLOE deassert to FIFO DATA Hold		10.5	ns

11.6 Sequence Diagram

11.6.1 Single and Burst Synchronous Read Example

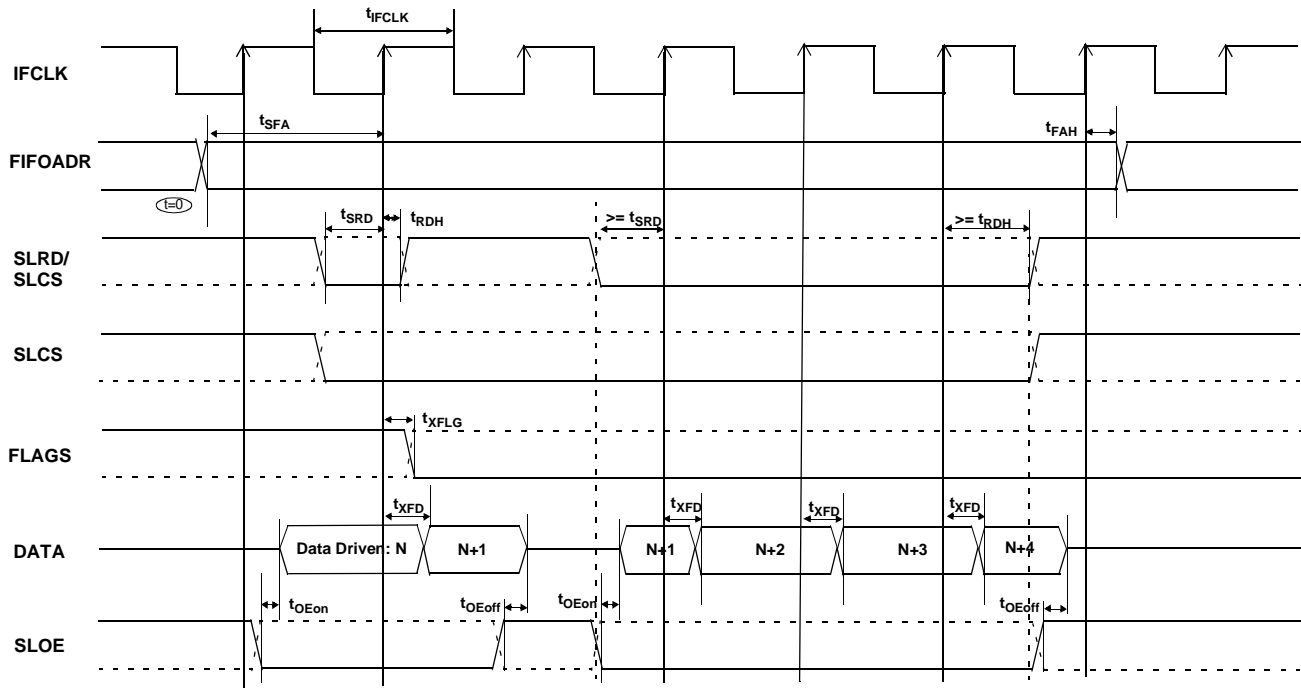


Figure 11-15. Slave FIFO Synchronous Read Sequence and Timing Diagram

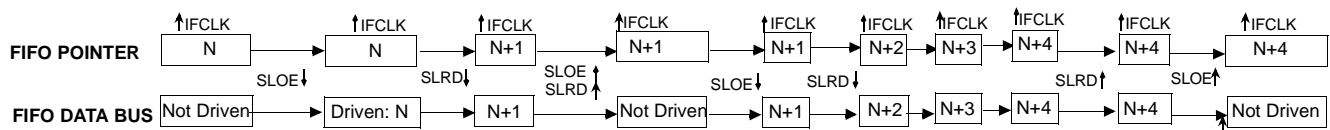


Figure 11-16. Slave FIFO Synchronous Sequence of Events Diagram

The above sequence diagrams show the timing relation between various SLAVE FIFO signals while performing a read from the FIFO in synchronous mode where the IFCLK is the clock source. The diagram shows a single read followed by a burst read.

