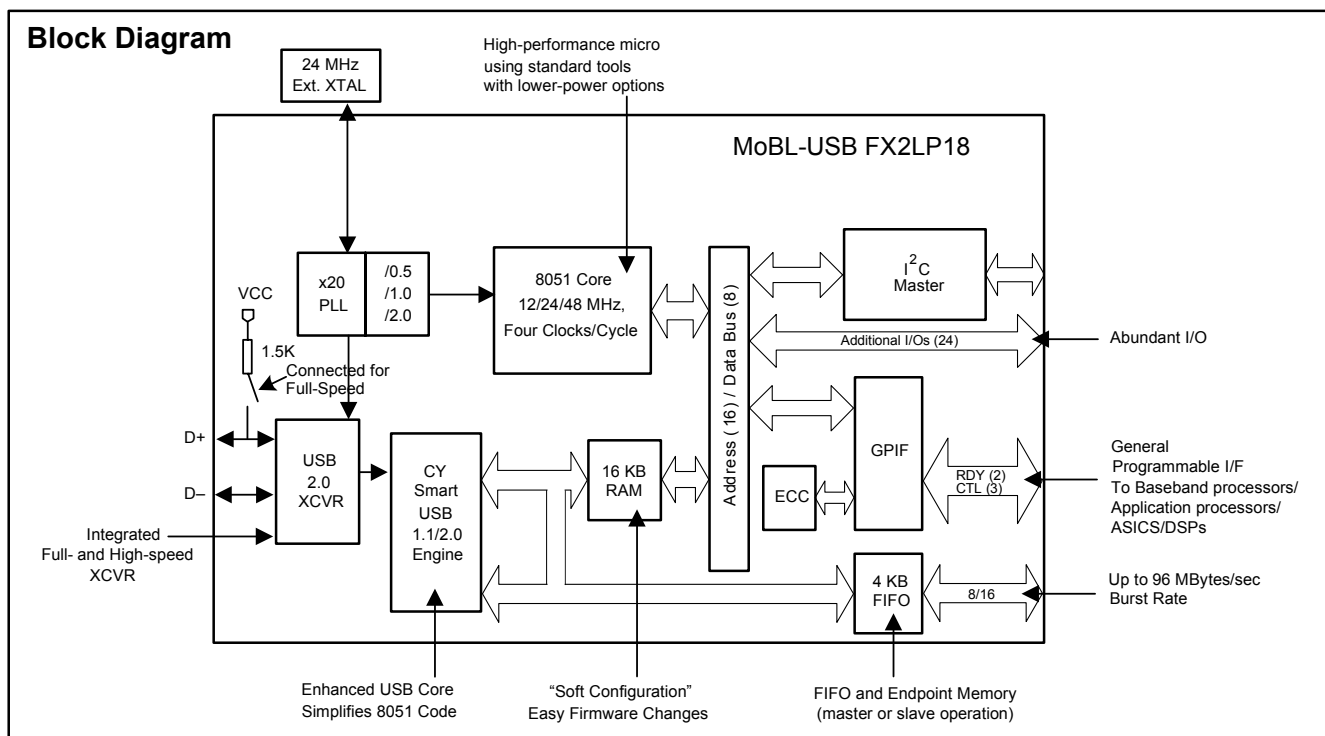


# MoBL-USB™ FX2LP18 USB Microcontroller

## 1.0 CY7C68053 Features

- USB 2.0 – USB-IF High-Speed and Full-Speed Compliant (TID# 40000188)
- Single-chip integrated USB 2.0 transceiver, smart SIE, and enhanced 8051 microprocessor
- Ideal for mobile applications (cell phone, smart phones, PDAs, MP3 players)
  - Ultra low power
  - Suspend current: 20  $\mu$ A (typical)
- Software: 8051 code runs from:
  - Internal RAM, which is loaded from EEPROM
- 16 kBytes of on-chip Code/Data RAM
- Four programmable BULK/INTERRUPT/ISOCRONOUS endpoints
  - Buffering options: double, triple, and quad
- Additional programmable (BULK/INTERRUPT) 64-byte endpoint
- 8- or 16-bit external data interface
- Smart Media Standard ECC generation
- GPIF (General Programmable Interface)
  - Allows direct connection to most parallel interface
  - Programmable waveform descriptors and configuration registers to define waveforms
  - Supports multiple Ready (RDY) inputs and Control (CTL) outputs
- Integrated, industry standard enhanced 8051
  - 48 MHz, 24 MHz, or 12 MHz CPU operation
  - Four clocks per instruction cycle
  - Three counter/timers
  - Expanded interrupt system
  - Two data pointers
- 1.8V core operation
- 1.8V - 3.3V IO operation
- Vectored USB interrupts and GPIF/FIFO interrupts
- Separate data buffers for the Set-up and Data portions of a CONTROL transfer
- Integrated I<sup>2</sup>C™ controller, runs at 100 or 400 kHz
- Four integrated FIFO's
  - Integrated glue logic and FIFO's lower system cost
  - Automatic conversion to and from 16-bit buses
  - Master or slave operation
  - Uses external clock or asynchronous strobes
  - Easy interface to ASIC and DSP IC's
- Available in Industrial temperature grade
- Available in one lead-free package with up to 24 GPIO's
  - 56-pin VFBGA (24 GPIO's)



Cypress Semiconductor Corporation's MoBL-USB™ FX2LP18 (CY7C68053) is a low-voltage (1.8 volt) version of the EZ-USB® FX2LP (CY7C68013A), which is a highly integrated, low-power USB 2.0 microcontroller. By integrating the USB 2.0 transceiver, serial interface engine (SIE), enhanced 8051 microcontroller, and a programmable peripheral interface in a single chip, Cypress has created a very cost-effective solution that provides superior time-to-market advantages with low power to enable bus powered applications.

The ingenious architecture of MoBL-USB FX2LP18 results in data transfer rates of over 53 Mbytes per second, the maximum allowable USB 2.0 bandwidth, while still using a low-cost 8051 microcontroller in a package as small as a 56 VFBGA (5 mm x 5 mm). Because it incorporates the USB 2.0 transceiver, the MoBL-USB FX2LP18 is more economical, providing a smaller footprint solution than USB 2.0 SIE or external transceiver implementations. With MoBL-USB FX2LP18, the Cypress Smart SIE handles most of the USB 1.1 and 2.0 protocol in hardware, freeing the embedded microcontroller for application-specific functions and decreasing development time to ensure USB compatibility.

The General Programmable Interface (GPIF) and Master/Slave Endpoint FIFO (8- or 16-bit data bus) provide an easy and glueless interface to popular interfaces such as ATA, UTOPIA, EPP, PCMCIA, and most DSP/processors.

The 56VFBGA package is defined for the family.

The MoBL-USB FX2LP18 is also referred to as FX2LP18 in this document.

## 2.0 Applications

There are a wide variety of applications for the MoBL-USB FX2LP18. It is used in cell phone, smart phones, PDAs, and MP3 players, to name a few.

The 'Reference Designs' section of the Cypress web site provides additional tools for typical USB 2.0 applications. Each reference design comes complete with firmware source and object code, schematics, and documentation. For more information, visit <http://www.cypress.com>.

## 3.0 Functional Overview

The functionality of this chip is described in the sections below.

### 3.1 USB Signaling Speed

FX2LP18 operates at two of the three rates defined in the USB Specification Revision 2.0, dated April 27, 2000.

- Full-speed, with a signaling bit rate of 12 Mbps
- High-speed, with a signaling bit rate of 480 Mbps.

FX2LP18 does not support the low-speed signaling mode of 1.5 Mbps.

## 3.2 8051 Microprocessor

The 8051 microprocessor embedded in the FX2LP18 family has 256 bytes of register RAM, an expanded interrupt system, and three timer/counters.

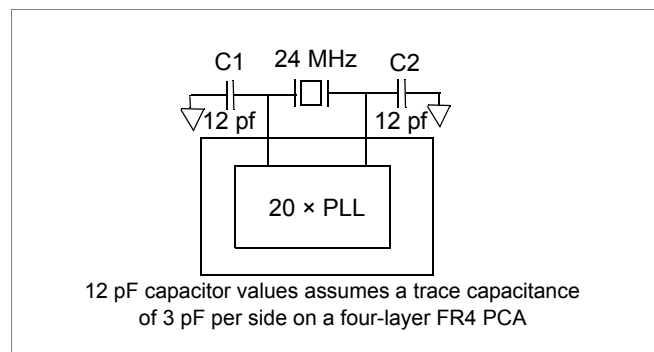
### 3.2.1 8051 Clock Frequency

FX2LP18 has an on-chip oscillator circuit that uses an external 24 MHz ( $\pm 100$ -ppm) crystal with the following characteristics:

- Parallel resonant
- Fundamental mode
- 500  $\mu$ W drive level
- 12 pF (5% tolerance) load capacitors

An on-chip PLL multiplies the 24 MHz oscillator up to 480 MHz, as required by the transceiver/PHY; internal counters divide it down for use as the 8051 clock. The default 8051 clock frequency is 12 MHz. The clock frequency of the 8051 can be changed by the 8051 through the CPUCS register, dynamically.

**Figure 3-1. Crystal Configuration**



The CLKOUT pin, which can be tri-stated and inverted using internal control bits, outputs the 50% duty cycle 8051 clock, at the selected 8051 clock frequency — 48, 24, or 12 MHz.

### 3.2.2 Special Function Registers

Certain 8051 Special Function Register (SFR) addresses are populated to provide fast access to critical FX2LP18 functions. These SFR additions are shown in *Table 3-1*. Bold type indicates non-standard, enhanced 8051 registers. The two SFR rows that end with '0' and '8' contain bit-addressable registers. The four IO ports A – D use the SFR addresses used in the standard 8051 for ports 0 – 3, which are not implemented in FX2LP18. Because of the faster and more efficient SFR addressing, the FX2LP18 IO ports are not addressable in external RAM space (using the MOVX instruction).

**Table 3-1. Special Function Registers**

x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx
0	IOA	IOB	IOC	IOD	SCON1	PSW	ACC	B
1	SP	EXIF	INT2CLR	IOE	SBUF1			
2	DPL0	MPAGE		OEA				
3	DPH0			OEB				
4	DPL1			OEC				
5	DPH1			OED				
6	DPS			OEE				
7	PCON							
8	TCON	SCON0	IE	IP	T2CON	EICON	EIE	EIP
9	TMOD	SBUF0						
A	TL0	AUTOPTRH1	EP2468STAT	EP01STAT	RCAP2L			
B	TL1	AUTOPTRL1	EP24FIFOFLGS	GPIFTRIG	RCAP2H			
C	TH0	Reserved	EP68FIFOFLGS		TL2			
D	TH1	AUTOPTRH2		GPIFSGLDATH	TH2			
E	CKCON	AUTOPTRL2		GPIFSGLDATLX				
F		Reserved	AUTOPTRSET-UP	GPIFSGLDATLNOX				

### 3.3 I<sup>2</sup>C™ Bus

FX2LP18 supports the I<sup>2</sup>C bus as a master only at 100-/400-KHz. SCL and SDA pins have open-drain outputs and hysteresis inputs. These signals must be pulled up to either V<sub>CC</sub> or V<sub>CC\_IO</sub>, even if no I<sup>2</sup>C device is connected. (Connecting to V<sub>CC\_IO</sub> may be more convenient.)

### 3.4 Buses

This 56-pin package has an 8- or 16-bit ‘FIFO’ bidirectional data bus, multiplexed on IO ports B and D.

### 3.5 USB Boot Methods

During the power-up sequence, internal logic checks the I<sup>2</sup>C port for the connection of an EEPROM whose first byte is 0xC2. If found, it boot-loads the EEPROM contents into internal RAM (0xC2 load). If no EEPROM is present, an external processor must emulate an I<sup>2</sup>C slave. The FX2LP18 does not enumerate using internally stored descriptors (for example, Cypress’ VID/PID/DID is not used for enumeration).<sup>[1]</sup>

### 3.6 ReNumeration™

Because the FX2LP18’s configuration is soft, one chip can take on the identities of multiple distinct USB devices.