- At $t = 0$ the FIFO address is set. Note that t_{SFA} being 25 ns implies that the FIFO address lines need to be set up a clock cycle prior to a read.
- Then the SLOE is asserted. This causes the data lines to be in a driven state with the data that the FIFO pointer is currently pointing to. In this case of the above example it is the first byte in the FIFO. Note that the data is pre-fetched and is output only when the SLOE is asserted. SLOE is an output enable only. It has no other functions besides enabling the data bus to be in a driven state.
- The SLRD and SLCS is then asserted at the same time. The SLRD must meet the setup time of t_{SRD} from the rising edge of the clock and must maintain the hold time of t_{RDH} from the rising edge of IFCLK. If SLCS is used then, SLCS must be in an asserted state when SLRD is also asserted. When SLCS is used, the internal logic basically ANDs these two signal to determine a valid condition. Hence these signals can also be tied together.
- The FIFO pointer is updated on each rising edge of the IFCLK while the SLRD is in an asserted state. This data pointed to by the updated FIFO pointer is then output after a propagation delay of t_{XFD} from the rising edge of the IFCLK. So in the above diagram data N is the first byte read from the FIFO. When the SLRD is asserted, in order to have the data output on the FIFO data bus, the SLOE MUST be in an asserted state also. The SLRD and SLOE can also be tied together.

For a burst read the SLRD and SLOE is left asserted during the entire duration of read of all the required bytes. In this burst read mode, when the SLOE is asserted the data output is the one that the FIFO pointer is pointing to. In this case after the first single read the FIFO pointer now points to N+1. After the SLRD is asserted, the FIFO pointer is incremented on the immediate rising edge of the IFCLK when the prefetched data is also output on the data bus. The FIFO pointer is incremented on each subsequent rising edge of the IFCLK while the SLRD is in an asserted state.

11.6.2 Single and Burst Synchronous Write

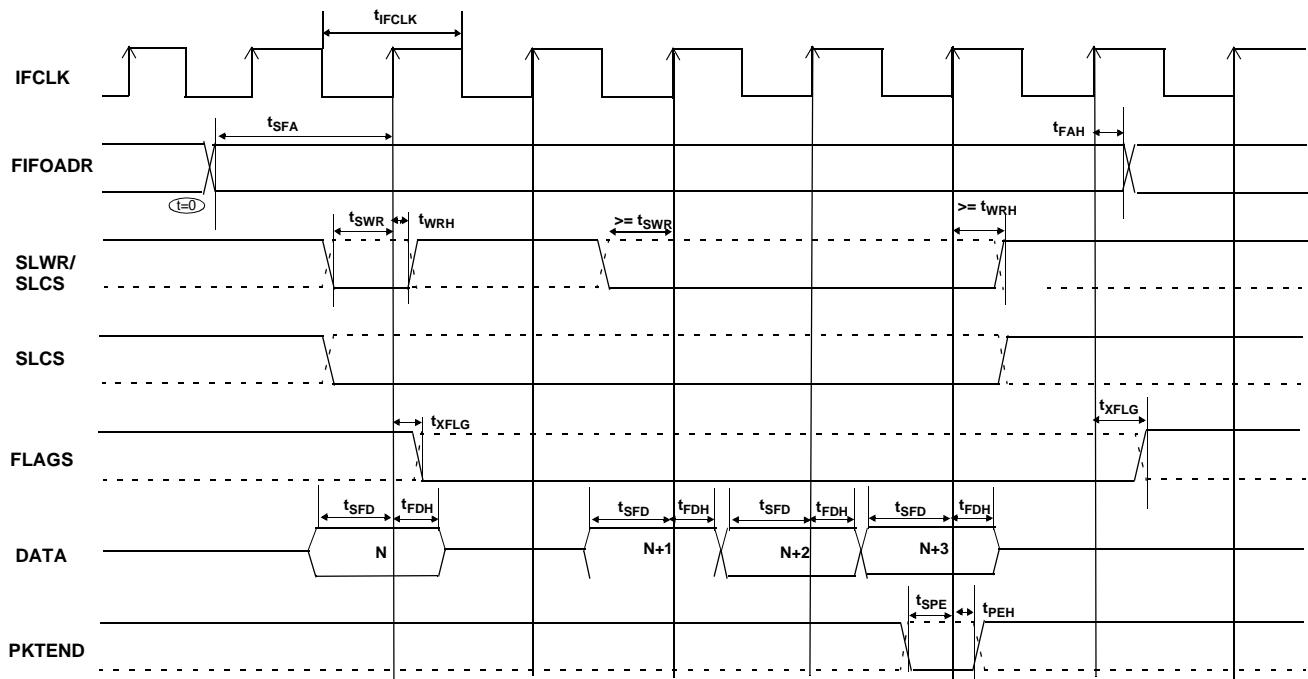


Figure 11-17. Slave FIFO Synchronous Write Sequence and Timing Diagram^[8]

The above sequence diagram shows the timing relation between various signals while performing a write to the FIFO in synchronous mode where the IFCLK is the clock source. The diagram shows a single write followed by burst write of 3 bytes and committing the 4 bytes short packet using the PKTEND pin.

- At $t = 0$ the FIFO address is set. Note that t_{SFA} being 25 ns implies that the FIFO address lines need to be set up a clock cycle prior to a read.
- Data is then written to the data bus t_{SFD} before the rising edge of the IFCLK to meet the data setup time.
- SLWR and SLCS are asserted at the same time. The SLWR must meet the setup time of t_{SWR} from the rising edge of the clock and must maintain the hold time of t_{WRH} measured from the rising edge of IFCLK. If SLCS is used then, SLCS must be in an

asserted state when SLWR is also asserted. The internal logic ANDs these two signals to determine a valid condition. Hence these signals can also be tied together.

- Asserting the SLWR causes the data on the data bus to be written to the FIFO and the FIFO pointer incremented. The flag is also updated after t_{XFLG} from the rising edge of the clock.

For a burst write the SLWR is left asserted during the entire duration of writing all the required bytes. In this burst write mode once the SLWR is asserted the data on the FIFO data bus is written to the FIFO on every rising edge of the IFCLK. The FIFO pointer is also updated on each rising edge of the IFCLK. In the above diagram once the four bytes are written to the FIFO, the SLWR is de-asserted. The short 4-byte packet can be committed to the host using the PKTEND pin.

There is no specific timing requirement that needs to be met for asserting PKTEND pin with regards to asserting SLWR signal. The PKTEND pin can be asserted anytime. In the scenario above the number of packets committed include the byte written last to the FIFO on the same rising edge of the IFCLK as the PKTEND is asserted. The FIFOADDR lines should be held constant during the PKTEND pin assertion.