When first plugged into USB, the FX2LP18 enumerates automatically and downloads firmware and USB descriptor tables over the USB cable. Next, the FX2LP18 enumerates again, this time as a device defined by the downloaded information. This patented two-step process, called ReNumeration™, happens instantly when the device is

plugged in, with no hint that the initial download step has occurred.

Two control bits in the USBCS (USB Control and Status) register control the ReNumeration process: DISCON and RENUM. To simulate a USB disconnect, the firmware sets DISCON to 1. To reconnect, the firmware clears DISCON to 0.

Before reconnecting, the firmware sets or clears the RENUM bit to indicate whether the firmware or the Default USB Device handles device requests over endpoint zero: if RENUM = 0, the Default USB Device handles device requests; if RENUM = 1, the firmware does.

### 3.7 Bus-powered Applications

The FX2LP18 fully supports bus-powered designs by enumerating with less than 100 mA as required by the USB 2.0 specification.

### 3.8 Interrupt System

The FX2LP18 interrupts are described in this section.

#### 3.8.1 INT2 Interrupt Request and Enable Registers

FX2LP18 implements an autovector feature for INT2. There are 27 INT2 (USB) vectors. See the *MoBL-USB™ Technical Reference Manual (TRM)* for more details.

#### 3.8.2 USB Interrupt Autovectors

The main USB interrupt is shared by 27 interrupt sources. To save the code and processing time that is normally required to identify the individual USB interrupt source, the FX2LP18 provides a second level of interrupt vectoring, called ‘Autovectoring.’ When a USB interrupt is asserted, the FX2LP18

#### Note

1. The I<sup>2</sup>C bus SCL and SDA pins must be pulled up, even if an EEPROM is not connected. Otherwise this detection method does not work properly.

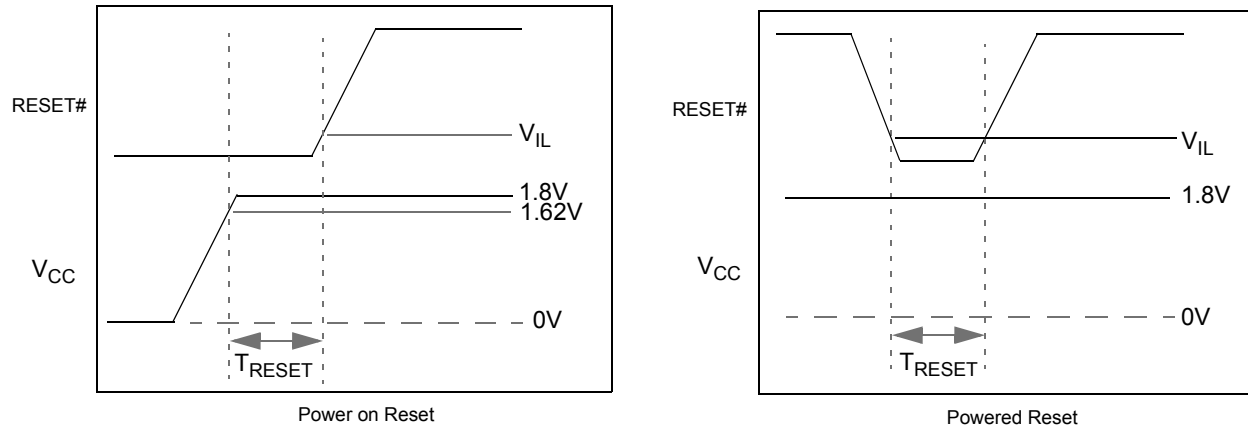
pushes the program counter onto its stack then jumps to address 0x0043, where it expects to find a 'jump' instruction to the USB interrupt service routine.

The FX2LP18 jump instruction is encoded as shown in *Table 3-2*.

If Autovectoring is enabled (AV2EN = 1 in the INTSET-UP register), the FX2LP18 substitutes its INT2VEC byte. Therefore, if the high byte ('page') of a jump-table address is preloaded at location 0x0044, the automatically-inserted INT2VEC byte at 0x0045 directs the jump to the correct address out of the 27 addresses within the page.

**Table 3-2. INT2 USB Interrupts**

USB INTERRUPT TABLE FOR INT2			
Priority	INT2VEC Value	Source	Notes
1	00	SUDAV	Set-up Data Available
2	04	SOF	Start of Frame (or microframe)
3	08	SUTOK	Set-up Token Received
4	0C	SUSPEND	USB Suspend request
5	10	USB RESET	Bus reset
6	14	HISPEED	Entered high-speed operation
7	18	EP0ACK	FX2LP18 ACK'd the CONTROL Handshake
8	1C		Reserved
9	20	EP0-IN	EP0-IN ready to be loaded with data
10	24	EP0-OUT	EP0-OUT has USB data
11	28	EP1-IN	EP1-IN ready to be loaded with data
12	2C	EP1-OUT	EP1-OUT has USB data
13	30	EP2	IN: buffer available. OUT: buffer has data
14	34	EP4	IN: buffer available. OUT: buffer has data
15	38	EP6	IN: buffer available. OUT: buffer has data
16	3C	EP8	IN: buffer available. OUT: buffer has data
17	40	IBN	IN-Bulk-NAK (any IN endpoint)
18	44		Reserved
19	48	EP0PING	EP0 OUT was Pinged and it NAK'd
20	4C	EP1PING	EP1 OUT was Pinged and it NAK'd
21	50	EP2PING	EP2 OUT was Pinged and it NAK'd
22	54	EP4PING	EP4 OUT was Pinged and it NAK'd
23	58	EP6PING	EP6 OUT was Pinged and it NAK'd
24	5C	EP8PING	EP8 OUT was Pinged and it NAK'd
25	60	ERRLIMIT	Bus errors exceeded the programmed limit
26	64		
27	68		Reserved
28	6C		Reserved
29	70	EP2ISOERR	ISO EP2 OUT PID sequence error
30	74	EP4ISOERR	ISO EP4 OUT PID sequence error
31	78	EP6ISOERR	ISO EP6 OUT PID sequence error
32	7C	EP8ISOERR	ISO EP8 OUT PID sequence error

**Figure 3-2. Reset Timing Plots**


### 3.9 Reset and Wakeup

The reset and wakeup pins are described in detail in this section.

#### 3.9.1 Reset Pin

The input pin, RESET#, resets the FX2LP18 when asserted. This pin has hysteresis and is active LOW. When a crystal is used with the CY7C68053, the reset period must allow for the stabilization of the crystal and the PLL. This reset period must be approximately 5 ms after VCC has reached 3.0V. If the crystal input pin is driven by a clock signal the internal PLL stabilizes in 200  $\mu$ s after VCC has reached 3.0V<sup>[2]</sup>. Figure 3-2 shows a power on reset condition and a reset applied during operation. A power on reset is defined as the time reset is asserted while power is being applied to the circuit. A powered reset is defined to be when the FX2LP18 has previously been powered on and operating and the RESET# pin is asserted.

Cypress provides an application note which describes and recommends power on reset implementation and can be found on the Cypress web site. For more information on reset implementation for the MoBL-USB™ family of products, visit the Cypress web site at <http://www.cypress.com>.

**Table 3-3. Reset Timing Values**

Condition	T <sub>RESET</sub>
Power on Reset with crystal	5 ms
Power on Reset with external clock	200 $\mu$ s + Clock stability time
Powered Reset	200 $\mu$ s

#### 3.9.2 Wakeup Pins

The 8051 puts itself and the rest of the chip into a power-down mode by setting PCON.0 = 1. This stops the oscillator and PLL. When WAKEUP is asserted by external logic, the oscillator restarts, after the PLL stabilizes, and then the 8051 receives a wakeup interrupt. This applies whether or not FX2LP18 is connected to the USB.

#### Note

- If the external clock is powered at the same time as the CY7C680xx and has a stabilization wait period, it must be added to the 200  $\mu$ s.

The FX2LP18 exits the power-down (USB suspend) state using one of the following methods:

- USB bus activity (if D+/D- lines are left floating, noise on these lines may indicate activity to the FX2LP18 and initiate a wakeup)
- External logic asserts the WAKEUP pin
- External logic asserts the PA3/WU2 pin

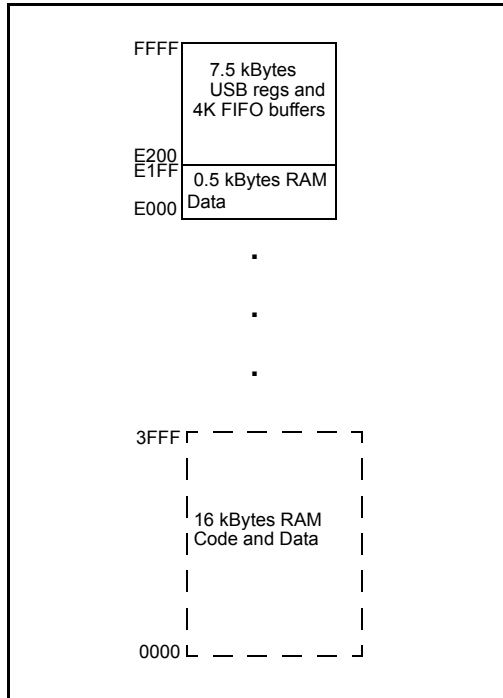
The second wakeup pin, WU2, can also be configured as a general purpose IO pin. This allows a simple external R-C network to be used as a periodic wakeup source. Note that WAKEUP is by default active LOW.

#### 3.9.3 Lowering Suspend Current

Good design practices for CMOS circuits dictate that any unused input pins must not be floating between V<sub>IL</sub> and V<sub>IH</sub>. Floating input pins will not damage the chip, but can substantially increase suspend current. To achieve the lowest suspend current, any unused port pins must be configured as outputs. Any unused input pins must be tied to ground. Some examples of pins that need attention during suspend are:

- Port pins. For Port A, B, D pins, extra care must be taken in shared bus situations.
  - Completely unused pins must be pulled to V<sub>CC\_IO</sub> or GND.
  - In a single-master system, the firmware must output enable all the port pins and drive them high or low, before FX2LP18 enters the suspend state.
  - In a multi-master system (FX2LP18 and another processor sharing a common data bus), when FX2LP18 is suspended, the external master must drive the pins high or low. The external master may not let the pins float.
- CLKOUT. If CLKOUT is not used, it must be tri-stated during normal operation, but driven during suspend.
- IFCLK, RDY0, RDY1. These pins must be pulled to V<sub>CC\_IO</sub> or GND or driven by another chip.
- CTL0-2. If tri-stated via GPIFIDLECTL, these pins must be pulled to V<sub>CC\_IO</sub> or GND or driven by another chip.
- RESET#, WAKEUP#. These pins must be pulled to V<sub>CC\_IO</sub> or GND or driven by another chip during suspend.

Figure 3-3. FX2LP18 Internal Code Memory



### 3.10 Program/Data RAM

This section describes the FX2LP18 RAM.

#### 3.10.1 Size

The FX2LP18 has 16 kBytes of internal program/data RAM. No USB control registers appear in this space.

Memory maps are shown in *Figure 3-3* and *Figure 3-4*.

#### 3.10.2 Internal Code Memory

This mode implements the internal 16-kByte block of RAM (starting at 0) as combined code and data memory. Only the **internal** 16 kBytes and **scratch pad** 0.5 kBytes RAM spaces have the following access:

- USB download
- USB upload
- Set-up data pointer
- I<sup>2</sup>C interface boot load

### 3.11 Register Addresses

Figure 3-4. Register Address Memory

FFFF	4 kBytes EP2-EP8 buffers (8 x 512)
F000	2 kBytes RESERVED
EFFF	
E800	64 Bytes EP1IN
E7FF	
E7C0	64 Bytes EP1OUT
E7BF	
E780	64 Bytes EP0 IN/OUT
E77F	
E740	64 Bytes RESERVED
E73F	
E700	8051 Addressable Registers (512)
E6FF	
E500	Reserved (128)
E4FF	
E480	128 Bytes GPIF Waveforms
E47F	
E400	Reserved (512)
E3FF	
E200	512 Bytes 8051 xdata RAM
E1FF	
E000	

### 3.12 Endpoint RAM

This section describes the FX2LP18 Endpoint RAM.