11.6.3 Sequence Diagram of a Single and Burst Asynchronous Read

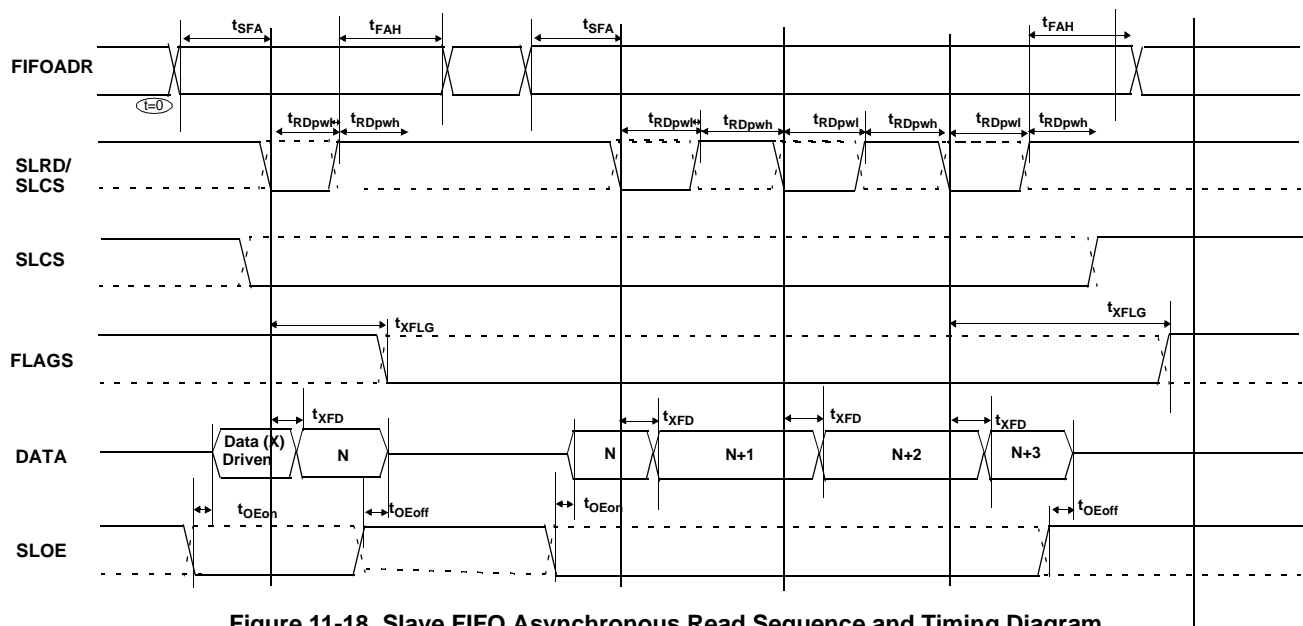


Figure 11-18. Slave FIFO Asynchronous Read Sequence and Timing Diagram

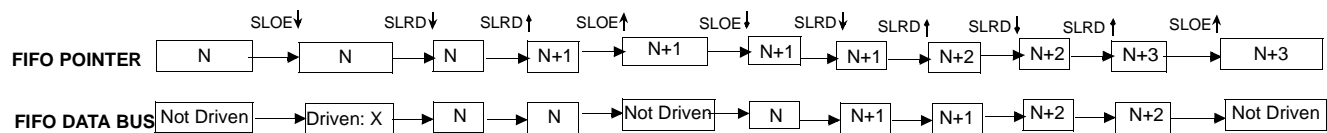


Figure 11-19. Slave FIFO Asynchronous Read Sequence of Events Diagram

The above sequence and timing diagram shows the timing relation between various signals while performing a read from the FIFO in asynchronous mode. The diagram shows a single read followed by a burst read.

- At $t = 0$ the FIFO address is set.
- SLOE is asserted. This causes the data lines to be in a driven state. The data output is the previous data, whatever was output last time around.
- The SLRD and SLCS is asserted at the same time. The SLRD must meet the minimum active pulse of t_{RDpwl} and minimum de-active pulse width of t_{RDpwh} . If SLCS is used then, SLCS must be in an asserted state when SLRD is also asserted. The internal logic basically ANDs these two signal to determine a valid condition. Hence these signals can be tied together.
- The data that is output on asserting the SLRD is the actual data from the FIFO. This data is output after a propagation delay of t_{XFD} from the activating edge of the SLRD. So in the above diagram data N is the actual first byte data read from the FIFO. When the SLRD is asserted, in order to have the data output on the FIFO data bus, the SLOE MUST be in an asserted state also. The SLRD and SLOE can also be tied together.

In burst read mode when the SLRD is asserted, the data is in a driven state and is the previous data. Once SLRD is asserted, the data from the FIFO is output on the data bus (when SLOE is also in an asserted state) and the FIFO pointer is incremented. Basically the data is fetched at the time of the read.

11.6.4 Sequence Diagram of a Single and Burst Asynchronous Write

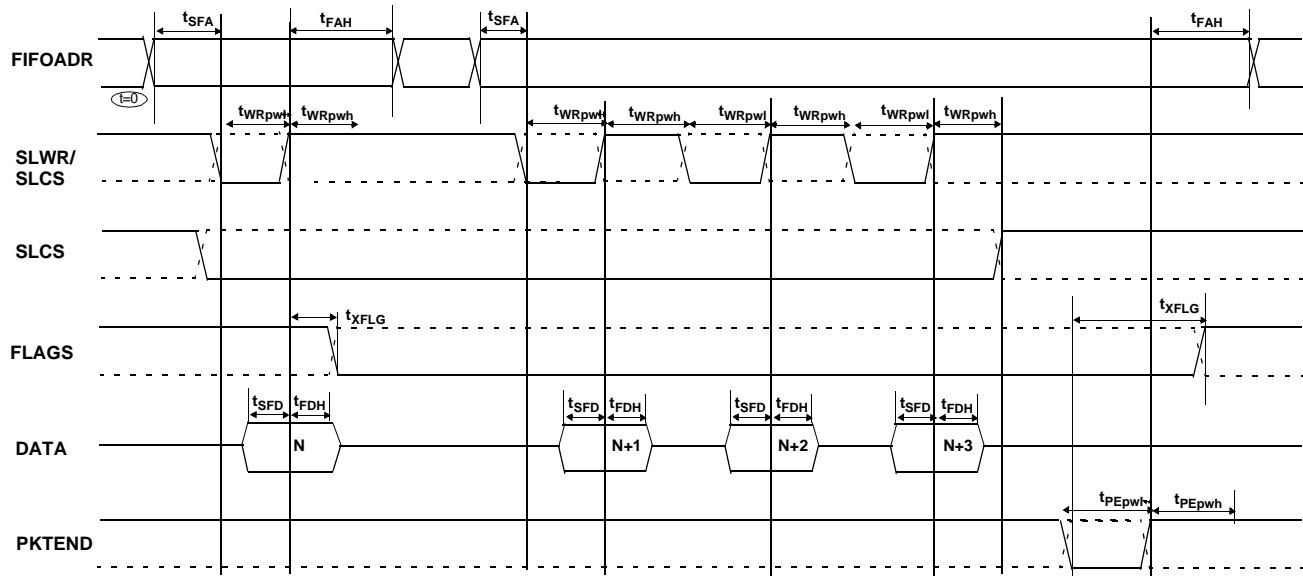


Figure 11-20. Slave FIFO Asynchronous Write Sequence and Timing Diagram^[8]

The above sequence diagram shows the timing relation between various signals while performing a write to the FIFO in asynchronous mode. The diagram shows a single write followed by burst write of 3 bytes and committing the 4-bytes-short packet using the PKTEND pin.

- At $t = 0$ the FIFO address is set making sure that it meets the setup time of t_{SFA} .
- SLWR and SLCS is asserted at the same time. The SLWR must meet the minimum active pulse of t_{WRpwl} and minimum deactive pulse width of t_{WRpwh} . If SLCS is used then, SLCS must be in an asserted state when SLWR is also asserted. The internal logic basically ANDs these two signal to determine a valid condition. Hence these signals can be tied together.
- Data is then written to the data bus t_{SFD} before the deasserting edge of the SLWR to meet the data setup time.
- De-asserting the SLWR causes the data on the data bus to be written to the FIFO and the FIFO pointer incremented. The flag is also updated after t_{XFLG} from the deasserting edge of SLWR.

In this burst write mode once SLWR is deasserted, the data is written to the FIFO and the FIFO pointer is incremented. Asserting the SLWR causes the data to be written to the FIFO and the FIFO pointer to be incremented to point to the next byte in the FIFO. The FIFO pointer is post incremented.

In the above diagram once the four bytes are written to the FIFO and the SLWR is de-asserted, the short 4-byte packet can be committed to the host using the PKTEND pin. The external device should be designed so as to not have SLWR and PKTEND pulsed at the same time. It should be designed so as to assert the PKTEND pin after the SLWR is deasserted and met the minimum deasserted pulse width. The FIFOADDR lines should be held constant during the PKTEND pin assertion.

12.0 Default Descriptor

```

//Device Descriptor
18,          //Descriptor length
1,          //Descriptor type
00,02,      //Specification Version (BCD)
00,         //Device class
00,         //Device sub-class
00,         //Device sub-sub-class
64,         //Maximum packet size
LSB(VID),MSB(VID), //Vendor ID
LSB(PID),MSB(PID), //Product ID
LSB(DID),MSB(DID), //Device ID
1,          //Manufacturer string index
2,          //Product string index
0,          //Serial number string index
1,          //Number of configurations

//DeviceQualDscr
10,         //Descriptor length
6,          //Descriptor type
0x00,0x02, //Specification Version (BCD)
00,         //Device class
00,         //Device sub-class
00,         //Device sub-sub-class
64,         //Maximum packet size
1,          //Number of configurations
0,          //Reserved

//HighSpeedConfigDscr
9,          //Descriptor length
2,          //Descriptor type
46,         //Total Length (LSB)
0,          //Total Length (MSB)
1,          //Number of interfaces
1,          //Configuration number
0,          //Configuration string
0xA0,      //Attributes (b7 - buspwr, b6 - selfpwr, b5 - rwu)
50,         //Power requirement (div 2 ma)

//Interface Descriptor
9,          //Descriptor length
4,          //Descriptor type
0,          //Zero-based index of this interface
0,          //Alternate setting
4,          //Number of end points
0xFF,      //Interface class
0x00,      //Interface sub class
0x00,      //Interface sub sub class
0,          //Interface descriptor string index

//Endpoint Descriptor
7,          //Descriptor length
5,          //Descriptor type
0x02,      //Endpoint number, and direction
2,          //Endpoint type
0x00,      //Maximum packet size (LSB)
0x02,      //Max packet size (MSB)
0x00,      //Polling interval