#### 3.12.1 Size

- 3 × 64 bytes (Endpoints 0, 1)
- 8 × 512 bytes (Endpoints 2, 4, 6, 8)

#### 3.12.2 Organization

- EP0
- Bidirectional endpoint zero, 64-byte buffer
- EP1IN, EP1OUT
- 64-byte buffers: bulk or interrupt
- EP2, 4, 6, 8
- Eight 512-byte buffers: bulk, interrupt, or isochronous. EP4 and EP8 can be double buffered, while EP2 and 6 can be double, triple, or quad buffered. For high-speed endpoint configuration options, see *Figure 3-5*.

#### 3.12.3 Set-up Data Buffer

A separate 8-byte buffer at 0xE6B8-0xE6BF holds the set-up data from a CONTROL transfer.

#### 3.12.4 Endpoint Configurations (High-speed Mode)

Endpoints 0 and 1 are the same for every configuration. Endpoint 0 is the only CONTROL endpoint, and endpoint 1 can be either BULK or INTERRUPT. The endpoint buffers can be configured in any one of the 12 configurations shown in the

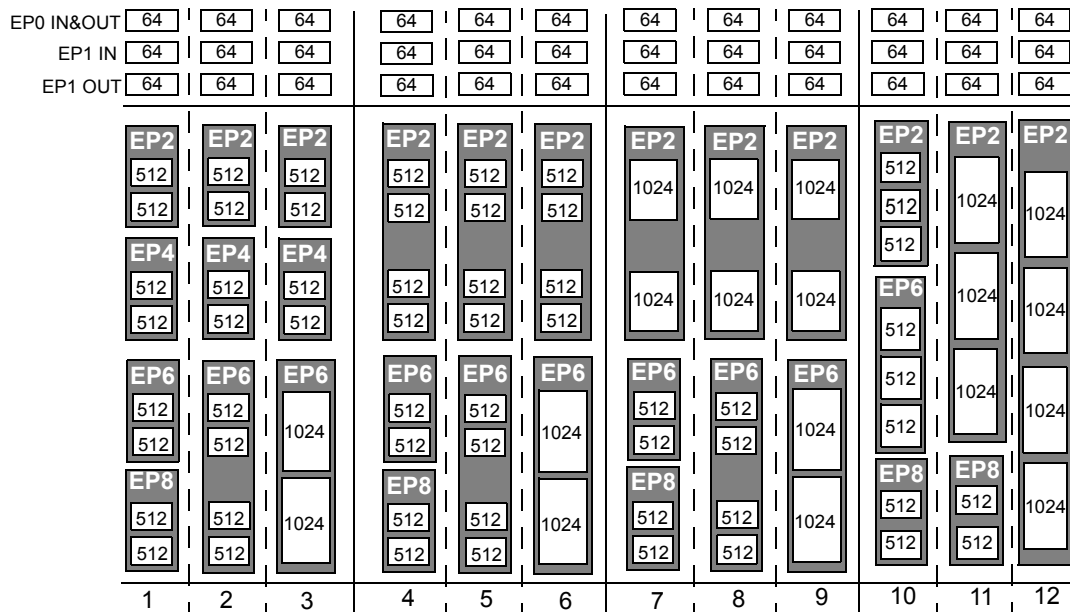


vertical columns of *Figure 3-5*. When operating in full-speed BULK mode only the first 64 bytes of each buffer are used. For example, in high-speed the maximum packet size is 512 bytes, but in full-speed it is 64 bytes. Even though a buffer is configured to be a 512 byte buffer, in full-speed only the first

64 bytes are used. The unused endpoint buffer space is not available for other operations. An example endpoint configuration is:

EP2–1024 double buffered; EP6–512 quad buffered (column 8).

**Figure 3-5. Endpoint Configuration**



3.12.5 Default Full-Speed Alternate Settings

**Table 3-4. Default Full-Speed Alternate Settings<sup>[3, 4]</sup>**

Alternate Setting	0	1	2	3
ep0	64	64	64	64
ep1out	0	64 bulk	64 int	64 int
ep1in	0	64 bulk	64 int	64 int
ep2	0	64 bulk out (2×)	64 int out (2×)	64 iso out (2×)
ep4	0	64 bulk out (2×)	64 bulk out (2×)	64 bulk out (2×)
ep6	0	64 bulk in (2×)	64 int in (2×)	64 iso in (2×)
ep8	0	64 bulk in (2×)	64 bulk in (2×)	64 bulk in (2×)

**Notes**

- 3. '0' means 'not implemented.'
- 4. '2×' means 'double buffered.'

### 3.12.6 Default High-Speed Alternate Settings

**Table 3-5. Default High-Speed Alternate Settings<sup>[3, 4]</sup>**

Alternate Setting	0	1	2	3
ep0	64	64	64	64
ep1out	0	512 bulk <sup>[5]</sup>	64 int	64 int
ep1in	0	512 bulk <sup>[5]</sup>	64 int	64 int
ep2	0	512 bulk out (2×)	512 int out (2×)	512 iso out (2×)
ep4	0	512 bulk out (2×)	512 bulk out (2×)	512 bulk out (2×)
ep6	0	512 bulk in (2×)	512 int in (2×)	512 iso in (2×)
ep8	0	512 bulk in (2×)	512 bulk in (2×)	512 bulk in (2×)

### 3.13 External FIFO Interface

The architecture, control signals, and clock rates are presented in this section.

#### 3.13.1 Architecture

The FX2LP18 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 (such as IFCLK, SLCS#, SLRD, SLWR, SLOE, PKTEND, and flags).

In operation, some of the eight RAM blocks fill or empty from the SIE while the others are connected to the IO transfer logic. The transfer logic takes two forms: the GPIF for internally generated control signals or the slave FIFO interface for externally controlled transfers.

#### 3.13.2 Master/Slave Control Signals

The FX2LP18 endpoint FIFO's are implemented as eight physically distinct 256x16 RAM blocks. The 8051/SIE can switch any of the RAM blocks between two domains, the USB (SIE) domain and the 8051-IO Unit domain. This switching is instantaneous, giving zero transfer time between 'USB FIFO's' and 'Slave FIFO's.' Since they are physically the same memory, no bytes are actually transferred between buffers.

At any given time, some RAM blocks are filling and emptying with USB data under SIE control, while other RAM blocks are available to the 8051 and/or the IO control unit. The RAM blocks operate as single port in the USB domain, and dual port in the 8051-IO domain. The blocks can be configured as single, double, triple, or quad buffered as previously shown.

The IO control unit implements either an internal master (M for master) or external master (S for Slave) interface.

In Master (M) mode, the GPIF internally controls FIFOADR[1:0] to select a FIFO. The two RDY pins can be used as flag inputs from an external FIFO or other logic. The GPIF can be run from either an internally derived clock or externally supplied clock (IFCLK), at a rate that transfers data up to 96 Megabytes/s (48 MHz IFCLK with 16-bit interface).

In Slave (S) mode, the FX2LP18 accepts either an internally derived clock or externally supplied clock (IFCLK, maximum frequency 48 MHz) and SLCS#, SLRD, SLWR, SLOE, PKTEND signals from external logic. When using an external IFCLK, the external clock must be present before switching to the external clock with the IFCLKSRC bit. Each endpoint can individually be selected for byte or word operation by an internal configuration bit, and a Slave FIFO Output Enable signal (SLOE) enables data of the selected width. External logic must insure that the output enable signal is inactive when writing data to a slave FIFO. The slave interface can also operate asynchronously, where the SLRD and SLWR signals act directly as strobes, rather than a clock qualifier as in synchronous mode. The signals SLRD, SLWR, SLOE and PKTEND are gated by the signal SLCS#.

#### 3.13.3 GPIF and FIFO Clock Rates

An 8051 register bit selects one of two frequencies for the internally supplied interface clock: 30 MHz and 48 MHz. Alternatively, an externally supplied clock of 5 MHz – 48 MHz feeding the IFCLK pin can be used as the interface clock. IFCLK can be configured to function as an output clock when the GPIF and FIFO's are internally clocked. An output enable bit in the IFCONFIG register turns this clock output off. Another bit within the IFCONFIG register will invert the IFCLK signal whether internally or externally sourced.

### 3.14 GPIF

The GPIF is a flexible 8- or 16-bit parallel interface driven by a user programmable finite state machine. It allows the CY7C68053 to perform local bus mastering, and can implement a wide variety of protocols such as ATA interface, parallel printer port, and Utopia.

The GPIF has three programmable control outputs (CTL), and two general purpose ready inputs (RDY). The data bus width can be 8 or 16 bits. Each GPIF vector defines the state of the control outputs, and determines what state a ready input (or multiple inputs) must be before proceeding. The GPIF vector can be programmed to advance a FIFO to the next data value, advance an address, and so on. A sequence of the GPIF vectors make up a single waveform that is executed to perform the desired data move between the FX2LP18 and the external device.

#### Notes

5. Even though these buffers are 64 bytes, they are reported as 512 for USB 2.0 compliance. The user must never transfer packets larger than 64 bytes to EP1.



### 3.14.1 Three Control OUT Signals

The 56-pin package brings out three of these signals, CTL0–CTL2. The 8051 programs the GPIF unit to define the CTL waveforms. CTLx waveform edges can be programmed to make transitions as fast as once per clock cycle (20.8 ns using a 48 MHz clock).

### 3.14.2 Two Ready IN Signals

The FX2LP18 package brings out all two Ready inputs (RDY0–RDY1). The 8051 programs the GPIF unit to test the RDY pins for GPIF branching.

### 3.14.3 Long Transfer Mode

In master mode, the 8051 appropriately sets GPIF transaction count registers (GPIFTCB3, GPIFTCB2, GPIFTCB1, or GPIFTCB0) for unattended transfers of up to  $2^{32}$  transactions. The GPIF automatically throttles data flow to prevent under or overflow until the full number of requested transactions complete. The GPIF decrements the value in these registers to represent the current status of the transaction.

## 3.15 ECC Generation<sup>[6]</sup>

The MoBL-USB can calculate Error Correcting Codes (ECC's) on data that passes across its GPIF or Slave FIFO interfaces. There are two ECC configurations: two ECC's, each calculated over 256 bytes (SmartMedia Standard) and one ECC calculated over 512 bytes.

The ECC can correct any 1-bit error or detect any 2-bit error.

### 3.15.1 ECC Implementation

The two ECC configurations are selected by the ECCM bit.

#### 3.15.1.1 ECCM = 0

Two 3-byte ECC's are each calculated over a 256-byte block of data. This configuration conforms to the SmartMedia Standard.

This configuration writes any value to ECCRESET, then passes data across the GPIF or Slave FIFO interface. The ECC for the first 256 bytes of data is calculated and stored in ECC1. The ECC for the next 256 bytes is stored in ECC2. After the second ECC is calculated, the values in the ECCx registers do not change until ECCRESET is written again, even if more data is subsequently passed across the interface.

#### 3.15.1.2 ECCM = 1

One 3-byte ECC is calculated over a 512-byte block of data.

This configuration writes any value to ECCRESET then passes data across the GPIF or Slave FIFO interface. The ECC for the first 512 bytes of data is calculated and stored in ECC1; ECC2 is unused. After the ECC is calculated, the value in ECC1 does not change until ECCRESET is written again, even if more data is subsequently passed across the interface.

## 3.16 USB Uploads and Downloads

The core has the ability to directly edit the data contents of the internal 16-kByte RAM and of the internal 512-byte scratch pad RAM via a vendor-specific command. This capability is normally used when 'soft' downloading user code and is available only to and from internal RAM, only when the 8051 is held in reset. The available RAM spaces are 16 kBytes from 0x0000–0x3FFF (code/data) and 512 bytes from 0xE000–0xE1FF (scratch pad data RAM).<sup>[7]</sup>

## 3.17 Autopointer Access

FX2LP18 provides two identical autopointers. They are similar to the internal 8051 data pointers, but with an additional feature: they can optionally increment after every memory access. The autopointers are available in external FX2LP18 registers, under control of a mode bit (AUTOPTSET-UP.0). Using the external FX2LP18 autopointer access (at 0xE67B – 0xE67C) allows the autopointer to access all RAM. Also, the autopointers can point to any FX2LP18 register or endpoint buffer space.

## 3.18 I<sup>2</sup>C Controller

FX2LP18 has one I<sup>2</sup>C port that is driven by two internal controllers. One automatically operates at boot time to load the VID/PID/DID, configuration byte, and firmware and a second controller that the 8051, once running, uses to control external I<sup>2</sup>C devices. The I<sup>2</sup>C port operates in master mode only.

### 3.18.1 I<sup>2</sup>C Port Pins

The I<sup>2</sup>C pins SCL and SDA must have external 2.2K ohm pull up resistors even if no EEPROM is connected to the FX2LP18. The value of the pull up resistors required may vary, depending on the combination of  $V_{CC\_IO}$  and the supply used for the EEPROM. The pull up resistors used must be such that when the EEPROM pulls SDA low, the voltage level meets the  $V_{IL}$  specification of the FX2LP18. For example, if the EEPROM runs off a 3.3V supply and  $V_{CC\_IO}$  is 1.8V, the pull up resistors recommended are 10K ohm. This requirement may also vary depending on the devices being run on the I<sup>2</sup>C pins. Refer to the I<sup>2</sup>C specifications for details.

External EEPROM device address pins must be configured properly. See *Table 3-6* for configuring the device address pins.

If no EEPROM is connected to the I<sup>2</sup>C port, EEPROM emulation is required by an external processor. This is because the FX2LP18 comes out of reset with the DISCON bit set, so the device will not enumerate without an EEPROM (C2 load) or EEPROM emulation.

## Notes

6. To use the ECC logic, the GPIF or Slave FIFO interface must be configured for byte-wide operation.
7. After the data has been downloaded from the host, a 'loader' can execute from internal RAM in order to transfer downloaded data to external memory.

**Table 3-6. Strap Boot EEPROM Address Lines to These Values**

Bytes	Example EEPROM	A2	A1	A0
16	24AA00 <sup>[8]</sup>	N/A	N/A	N/A
128	24AA01	0	0	0
256	24AA02	0	0	0
4K	24AA32	0	0	1
8K	24AA64	0	0	1
16K	24AA128	0	0	1

**3.18.2 I<sup>2</sup>C Interface Boot Load Access**

At power on reset the I<sup>2</sup>C interface boot loader loads the VID/PID/DID and configuration bytes and up to 16 kBytes of program/data. The available RAM spaces are 16 kBytes from

0x0000–0x3FFF and 512 bytes from 0xE000–0xE1FF. The 8051 is reset. I<sup>2</sup>C interface boot loads only occur after power on reset.

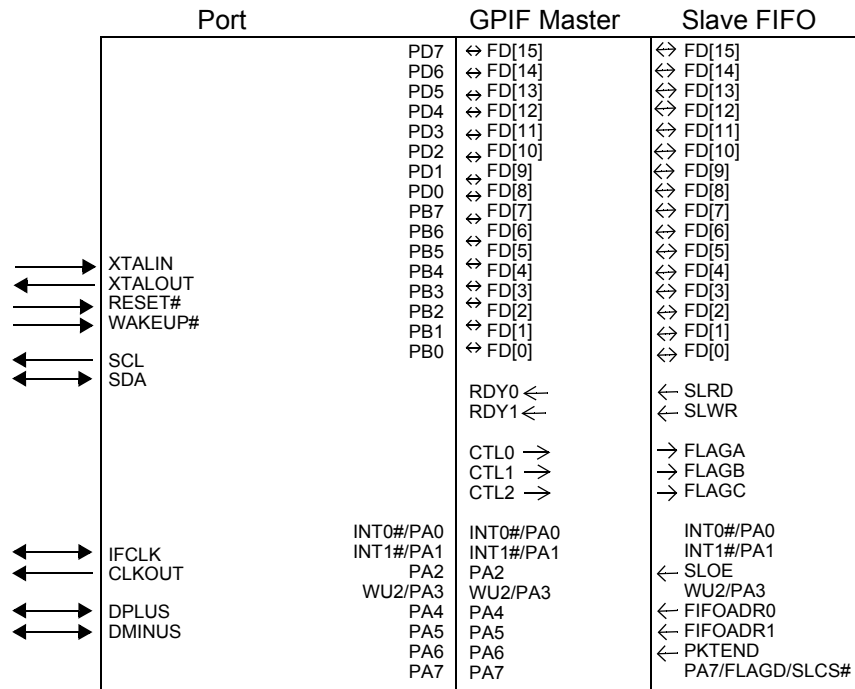
**3.18.3 I<sup>2</sup>C Interface General Purpose Access**

The 8051 can control peripherals connected to the I<sup>2</sup>C bus using the I2CTL and I2DAT registers. FX2LP18 provides I<sup>2</sup>C master control only, it is never an I<sup>2</sup>C slave.

**4.0 Pin Assignments**

Figure 4-1 identifies all signals for the package. It is followed by the pin diagram. Three modes are available: Port, GPIF master, and Slave FIFO. These modes define the signals on the right edge of the diagram. The 8051 selects the interface mode using the IFCONFIG[1:0] register bits. Port mode is the power on default configuration.

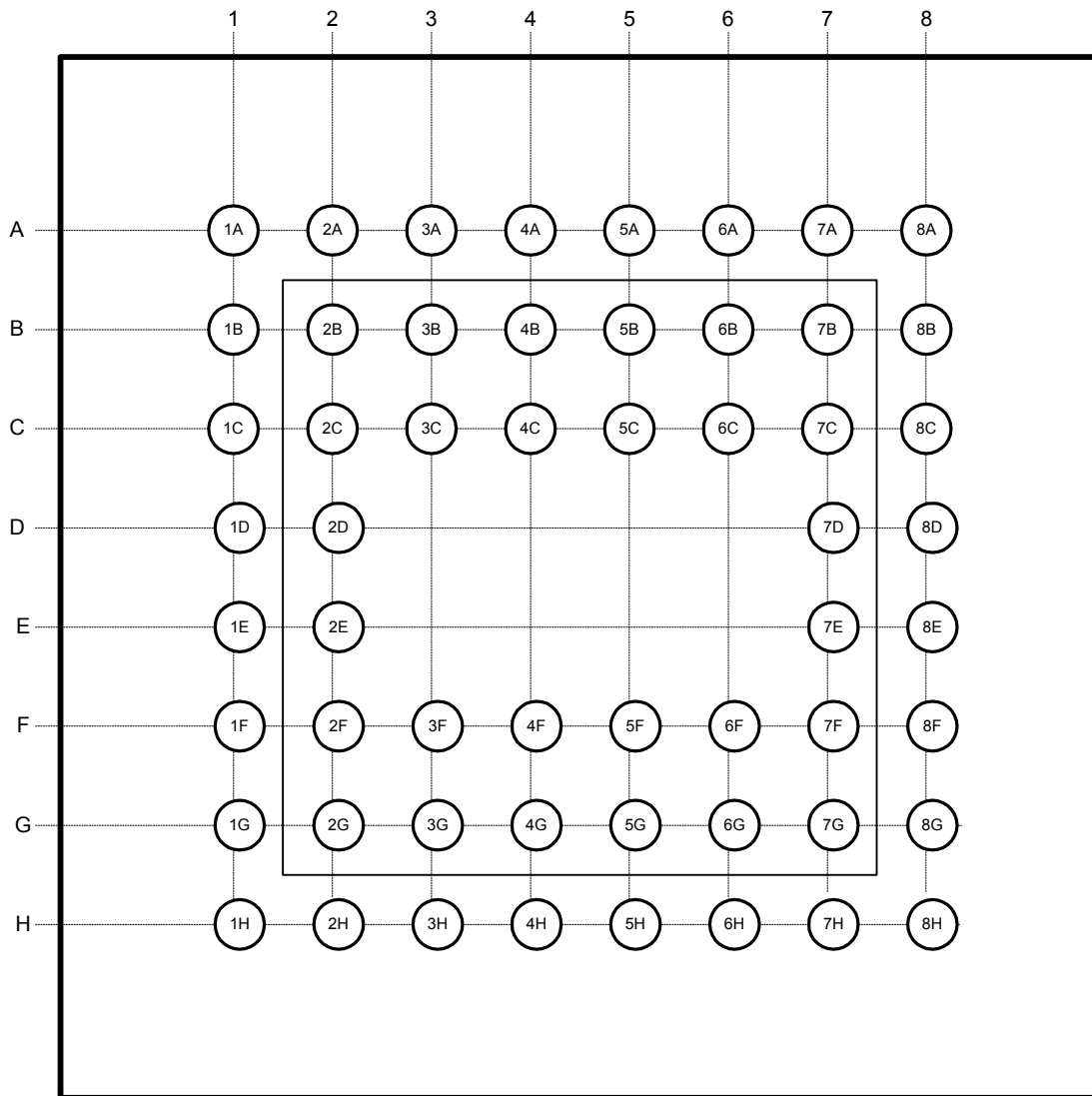
**Figure 4-1. Signals**



**Note**

8. This EEPROM does not have address pins.

**Figure 4-2. CY7C68053 56-pin VFBGA Pin Assignment - Top view**



**4.1 CY7C68053 Pin Descriptions**
**Table 4-1. FX2LP18 Pin Descriptions** <sup>[9]</sup>

56 VFBGA	Name	Type	Default	Description
2D	AV <sub>CC</sub>	Power	N/A	<b>Analog VCC.</b> Connect this pin to 3.3V power source. This signal provides power to the analog section of the chip. Appropriate bulk/bypass capacitance should be provided for this supply rail.
1D	AV <sub>CC</sub>	Power	N/A	<b>Analog VCC.</b> Connect this pin to 3.3V power source. This signal provides power to the analog section of the chip.
2F	AGND	Ground	N/A	<b>Analog Ground.</b> Connect this pin to ground with as short a path as possible.
1F	AGND	Ground	N/A	<b>Analog Ground.</b> Connect to this pin ground with as short a path as possible.
1E	DMINUS	I/O/Z	Z	<b>USB D– Signal.</b> Connect this pin to the USB D– signal.
2E	DPLUS	I/O/Z	Z	<b>USB D+ Signal.</b> Connect this pin to the USB D+ signal.
8B	RESET#	Input	N/A	<b>Active LOW Reset.</b> This pin resets the entire chip. See Section 3.9 "Reset and Wakeup" on page 5 for more details.
1C	XTALIN	Input	N/A	<b>Crystal Input.</b> Connect this signal to a 24 MHz parallel resonant, fundamental mode crystal and load capacitor to GND. It is also correct to drive XTALIN with an external 24-MHz square wave derived from another clock source.
2C	XTALOUT	Output	N/A	<b>Crystal Output.</b> Connect this signal to a 24 MHz parallel resonant, fundamental mode crystal and load capacitor to GND. If an external clock is used to drive XTALIN, leave this pin open.
2B	CLKOUT	O/Z	12 MHz	<b>CLKOUT.</b> 12-, 24- or 48-MHz clock, phase locked to the 24 MHz input clock. The 8051 defaults to 12 MHz operation. The 8051 may tri-state this output by setting CPUCS.1 = 1.
<b>Port A</b>				
8G	PA0 or INT0#	I/O/Z	I (PA0)	Multiplexed pin whose function is selected by PORTACFG.0 <b>PA0</b> is a bidirectional IO port pin. <b>INT0#</b> is the active LOW 8051 INT0 interrupt input signal, which is either edge triggered (IT0 = 1) or level triggered (IT0 = 0).
6G	PA1 or INT1#	I/O/Z	I (PA1)	Multiplexed pin whose function is selected by: PORTACFG.1 <b>PA1</b> is a bidirectional IO port pin. <b>INT1#</b> is the active LOW 8051 INT1 interrupt input signal, which is either edge triggered (IT1 = 1) or level triggered (IT1 = 0).
8F	PA2 or SLOE	I/O/Z	I (PA2)	Multiplexed pin whose function is selected by two bits: IFCONFIG[1:0]. <b>PA2</b> is a bidirectional IO port pin. <b>SLOE</b> is an input-only output enable with programmable polarity (FIFOPINPOLAR.4) for the slave FIFO's connected to FD[7:0] or FD[15:0].
7F	PA3 or WU2	I/O/Z	I (PA3)	Multiplexed pin whose function is selected by: WAKEUP.7 and OEA.3 <b>PA3</b> is a bidirectional IO port pin. <b>WU2</b> is an alternate source for <b>USB Wakeup</b> , enabled by WU2EN bit (WAKEUP.1) and polarity set by WU2POL (WAKEUP.4). If the 8051 is in suspend and WU2EN = 1, a transition on this pin starts up the oscillator and interrupts the 8051 to allow it to exit the suspend mode. Asserting this pin inhibits the chip from suspending, if WU2EN = 1.