```

```
//Endpoint Descriptor
7,          //Descriptor length
5,          //Descriptor type
0x04,      //Endpoint number, and direction
2,          //Endpoint type
0x00,      //Maximum packet size (LSB)
0x02,      //Max packet size (MSB)
0x00,      //Polling interval

//Endpoint Descriptor
7,          //Descriptor length
5,          //Descriptor type
0x86,      //Endpoint number, and direction
2,          //Endpoint type
0x00,      //Maximum packet size (LSB)
0x02,      //Max packet size (MSB)
0x00,      //Polling interval

//Endpoint Descriptor
7,          //Descriptor length
5,          //Descriptor type
0x88,      //Endpoint number, and direction
2,          //Endpoint type
0x00,      //Maximum packet size (LSB)
0x02,      //Max packet size (MSB)
0x00,      //Polling interval

//FullSpeedConfigDscr
9,          //Descriptor length
2,          //Descriptor type
46,        //Total Length (LSB)
0,          //Total Length (MSB)
1,          //Number of interfaces
1,          //Configuration number
0,          //Configuration string
0xA0,      //Attributes (b7 - buspwr, b6 - selfpwr, b5 - rwu)
50,        //Power requirement (div 2 ma)

//Interface Descriptor
9,          //Descriptor length
4,          //Descriptor type
0,          //Zero-based index of this interface
0,          //Alternate setting
4,          //Number of end points
0xFF,      //Interface class
0x00,      //Interface sub class
0x00,      //Interface sub sub class
0,          //Interface descriptor string index

//Endpoint Descriptor
7,          //Descriptor length
5,          //Descriptor type
0x02,      //Endpoint number, and direction
2,          //Endpoint type
0x40,      //Maximum packet size (LSB)
0x00,      //Max packet size (MSB)
0x00,      //Polling interval
```

```
//Endpoint Descriptor
7, //Descriptor length
5, //Descriptor type
0x04, //Endpoint number, and direction
2, //Endpoint type
0x40, //Maximum packet size (LSB)
0x00, //Max packet size (MSB)
0x00, //Polling interval

//Endpoint Descriptor
7, //Descriptor length
5, //Descriptor type
0x86, //Endpoint number, and direction
2, //Endpoint type
0x40, //Maximum packet size (LSB)
0x00, //Max packet size (MSB)
0x00, //Polling interval

//Endpoint Descriptor
7, //Descriptor length
5, //Descriptor type
0x88, //Endpoint number, and direction
2, //Endpoint type
0x40, //Maximum packet size (LSB)
0x00, //Max packet size (MSB)
0x00, //Polling interval

//StringDscr

//StringDscr0
4, //String descriptor length
3, //String Descriptor
0x09,0x04, //US LANGID Code

//StringDscr1
16, //String descriptor length
3, //String Descriptor
'C',00,
'y',00,
'p',00,
'r',00,
'e',00,
's',00,
's',00,

//StringDscr2
20, //String descriptor length
3, //String Descriptor
'C',00,
'Y',00,
'7',00,
'C',00,
'6',00,
'8',00,
'0',00,
'0',00,
'1',00,
```

13.0 General PCB Layout Guidelines^[2]

The following recommendations should be followed to ensure reliable high-performance operation.

- At least a four-layer impedance controlled boards are required to maintain signal quality.
- Specify impedance targets (ask your board vendor what they can achieve).
- To control impedance, maintain trace widths and trace spacing.
- Minimize stubs to minimize reflected signals.
- Connections between the USB connector shell and signal ground must be done near the USB connector.
- Bypass/flyback caps on VBus, near connector, are recommended.
- DPLUS and DMINUS trace lengths should be kept to within 2 mm of each other in length, with preferred length of 20–30 mm.
- Maintain a solid ground plane under the DPLUS and DMINUS traces. Do not allow the plane to be split under these traces.
- It is preferred to have no vias placed on the DPLUS or DMINUS trace routing.
- Isolate the DPLUS and DMINUS traces from all other signal traces by no less than 10 mm.