**Note**

9. Unused inputs must not be left floating. Tie either HIGH or LOW as appropriate. Outputs should only be pulled up or down to ensure signals at power up and in standby. Note also that no pins should be driven while the device is powered down

**Table 4-1. FX2LP18 Pin Descriptions (continued)<sup>[9]</sup>**

56 VFBGA	Name	Type	Default	Description
6F	PA4 or FIFOADR0	I/O/Z	I (PA4)	Multiplexed pin whose function is selected by: IFCONFIG[1:0]. <b>PA4</b> is a bidirectional IO port pin. <b>FIFOADR0</b> is an input-only address select for the slave FIFO's connected to FD[7:0] or FD[15:0].
8C	PA5 or FIFOADR1	I/O/Z	I (PA5)	Multiplexed pin whose function is selected by: IFCONFIG[1:0]. <b>PA5</b> is a bidirectional IO port pin. <b>FIFOADR1</b> is an input-only address select for the slave FIFO's connected to FD[7:0] or FD[15:0].
7C	PA6 or PKTEND	I/O/Z	I (PA6)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] bits. <b>PA6</b> is a bidirectional IO port pin. <b>PKTEND</b> is an input used to commit the FIFO packet data to the endpoint and whose polarity is programmable using FIFOPINPOLAR.5.
6C	PA7 or FLAGD or SLCS#	I/O/Z	I (PA7)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and PORTACFG.7 bits. <b>PA7</b> is a bidirectional IO port pin. <b>FLAGD</b> is a programmable slave FIFO output status flag signal. <b>SLCS#</b> gates all other slave FIFO enable/strobes
<b>Port B</b>				
3H	PB0 or FD[0]	I/O/Z	I (PB0)	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>PB0</b> is a bidirectional IO port pin. <b>FD[0]</b> is the bidirectional FIFO/GPIF data bus.
4F	PB1 or FD[1]	I/O/Z	I (PB1)	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>PB1</b> is a bidirectional IO port pin. <b>FD[1]</b> is the bidirectional FIFO/GPIF data bus.
4H	PB2 or FD[2]	I/O/Z	I (PB2)	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>PB2</b> is a bidirectional IO port pin. <b>FD[2]</b> is the bidirectional FIFO/GPIF data bus.
4G	PB3 or FD[3]	I/O/Z	I (PB3)	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>PB3</b> is a bidirectional IO port pin. <b>FD[3]</b> is the bidirectional FIFO/GPIF data bus.
5H	PB4 or FD[4]	I/O/Z	I (PB4)	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>PB4</b> is a bidirectional IO port pin. <b>FD[4]</b> is the bidirectional FIFO/GPIF data bus.
5G	PB5 or FD[5]	I/O/Z	I (PB5)	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>PB5</b> is a bidirectional IO port pin. <b>FD[5]</b> is the bidirectional FIFO/GPIF data bus.
5F	PB6 or FD[6]	I/O/Z	I (PB6)	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>PB6</b> is a bidirectional IO port pin. <b>FD[6]</b> is the bidirectional FIFO/GPIF data bus.
6H	PB7 or FD[7]	I/O/Z	I (PB7)	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>PB7</b> is a bidirectional IO port pin. <b>FD[7]</b> is the bidirectional FIFO/GPIF data bus.

**Table 4-1. FX2LP18 Pin Descriptions (continued)<sup>[9]</sup>**

56 VFBGA	Name	Type	Default	Description
<b>PORT D</b>				
8A	PD0 or FD[8]	I/O/Z	I (PD0)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. <b>FD[8]</b> is the bidirectional FIFO/GPIF data bus.
7A	PD1 or FD[9]	I/O/Z	I (PD1)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. <b>FD[9]</b> is the bidirectional FIFO/GPIF data bus.
6B	PD2 or FD[10]	I/O/Z	I (PD2)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. <b>FD[10]</b> is the bidirectional FIFO/GPIF data bus.
6A	PD3 or FD[11]	I/O/Z	I (PD3)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. <b>FD[11]</b> is the bidirectional FIFO/GPIF data bus.
3B	PD4 or FD[12]	I/O/Z	I (PD4)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. <b>FD[12]</b> is the bidirectional FIFO/GPIF data bus.
3A	PD5 or FD[13]	I/O/Z	I (PD5)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. <b>FD[13]</b> is the bidirectional FIFO/GPIF data bus.
3C	PD6 or FD[14]	I/O/Z	I (PD6)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. <b>FD[14]</b> is the bidirectional FIFO/GPIF data bus.
2A	PD7 or FD[15]	I/O/Z	I (PD7)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. <b>FD[15]</b> is the bidirectional FIFO/GPIF data bus.
1A	RDY0 or SLRD	Input	N/A	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>RDY0</b> is a GPIF input signal. <b>SLRD</b> is the input only read strobe with programmable polarity (FIFOPIN-POLAR.3) for the slave FIFO's connected to FD[7:0] or FD[15:0].
1B	RDY1 or SLWR	Input	N/A	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>RDY1</b> is a GPIF input signal. <b>SLWR</b> is the input only write strobe with programmable polarity (FIFOPIN-POLAR.2) for the slave FIFO's connected to FD[7:0] or FD[15:0].
7H	CTL0 or FLAGA	O/Z	H	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>CTL0</b> is a GPIF control output. <b>FLAGA</b> is a programmable slave FIFO output status flag signal. Defaults to programmable for the FIFO selected by the FIFOADR[1:0] pins.
7G	CTL1 or FLAGB	O/Z	H	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>CTL1</b> is a GPIF control output. <b>FLAGB</b> is a programmable slave FIFO output status flag signal. Defaults to FULL for the FIFO selected by the FIFOADR[1:0] pins.
8H	CTL2 or FLAGC	O/Z	H	Multiplexed pin whose function is selected by the following bits: IFCONFIG[1:0]. <b>CTL2</b> is a GPIF control output. <b>FLAGC</b> is a programmable slave FIFO output status flag signal. Defaults to EMPTY for the FIFO selected by the FIFOADR[1:0] pins.



**Table 4-1. FX2LP18 Pin Descriptions (continued)<sup>[9]</sup>**

56 VFBGA	Name	Type	Default	Description
2G	IFCLK	I/O/Z	Z	Interface Clock, used for synchronously clocking data into or out of the slave FIFO's. IFCLK also serves as a timing reference for all slave FIFO control signals and GPIF. When internal clocking is used (IFCONFIG.7 = 1) the IFCLK pin can be configured to output 30/48 MHz by bits IFCONFIG.5 and IFCONFIG.6. IFCLK may be inverted, whether internally or externally sourced, by setting the bit IFCONFIG.4 =1.
7B	WAKEUP	Input	N/A	<b>USB Wakeup.</b> If the 8051 is in suspend, asserting this pin starts up the oscillator and interrupts the 8051 to allow it to exit the suspend mode. Holding WAKEUP asserted inhibits the MoBL-USB <sup>®</sup> chip from suspending. This pin has programmable polarity (WAKEUP.4).
3F	SCL	OD	Z	<b>Clock</b> for the I <sup>2</sup> C interface. Connect to V <sub>CC_IO</sub> or V <sub>CC</sub> with a 2.2K - 10K pull up resistor. (An I <sup>2</sup> C peripheral is required).
3G	SDA	OD	Z	<b>Data</b> for the I <sup>2</sup> C interface. Connect to V <sub>CC_IO</sub> or V <sub>CC</sub> with a 2.2K - 10K pull up resistor. (An I <sup>2</sup> C peripheral is required).
5A	V <sub>CC_IO</sub>	Power	N/A	<b>VCC.</b> Connect this pin to 1.8V to 3.3V power source. Appropriate bulk/bypass capacitance should be provided for this supply rail.
5B	V <sub>CC_IO</sub>	Power	N/A	<b>VCC.</b> Connect this pin to 1.8V to 3.3V power source
7E	V <sub>CC_IO</sub>	Power	N/A	<b>VCC.</b> Connect this pin to 1.8V to 3.3V power source.
8E	V <sub>CC_IO</sub>	Power	N/A	<b>VCC.</b> Connect this pin to 1.8V to 3.3V power source.
5C	V <sub>CC_D</sub>	Power	N/A	<b>VCC.</b> Connect this pin to 1.8V power source.(Supplies power to internal digital 1.8V circuits) Appropriate bulk/bypass capacitance should be provided for this supply rail.
1G	V <sub>CC_A</sub>	Power	N/A	<b>VCC.</b> Connect this pin to 1.8V power source.(Supplies power to internal analog 1.8V circuits)
1H	GND	Ground	N/A	<b>Ground.</b>
2H	GND	Ground	N/A	<b>Ground.</b>
4A	GND	Ground	N/A	<b>Ground.</b>
4B	GND	Ground	N/A	<b>Ground.</b>
4C	GND	Ground	N/A	<b>Ground.</b>
7D	GND	Ground	N/A	<b>Ground.</b>
8D	GND	Ground	N/A	<b>Ground.</b>

## 5.0 Register Summary

FX2LP18 register bit definitions are described in the *MoBL-USB TRM* in greater detail.