14.0 Quad Flat Package No Leads (QFN) Package Design Notes

Electrical contact of the part to the Printed Circuit Board (PCB) is made by soldering the leads on the bottom surface of the package to the PCB. Hence, special attention is required to the heat transfer area below the package to provide a good thermal bond to the circuit board. A Copper (Cu) fill is to be designed into the PCB as a thermal pad under the package. Heat is transferred from the SX2 through the device's metal paddle on the bottom side of the package. Heat from here, is conducted to the PCB at the thermal pad. It is then conducted from the thermal pad to the PCB inner ground plane by a 5 x 5 array of via. A via is a plated through hole in the PCB with a finished diameter of 13 mil. The QFN's metal die paddle must be soldered to the PCB's thermal pad. Solder mask is placed on the board top side over each via to resist solder flow into the via. The mask on the top side also minimizes outgassing during the solder reflow process.

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

Figure 14-1 below display a cross-sectional area underneath the package. The cross section is of only one via. The solder paste template needs to be designed to allow at least 50% solder coverage. The thickness of the solder paste template should be 5 mil. It is recommended that "No Clean" type 3 solder paste is used for mounting the part. Nitrogen purge is recommended during reflow.

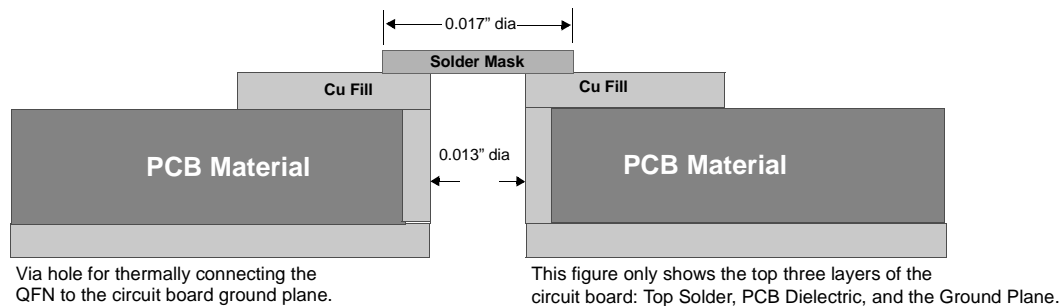


Figure 14-1. Crosssection of the Area Underneath the QFN Package

Figure 14-2a is a plot of the solder mask pattern and Figure 14-2b displays an X-Ray image of the assembly (darker areas indicate solder).

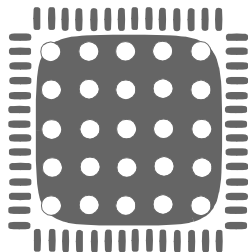


Figure 14-2. (a) Plot of the Solder Mask (White Area)

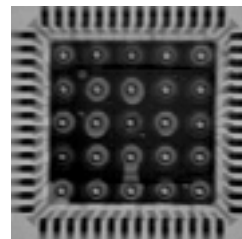


Figure 0-1. (b) X-ray Image of the Assembly

Notes:

2. Source for recommendations: *High-Speed USB Platform Design Guidelines*, http://www.usb.org/developers/data/hs_usb_pdg_r1_0.pdf.

15.0 Ordering Information

Table 15-1. Ordering Information

Ordering Code	Package Type
CY7C68001-56PVC	56 SSOP
CY7C68001-56LFC	56 QFN
CY3682	EZ-USB SX2 Development Kit