**Table 5-1. FX2LP18 Register Summary**

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
GPIF Waveform Memories													
E400	128	WAVEDATA	GPIF Waveform Descriptor 0, 1, 2, 3 data	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
E480	128	Reserved											
GENERAL CONFIGURATION													
E50D		GPCR2	General Purpose Configuration Register 2	Reserved	Reserved	Reserved	FULL_SPEED_ONLY	Reserved	Reserved	Reserved	Reserved	00000000	R
E600	1	CPUCS	CPU Control & Status	0	0	PORTCSTB	CLKSPD1	CLKSPD0	CLKINV	CLKOE	8051RES	00000010	rrbbbbb
E601	1	IFCONFIG	Interface Configuration (Ports, GPIF, slave FIFO's)	IFCLKSRC	3048MHZ	IFCLKOE	IFCLKPOL	ASYNC	GSTATE	IFCFG1	IFCFG0	10000000	RW
E602	1	PINFLAGSAB <sup>[10]</sup>	Slave FIFO FLAGA and FLAGB Pin Configuration	FLAGB3	FLAGB2	FLAGB1	FLAGB0	FLAGA3	FLAGA2	FLAGA1	FLAGA0	00000000	RW
E603	1	PINFLAGSCD <sup>[10]</sup>	Slave FIFO FLAGC and FLAGD Pin Configuration	FLAGD3	FLAGD2	FLAGD1	FLAGD0	FLAGC3	FLAGC2	FLAGC1	FLAGC0	00000000	RW
E604	1	FIFORESET <sup>[10]</sup>	Restore FIFO's to default state	NAKALL	0	0	0	EP3	EP2	EP1	EP0	xxxxxxx	W
E605	1	BREAKPT	Breakpoint Control	0	0	0	0	BREAK	BPPULSE	BPEN	0	00000000	rrrrbbbr
E606	1	BPADDRH	Breakpoint Address H	A15	A14	A13	A12	A11	A10	A9	A8	xxxxxxx	RW
E607	1	BPADDRL	Breakpoint Address L	A7	A6	A5	A4	A3	A2	A1	A0	xxxxxxx	RW
E608	1	Reserved	Reserved	0	0	0	0	0	0	0	0	00000000	rrrrrrbb
E609	1	FIFOPINPOLAR <sup>[10]</sup>	Slave FIFO Interface pins polarity	0	0	PKTEND	SLOE	SLRD	SLWR	EF	FF	00000000	rrbbbbb
E60A	1	REVID	Chip Revision	rv7	rv6	rv5	rv4	rv3	rv2	rv1	rv0	RevA 00000001	R
E60B	1	REVCTL <sup>[10]</sup>	Chip Revision Control	0	0	0	0	0	0	dyn_out	enh_pkt	00000000	rrrrrrbb
UDMA													
E60C	1	GPIFHOLDAMOUNT	MSTB Hold Time (for UDMA)	0	0	0	0	0	0	HOLDTIME1	HOLDTIME0	00000000	rrrrrrbb
	3	Reserved											
ENDPOINT CONFIGURATION													
E610	1	EP1OUTCFG	Endpoint 1-OUT Configuration	VALID	0	TYPE1	TYPE0	0	0	0	0	10100000	rbbrrrr
E611	1	EP1INCFG	Endpoint 1-IN Configuration	VALID	0	TYPE1	TYPE0	0	0	0	0	10100000	rbbrrrr
E612	1	EP2CFG	Endpoint 2 Configuration	VALID	DIR	TYPE1	TYPE0	SIZE	0	BUF1	BUF0	10100010	bbbbbrbb
E613	1	EP4CFG	Endpoint 4 Configuration	VALID	DIR	TYPE1	TYPE0	0	0	0	0	10100000	bbbbrrrr
E614	1	EP6CFG	Endpoint 6 Configuration	VALID	DIR	TYPE1	TYPE0	SIZE	0	BUF1	BUF0	11100010	bbbbbrbb
E615	1	EP8CFG	Endpoint 8 Configuration	VALID	DIR	TYPE1	TYPE0	0	0	0	0	11100000	bbbbrrrr
	2	Reserved											
E618	1	EP2FIFOCFG <sup>[10]</sup>	Endpoint 2/slave FIFO configuration	0	INFM1	OEP1	AUTOOUT	AUTOIN	ZEROLENIN	0	WORDWIDE	00000101	rbbbbrbb
E619	1	EP4FIFOCFG <sup>[10]</sup>	Endpoint 4/slave FIFO configuration	0	INFM1	OEP1	AUTOOUT	AUTOIN	ZEROLENIN	0	WORDWIDE	00000101	rbbbbrbb
E61A	1	EP6FIFOCFG <sup>[10]</sup>	Endpoint 6/slave FIFO configuration	0	INFM1	OEP1	AUTOOUT	AUTOIN	ZEROLENIN	0	WORDWIDE	00000101	rbbbbrbb
E61B	1	EP8FIFOCFG <sup>[10]</sup>	Endpoint 8/slave FIFO configuration	0	INFM1	OEP1	AUTOOUT	AUTOIN	ZEROLENIN	0	WORDWIDE	00000101	rbbbbrbb
E61C	4	Reserved											
E620	1	EP2AUTOINLENH <sup>[10]</sup>	Endpoint 2 AUTOIN Packet Length H	0	0	0	0	0	PL10	PL9	PL8	00000010	rrrrrrbb
E621	1	EP2AUTOINLENL <sup>[10]</sup>	Endpoint 2 AUTOIN Packet Length L	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
E622	1	EP4AUTOINLENH <sup>[10]</sup>	Endpoint 4 AUTOIN Packet Length H	0	0	0	0	0	0	PL9	PL8	00000010	rrrrrrbb
E623	1	EP4AUTOINLENL <sup>[10]</sup>	Endpoint 4 AUTOIN Packet Length L	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
E624	1	EP6AUTOINLENH <sup>[10]</sup>	Endpoint 6 AUTOIN Packet Length H	0	0	0	0	0	PL10	PL9	PL8	00000010	rrrrrrbb
E625	1	EP6AUTOINLENL <sup>[10]</sup>	Endpoint 6 AUTOIN Packet Length L	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
E626	1	EP8AUTOINLENH <sup>[10]</sup>	Endpoint 8 AUTOIN Packet Length H	0	0	0	0	0	0	PL9	PL8	00000010	rrrrrrbb
E627	1	EP8AUTOINLENL <sup>[10]</sup>	Endpoint 8 AUTOIN Packet Length L	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
E628	1	ECCCFG	ECC Configuration	0	0	0	0	0	0	0	ECCM	00000000	rrrrrrbb
E629	1	ECCRESET	ECC Reset	x	x	x	x	x	x	x	x	00000000	W
E62A	1	ECC1B0	ECC1 Byte 0 Address	LINE15	LINE14	LINE13	LINE12	LINE11	LINE10	LINE9	LINE8	00000000	R
E62B	1	ECC1B1	ECC1 Byte 1 Address	LINE7	LINE6	LINE5	LINE4	LINE3	LINE2	LINE1	LINE0	00000000	R

**Note**

10. Read and writes to these registers may require synchronization delay, see Technical Reference Manual for 'Synchronization Delay.'

Table 5-1. FX2LP18 Register Summary (continued)

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
E62C	1	ECC1B2	ECC1 Byte 2 Address	COL5	COL4	COL3	COL2	COL1	COL0	LINE17	LINE16	00000000	R
E62D	1	ECC2B0	ECC2 Byte 0 Address	LINE15	LINE14	LINE13	LINE12	LINE11	LINE10	LINE9	LINE8	00000000	R
E62E	1	ECC2B1	ECC2 Byte 1 Address	LINE7	LINE6	LINE5	LINE4	LINE3	LINE2	LINE1	LINE0	00000000	R
E62F	1	ECC2B2	ECC2 Byte 2 Address	COL5	COL4	COL3	COL2	COL1	COL0	0	0	00000000	R
E630 H.S.	1	EP2FIFO PFH <sup>[10]</sup>	Endpoint 2/slave FIFO Programmable Flag H	DECIS	PKTSTAT	IN:PKTS[2] OUT:PFC12	IN:PKTS[1] OUT:PFC11	IN:PKTS[0] OUT:PFC10	0	PFC9	PFC8	10001000	bbbbbrbb
E630 F.S.	1	EP2FIFO PFH <sup>[10]</sup>	Endpoint 2/slave FIFO Programmable Flag H	DECIS	PKTSTAT	OUT:PFC12	OUT:PFC11	OUT:PFC10	0	PFC9	IN:PKTS[2] OUT:PFC8	10001000	bbbbbrbb
E631 H.S.	1	EP2FIFO PFL <sup>[10]</sup>	Endpoint 2/slave FIFO Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E631 F.S.	1	EP2FIFO PFL <sup>[10]</sup>	Endpoint 2/slave FIFO Programmable Flag L	IN:PKTS[1] OUT:PFC7	IN:PKTS[0] OUT:PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E632 H.S.	1	EP4FIFO PFH <sup>[10]</sup>	Endpoint 4/slave FIFO Programmable Flag H	DECIS	PKTSTAT	0	IN:PKTS[1] OUT:PFC10	IN:PKTS[0] OUT:PFC9	0	0	PFC8	10001000	bbrbrrbb
E632 F.S.	1	EP4FIFO PFH <sup>[10]</sup>	Endpoint 4/slave FIFO Programmable Flag H	DECIS	PKTSTAT	0	OUT:PFC10	OUT:PFC9	0	0	PFC8	10001000	bbrbrrbb
E633 H.S.	1	EP4FIFO PFL <sup>[10]</sup>	Endpoint 4/slave FIFO Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E633 F.S.	1	EP4FIFO PFL <sup>[10]</sup>	Endpoint 4/slave FIFO Programmable Flag L	IN:PKTS[1] OUT:PFC7	IN:PKTS[0] OUT:PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E634 H.S.	1	EP6FIFO PFH <sup>[10]</sup>	Endpoint 6/slave FIFO Programmable Flag H	DECIS	PKTSTAT	IN:PKTS[2] OUT:PFC12	IN:PKTS[1] OUT:PFC11	IN:PKTS[0] OUT:PFC10	0	PFC9	PFC8	00001000	bbbbbrbb
E634 F.S.	1	EP6FIFO PFH <sup>[10]</sup>	Endpoint 6/slave FIFO Programmable Flag H	DECIS	PKTSTAT	OUT:PFC12	OUT:PFC11	OUT:PFC10	0	PFC9	IN:PKTS[2] OUT:PFC8	00001000	bbbbbrbb
E635 H.S.	1	EP6FIFO PFL <sup>[10]</sup>	Endpoint 6/slave FIFO Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E635 F.S.	1	EP6FIFO PFL <sup>[10]</sup>	Endpoint 6/slave FIFO Programmable Flag L	IN:PKTS[1] OUT:PFC7	IN:PKTS[0] OUT:PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E636 H.S.	1	EP8FIFO PFH <sup>[10]</sup>	Endpoint 8/slave FIFO Programmable Flag H	DECIS	PKTSTAT	0	IN:PKTS[1] OUT:PFC10	IN:PKTS[0] OUT:PFC9	0	0	PFC8	00001000	bbrbrrbb
E636 F.S.	1	EP8FIFO PFH <sup>[10]</sup>	Endpoint 8/slave FIFO Programmable Flag H	DECIS	PKTSTAT	0	OUT:PFC10	OUT:PFC9	0	0	PFC8	00001000	bbrbrrbb
E637 H.S.	1	EP8FIFO PFL <sup>[10]</sup>	Endpoint 8/slave FIFO Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E637 F.S.	1	EP8FIFO PFL <sup>[10]</sup>	Endpoint 8/slave FIFO Programmable Flag L	IN:PKTS[1] OUT:PFC7	IN:PKTS[0] OUT:PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
	8	Reserved											
E640	1	EP2ISOINPKTS	EP2 (if ISO) IN Packets per frame (1-3)	AADJ	0	0	0	0	0	INPPF1	INPPF0	00000001	brrrrrbb
E641	1	EP4ISOINPKTS	EP4 (if ISO) IN Packets per frame (1-3)	AADJ	0	0	0	0	0	INPPF1	INPPF0	00000001	brrrrrrr
E642	1	EP6ISOINPKTS	EP6 (if ISO) IN Packets per frame (1-3)	AADJ	0	0	0	0	0	INPPF1	INPPF0	00000001	brrrrrbb
E643	1	EP8ISOINPKTS	EP8 (if ISO) IN Packets per frame (1-3)	AADJ	0	0	0	0	0	INPPF1	INPPF0	00000001	brrrrrrr
E644	4	Reserved											
E648	1	INPKTEND <sup>[10]</sup>	Force IN Packet End	Skip	0	0	0	EP3	EP2	EP1	EP0	xxxxxxx	IW
E649	1	OUTPKTEND <sup>[10]</sup>	Force OUT Packet End	Skip	0	0	0	EP3	EP2	EP1	EP0	xxxxxxx	IW
		INTERRUPTS											
E650	1	EP2FIFOIE <sup>[10]</sup>	Endpoint 2 slave FIFO Flag Interrupt Enable	0	0	0	0	EDGE PF	PF	EF	FF	00000000	RW
E651	1	EP2FIFOIRQ <sup>[10,11]</sup>	Endpoint 2 slave FIFO Flag Interrupt Request	0	0	0	0	0	PF	EF	FF	00000000	rrrrrbbb
E652	1	EP4FIFOIE <sup>[10]</sup>	Endpoint 4 slave FIFO Flag Interrupt Enable	0	0	0	0	EDGE PF	PF	EF	FF	00000000	RW
E653	1	EP4FIFOIRQ <sup>[10,11]</sup>	Endpoint 4 slave FIFO Flag Interrupt Request	0	0	0	0	0	PF	EF	FF	00000000	rrrrrbbb
E654	1	EP6FIFOIE <sup>[10]</sup>	Endpoint 6 slave FIFO Flag Interrupt Enable	0	0	0	0	EDGE PF	PF	EF	FF	00000000	RW
E655	1	EP6FIFOIRQ <sup>[10,11]</sup>	Endpoint 6 slave FIFO Flag Interrupt Request	0	0	0	0	0	PF	EF	FF	00000000	rrrrrbbb
E656	1	EP8FIFOIE <sup>[10]</sup>	Endpoint 8 slave FIFO Flag Interrupt Enable	0	0	0	0	EDGE PF	PF	EF	FF	00000000	RW
E657	1	EP8FIFOIRQ <sup>[10,11]</sup>	Endpoint 8 slave FIFO Flag Interrupt Request	0	0	0	0	0	PF	EF	FF	00000000	rrrrrbbb
E658	1	IBNIE	IN-BULK-NAK Interrupt Enable	0	0	EP8	EP6	EP4	EP2	EP1	EP0	00000000	RW
E659	1	IBNIRQ <sup>[11]</sup>	IN-BULK-NAK interrupt Request	0	0	EP8	EP6	EP4	EP2	EP1	EP0	0xxxxx	rrbbbbb
E65A	1	NAKIE	Endpoint Ping-NAK/IBN Interrupt Enable	EP8	EP6	EP4	EP2	EP1	EP0	0	IBN	00000000	RW
E65B	1	NAKIRQ <sup>[11]</sup>	Endpoint Ping-NAK/IBN Interrupt Request	EP8	EP6	EP4	EP2	EP1	EP0	0	IBN	xxxxxx0x	bbbbbrbb
E65C	1	USBIE	USB Int Enables	0	EP0ACK	HSGRANT	URES	SUSP	SUTOK	SOF	SUDAV	00000000	RW
E65D	1	USBIRQ <sup>[11]</sup>	USB Interrupt Requests	0	EP0ACK	HSGRANT	URES	SUSP	SUTOK	SOF	SUDAV	0xxxxxx	rbbbbbb

Note

11. The register can only be reset, it cannot be set.

Table 5-1. FX2LP18 Register Summary (continued)

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
E65E	1	EPIE	Endpoint Interrupt Enables	EP8	EP6	EP4	EP2	EP1OUT	EP1IN	EP0OUT	EP0IN	00000000	RW
E65F	1	EPIRQ <sup>[10]</sup>	Endpoint Interrupt Requests	EP8	EP6	EP4	EP2	EP1OUT	EP1IN	EP0OUT	EP0IN	0	RW
E660	1	GPIFIE <sup>[10]</sup>	GPIF Interrupt Enable	0	0	0	0	0	0	GPIFWF	GPIFDONE	00000000	RW
E661	1	GPIFIRQ <sup>[10]</sup>	GPIF Interrupt Request	0	0	0	0	0	0	GPIFWF	GPIFDONE	000000xx	RW
E662	1	USBERRIE	USB Error Interrupt Enables	ISOEP8	ISOEP6	ISOEP4	ISOEP2	0	0	0	ERRLIMIT	00000000	RW
E663	1	USBERRIRQ <sup>[10]</sup>	USB Error Interrupt Requests	ISOEP8	ISOEP6	ISOEP4	ISOEP2	0	0	0	ERRLIMIT	0000000x	bbbbrrb
E664	1	ERRCNTLIM	USB Error counter and limit	EC3	EC2	EC1	EC0	LIMIT3	LIMIT2	LIMIT1	LIMIT0	xxxx0100	rrrrbbbb
E665	1	CLRERRCNT	Clear Error Counter EC3:0	x	x	x	x	x	x	x	x	xxxxxxx	W
E666	1	INT2IVEC	Interrupt 2 (USB) Autovector	0	I2V4	I2V3	I2V2	I2V1	I2V0	0	0	00000000	R
E667	1	Reserved		1	0	0	0	0	0	0	0	10000000	R
E668	1	INTSET-UP	Interrupt 2&4 set-up	0	0	0	0	AV2EN	0	Reserved	AV4EN	00000000	RW
E669	7	Reserved											
		INPUT/OUTPUT											
E670	1	PORTACFG	I/O PORTA Alternate Configuration	FLAGD	SLCS	0	0	0	0	INT1	INT0	00000000	RW
E671	1	PORTCCFG	I/O PORTC Alternate Configuration	GPIFA7	GPIFA6	GPIFA5	GPIFA4	GPIFA3	GPIFA2	GPIFA1	GPIFA0	00000000	RW
E672	1	PORTECFG	I/O PORTE Alternate Configuration	GPIFA8	T2EX	INT6	RXD1OUT	RXD0OUT	T2OUT	T1OUT	T0OUT	00000000	RW
E673	4	Reserved											
E677	1	Reserved											
E678	1	I2CS	I <sup>2</sup> C Bus Control & Status	START	STOP	LASTRD	ID1	ID0	BERR	ACK	DONE	000x0000	bbbbrrrr
E679	1	I2DAT	I <sup>2</sup> C Bus Data	d7	d6	d5	d4	d3	d2	d1	d0	xxxxxxx	RW
E67A	1	I2CTL	I <sup>2</sup> C Bus Control	0	0	0	0	0	0	STOPIE	400KHZ	00000000	RW
E67B	1	XAUTODAT1	Autoptr1 MOVX access, when APTREN=1	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
E67C	1	XAUTODAT2	Autoptr2 MOVX access, when APTREN=1	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
		UDMA CRC											
E67D	1	UDMACRCH <sup>[10]</sup>	UDMA CRC MSB	CRC15	CRC14	CRC13	CRC12	CRC11	CRC10	CRC9	CRC8	01001010	RW
E67E	1	UDMACRCL <sup>[10]</sup>	UDMA CRC LSB	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0	10111010	RW
E67F	1	UDMACRC-QUALIFIER	UDMA CRC Qualifier	QENABLE	0	0	0	QSTATE	Q SIGNAL2	Q SIGNAL1	Q SIGNAL0	00000000	bbbbrrrr
		USB CONTROL											
E680	1	USBCS	USB Control & Status	HSM	0	0	0	DISCON	NOSYNSOF	RENUM	SIGRSUME	x0000000	rrrrbbbb
E681	1	SUSPEND	Put chip into suspend	x	x	x	x	x	x	x	x	xxxxxxx	W
E682	1	WAKEUPCS	Wakeup Control & Status	WU2	WU	WU2POL	WUPOL	0	DPEN	WU2EN	WUEN	xx000101	bbbbrrbbb
E683	1	TOGCTL	Toggle Control	Q	S	R	IO	EP3	EP2	EP1	EP0	x0000000	rrrrbbbb
E684	1	USBFRAMEH	USB Frame count H	0	0	0	0	0	FC10	FC9	FC8	00000xxx	R
E685	1	USBFRAMEL	USB Frame count L	FC7	FC6	FC5	FC4	FC3	FC2	FC1	FC0	xxxxxxx	R
E686	1	MICROFRAME	Microframe count, 0-7	0	0	0	0	0	MF2	MF1	MF0	00000xxx	R
E687	1	FNADDR	USB Function address	0	FA6	FA5	FA4	FA3	FA2	FA1	FA0	0xxxxxx	R
E688	2	Reserved											
		ENDPOINTS											
E68A	1	EP0BCH <sup>[10]</sup>	Endpoint 0 Byte Count H	(BC15)	(BC14)	(BC13)	(BC12)	(BC11)	(BC10)	(BC9)	(BC8)	xxxxxxx	RW
E68B	1	EP0BCL <sup>[10]</sup>	Endpoint 0 Byte Count L	(BC7)	BC6	BC5	BC4	BC3	BC2	BC1	BC0	xxxxxxx	RW
E68C	1	Reserved											
E68D	1	EP1OUTBC	Endpoint 1 OUT Byte Count	0	BC6	BC5	BC4	BC3	BC2	BC1	BC0	0xxxxxx	RW
E68E	1	Reserved											
E68F	1	EP1INBC	Endpoint 1 IN Byte Count	0	BC6	BC5	BC4	BC3	BC2	BC1	BC0	0xxxxxx	RW
E690	1	EP2BCH <sup>[10]</sup>	Endpoint 2 Byte Count H	0	0	0	0	0	BC10	BC9	BC8	00000xxx	RW
E691	1	EP2BCL <sup>[10]</sup>	Endpoint 2 Byte Count L	BC7/SKIP	BC6	BC5	BC4	BC3	BC2	BC1	BC0	xxxxxxx	RW
E692	2	Reserved											
E694	1	EP4BCH <sup>[10]</sup>	Endpoint 4 Byte Count H	0	0	0	0	0	0	BC9	BC8	00000xxx	RW
E695	1	EP4BCL <sup>[10]</sup>	Endpoint 4 Byte Count L	BC7/SKIP	BC6	BC5	BC4	BC3	BC2	BC1	BC0	xxxxxxx	RW
E696	2	Reserved											
E698	1	EP6BCH <sup>[10]</sup>	Endpoint 6 Byte Count H	0	0	0	0	0	BC10	BC9	BC8	00000xxx	RW
E699	1	EP6BCL <sup>[10]</sup>	Endpoint 6 Byte Count L	BC7/SKIP	BC6	BC5	BC4	BC3	BC2	BC1	BC0	xxxxxxx	RW
E69A	2	Reserved											
E69C	1	EP8BCH <sup>[10]</sup>	Endpoint 8 Byte Count H	0	0	0	0	0	0	BC9	BC8	00000xxx	RW
E69D	1	EP8BCL <sup>[10]</sup>	Endpoint 8 Byte Count L	BC7/SKIP	BC6	BC5	BC4	BC3	BC2	BC1	BC0	xxxxxxx	RW
E69E	2	Reserved											
E6A0	1	EP0CS	Endpoint 0 Control and Status	HSNAK	0	0	0	0	0	BUSY	STALL	10000000	bbbbbbrrb

Table 5-1. FX2LP18 Register Summary (continued)