16.0 Package Diagrams

16.1 56-pin SSOP Package

56-pin Shrunk Small Outline Package 056

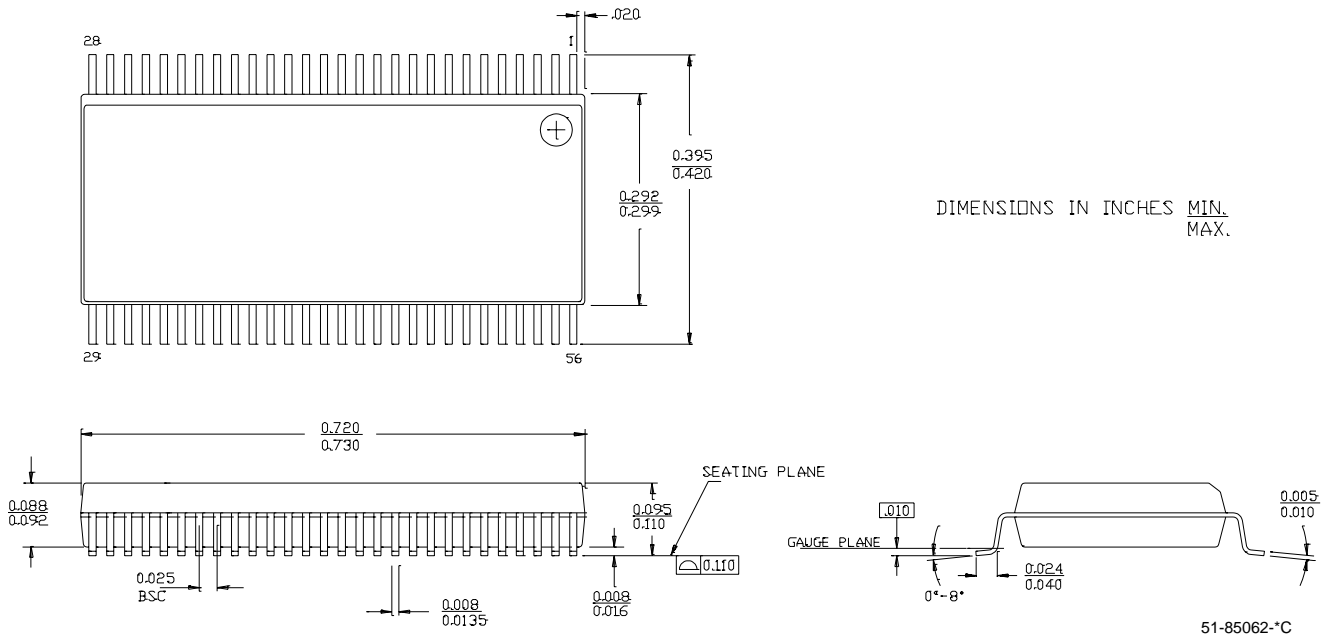


Figure 16-1. 56-lead Shrunk Small Outline Package

Note:

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

17.0 Document Revision History

Description Title: CY7C68001 EZ-USB SX2™ High-Speed USB Interface Device				
Document Number: 38-08013				
REV.	ECN No.	Issue Date	Orig. of Change	Description of Change
**	111807	06/07/02	BHA	New Data Sheet
*A	123155	02/07/03	BHA	Minor clean-up and clarification Removed references to IRQ Register and replaced them with references to Interrupt Status Byte Modified pin-out description for XTALIN and XTALOUT Added CS# timing to <i>Figure 11-10</i> , <i>Figure 11-8</i> , and <i>Figure 11-12</i> Changed Command Protocol example to IFCONFIG (0x01) Edited PCB Layout Recommendations Added AR#10691 Added USB high-speed logo
*B	126324	07/02/03	MON	Default state of registers specified in section where the register bits are defined Reorganized timing diagram presentation: First all timing related to synchronous interface, followed by timing related to asynchronous interface, followed by timing diagrams common to both interfaces Provided further information in section 3.3 regarding boot methods Provided timing diagram that encapsulates ALL relevant signals for a synchronous and asynchronous slave read and write interface Added section on (QFN) Package Design Notes FIFOADR[2:0] Hold Time (t_{FAH}) for Asynchronous FIFO Interface has been updated as follows: SLRD/PKTEND to FIFOADR[2:0] Hold Time: 20 ns; SLWR to FIFOADR[2:0] Hold Time:70 ns (recommended) Added information on the polarity of the programmable flag Fixed the Command Synchronous Write Timing Diagram Fixed the Command Asynchronous Write Timing Diagram Added information on the delay required when endpoint configuration registers are changed after SX2 has already enumerated