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
E6A1	1	EP1OUTCS	Endpoint 1 OUT Control and Status	0	0	0	0	0	0	BUSY	STALL	00000000	bbbbbrb
E6A2	1	EP1INCS	Endpoint 1 IN Control and Status	0	0	0	0	0	0	BUSY	STALL	00000000	bbbbbrb
E6A3	1	EP2CS	Endpoint 2 Control and Status	0	NPAK2	NPAK1	NPAK0	FULL	EMPTY	0	STALL	00101000	rrrrrrb
E6A4	1	EP4CS	Endpoint 4 Control and Status	0	0	NPAK1	NPAK0	FULL	EMPTY	0	STALL	00101000	rrrrrrb
E6A5	1	EP6CS	Endpoint 6 Control and Status	0	NPAK2	NPAK1	NPAK0	FULL	EMPTY	0	STALL	00000100	rrrrrrb
E6A6	1	EP8CS	Endpoint 8 Control and Status	0	0	NPAK1	NPAK0	FULL	EMPTY	0	STALL	00000100	rrrrrrb
E6A7	1	EP2FIFOFLGS	Endpoint 2 slave FIFO Flags	0	0	0	0	0	PF	EF	FF	00000010	R
E6A8	1	EP4FIFOFLGS	Endpoint 4 slave FIFO Flags	0	0	0	0	0	PF	EF	FF	00000010	R
E6A9	1	EP6FIFOFLGS	Endpoint 6 slave FIFO Flags	0	0	0	0	0	PF	EF	FF	00000110	R
E6AA	1	EP8FIFOFLGS	Endpoint 8 slave FIFO Flags	0	0	0	0	0	PF	EF	FF	00000110	R
E6AB	1	EP2FIFOBCH	Endpoint 2 slave FIFO total byte count H	0	0	0	BC12	BC11	BC10	BC9	BC8	00000000	R
E6AC	1	EP2FIFOBCL	Endpoint 2 slave FIFO total byte count L	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	00000000	R
E6AD	1	EP4FIFOBCH	Endpoint 4 slave FIFO total byte count H	0	0	0	0	0	BC10	BC9	BC8	00000000	R
E6AE	1	EP4FIFOBCL	Endpoint 4 slave FIFO total byte count L	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	00000000	R
E6AF	1	EP6FIFOBCH	Endpoint 6 slave FIFO total byte count H	0	0	0	0	BC11	BC10	BC9	BC8	00000000	R
E6B0	1	EP6FIFOBCL	Endpoint 6 slave FIFO total byte count L	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	00000000	R
E6B1	1	EP8FIFOBCH	Endpoint 8 slave FIFO total byte count H	0	0	0	0	0	BC10	BC9	BC8	00000000	R
E6B2	1	EP8FIFOBCL	Endpoint 8 slave FIFO total byte count L	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	00000000	R
E6B3	1	SUDPTRH	Set-up Data Pointer high address byte	A15	A14	A13	A12	A11	A10	A9	A8	xxxxxxx	RW
E6B4	1	SUDPTL	Set-up Data Pointer low address byte	A7	A6	A5	A4	A3	A2	A1	0	xxxxxxx	bbbbbrb
E6B5	1	SUDPTRCTL	Set-up Data Pointer Auto Mode	0	0	0	0	0	0	0	SDPAUTO	00000001	RW
	2	Reserved											
E6B8	8	SET-UPDAT	8 bytes of set-up data	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	R
			SET-UPDAT[0] = bmRequestType										
			SET-UPDAT[1] = bmRequest										
			SET-UPDAT[2:3] = wValue										
			SET-UPDAT[4:5] = wIndex										
			SET-UPDAT[6:7] = wLength										
		GPIF											
E6C0	1	GPIWFSELECT	Waveform Selector	SINGLEWR1	SINGLEWR0	SINGLERD1	SINGLERD0	FIFOWR1	FIFOWR0	FIFORD1	FIFORD0	11100100	RW
E6C1	1	GPIFIDLECS	GPIF Done, GPIF IDLE drive mode	DONE	0	0	0	0	0	0	IDLEDRV	10000000	RW
E6C2	1	GPIFIDLECTL	Inactive Bus, CTL states	0	0	0	0	0	CTL2	CTL1	CTL0	11111111	RW
E6C3	1	GPIFCTLCFG	CTL Drive Type	TRICTL	0	0	0	0	CTL2	CTL1	CTL0	00000000	RW
E6C4	1	Reserved										00000000	
E6C5	1	Reserved										00000000	
		FLOWSTATE											
E6C6	1	FLOWSTATE	Flowstate Enable and Selector	FSE	0	0	0	0	FS2	FS1	FS0	00000000	brrrrbb
E6C7	1	FLOWLOGIC	Flowstate Logic	LFUNC1	LFUNC0	TERMA2	TERMA1	TERMA0	TERMB2	TERMB1	TERMB0	00000000	RW
E6C8	1	FLOWEQ0CTL	CTL-Pin States in Flowstate (when Logic = 0)	CTL0E3	CTL0E2	CTL0E1	CTL0E0	0	CTL2	CTL1	CTL0	00000000	RW
E6C9	1	FLOWEQ1CTL	CTL-Pin States in Flowstate (when Logic = 1)	CTL0E3	CTL0E2	CTL0E1	CTL0E0	0	CTL2	CTL1	CTL0	00000000	RW
E6CA	1	FLOWHOLDOFF	Holdoff Configuration	HOPERIOD3	HOPERIOD2	HOPERIOD1	HOPERIOD0	HOSTATE	HOCTL2	HOCTL1	HOCTL0	00010010	RW
E6CB	1	FLOWSTB	Flowstate Strobe Configuration	SLAVE	RDYASYNC	CTLTOGL	SUSTAIN	0	MSTB2	MSTB1	MSTB0	00100000	RW
E6CC	1	FLOWSTBEDGE	Flowstate Rising/Falling Edge Configuration	0	0	0	0	0	FALLING	RISING		00000001	rrrrrrbb
E6CD	1	FLOWSTBPERIOD	Master-Strobe Half-Period	D7	D6	D5	D4	D3	D2	D1	D0	00000010	RW
E6CE	1	GPIFTCB3 <sup>[10]</sup>	GPIF Transaction Count Byte 3	TC31	TC30	TC29	TC28	TC27	TC26	TC25	TC24	00000000	RW

Table 5-1. FX2LP18 Register Summary (continued)

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
E6CF	1	GPIFTCB2 <sup>[10]</sup>	GPIF Transaction Count Byte 2	TC23	TC22	TC21	TC20	TC19	TC18	TC17	TC16	00000000	RW
E6D0	1	GPIFTCB1 <sup>[10]</sup>	GPIF Transaction Count Byte 1	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	00000000	RW
E6D1	1	GPIFTCB0 <sup>[10]</sup>	GPIF Transaction Count Byte 0	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	00000001	RW
	2	Reserved										00000000	RW
		Reserved											
		Reserved											
E6D2	1	EP2GPIFFLGSEL <sup>[10]</sup>	Endpoint 2 GPIF Flag select	0	0	0	0	0	0	FS1	FS0	00000000	RW
E6D3	1	EP2GPIPFSTOP	Endpoint 2 GPIF stop transaction on prog. flag	0	0	0	0	0	0	0	FIFO2FLAG	00000000	RW
E6D4	1	EP2GPIFTRIG <sup>[10]</sup>	Endpoint 2 GPIF Trigger	x	x	x	x	x	x	x	x	xxxxxxx	IW
	3	Reserved											
		Reserved											
		Reserved											
E6DA	1	EP4GPIFFLGSEL <sup>[10]</sup>	Endpoint 4 GPIF Flag select	0	0	0	0	0	0	FS1	FS0	00000000	RW
E6DB	1	EP4GPIPFSTOP	Endpoint 4 GPIF stop transaction on prog. flag	0	0	0	0	0	0	0	FIFO4FLAG	00000000	RW
E6DC	1	EP4GPIFTRIG <sup>[10]</sup>	Endpoint 4 GPIF Trigger	x	x	x	x	x	x	x	x	xxxxxxx	IW
	3	Reserved											
		Reserved											
		Reserved											
E6E2	1	EP6GPIFFLGSEL <sup>[10]</sup>	Endpoint 6 GPIF Flag select	0	0	0	0	0	0	FS1	FS0	00000000	RW
E6E3	1	EP6GPIPFSTOP	Endpoint 6 GPIF stop transaction on prog. flag	0	0	0	0	0	0	0	FIFO6FLAG	00000000	RW
E6E4	1	EP6GPIFTRIG <sup>[10]</sup>	Endpoint 6 GPIF Trigger	x	x	x	x	x	x	x	x	xxxxxxx	IW
	3	Reserved											
		Reserved											
		Reserved											
E6EA	1	EP8GPIFFLGSEL <sup>[10]</sup>	Endpoint 8 GPIF Flag select	0	0	0	0	0	0	FS1	FS0	00000000	RW
E6EB	1	EP8GPIPFSTOP	Endpoint 8 GPIF stop transaction on prog. flag	0	0	0	0	0	0	0	FIFO8FLAG	00000000	RW
E6EC	1	EP8GPIFTRIG <sup>[10]</sup>	Endpoint 8 GPIF Trigger	x	x	x	x	x	x	x	x	xxxxxxx	IW
	3	Reserved											
E6F0	1	XGPIFSGLDATH	GPIF Data H (16-bit mode only)	D15	D14	D13	D12	D11	D10	D9	D8	xxxxxxx	RW
E6F1	1	XGPIFSGLDATLX	Read/Write GPIF Data L & trigger transaction	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
E6F2	1	XGPIFSGLDATL-NOX	Read GPIF Data L, no transaction trigger	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	R
E6F3	1	GPIFREADYCFG	Internal RDY, Sync/Async, RDY pin states	INTRDY	SAS	TCXRDY5	0	0	0	0	0	00000000	bbrrrrr
E6F4	1	GPIFREADYSTAT	GPIF Ready Status	0	0	0	0	0	0	RDY1	RDY0	00xxxxxx	R
E6F5	1	GPIFABORT	Abort GPIF Waveforms	x	x	x	x	x	x	x	x	xxxxxxx	IW
E6F6	2	Reserved											
		ENDPOINT BUFFERS											
E740	64	EP0BUF	EP0-IN/-OUT buffer	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
E780	64	EP10UTBUF	EP1-OUT buffer	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
E7C0	64	EP1INBUF	EP1-IN buffer	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
E800	2048	Reserved											RW
F000	1024	EP2FIFOBUF	512/1024-byte EP 2/slave FIFO buffer (IN or OUT)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
F400	512	EP4FIFOBUF	512 byte EP 4/slave FIFO buffer (IN or OUT)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
F600	512	Reserved											
F800	1024	EP6FIFOBUF	512/1024-byte EP 6/slave FIFO buffer (IN or OUT)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
FC00	512	EP8FIFOBUF	512 byte EP 8/slave FIFO buffer (IN or OUT)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
FE00	512	Reserved											
xxxx		I <sup>2</sup> C Configuration Byte		0	DISCON	0	0	0	0	0	400KHZ	xxxxxxx <sup>[13]</sup>	n/a
		Special Function Registers (SFRs)											
80	1	IOA <sup>[12]</sup>	Port A (bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW

Notes

- 12. SFRs not part of the standard 8051 architecture.
- 13. If no EEPROM is detected by the SIE then the default is 00000000.



**Table 5-1. FX2LP18 Register Summary (continued)**

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
81	1	SP	Stack Pointer	D7	D6	D5	D4	D3	D2	D1	D0	00001111	RW
82	1	DPL0	Data Pointer 0 L	A7	A6	A5	A4	A3	A2	A1	A0	00000000	RW
83	1	DPH0	Data Pointer 0 H	A15	A14	A13	A12	A11	A10	A9	A8	00000000	RW
84	1	DPL1 <sup>[12]</sup>	Data Pointer 1 L	A7	A6	A5	A4	A3	A2	A1	A0	00000000	RW
85	1	DPH1 <sup>[12]</sup>	Data Pointer 1 H	A15	A14	A13	A12	A11	A10	A9	A8	00000000	RW
86	1	DPS <sup>[12]</sup>	Data Pointer 0/1 select	0	0	0	0	0	0	0	SEL	00000000	RW
87	1	PCON	Power Control	SMOD0	x	1	1	x	x	x	IDLE	00110000	RW
88	1	TCON	Timer/Counter Control (bit addressable)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000	RW
89	1	TMOD	Timer/Counter Mode Control	GATE	CT	M1	M0	GATE	CT	M1	M0	00000000	RW
8A	1	TL0	Timer 0 reload L	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
8B	1	TL1	Timer 1 reload L	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
8C	1	TH0	Timer 0 reload H	D15	D14	D13	D12	D11	D10	D9	D8	00000000	RW
8D	1	TH1	Timer 1 reload H	D15	D14	D13	D12	D11	D10	D9	D8	00000000	RW
8E	1	CKCON <sup>[12]</sup>	Clock Control	x	x	T2M	T1M	T0M	MD2	MD1	MD0	00000001	RW
8F	1	Reserved											
90	1	IOB <sup>[12]</sup>	Port B (bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
91	1	EXIF <sup>[12]</sup>	External Interrupt Flag(s)	IE5	IE4	I <sup>2</sup> CINT	USBNT	1	0	0	0	00001000	RW
92	1	MPAGE <sup>[12]</sup>	Upper Addr Byte of MOVX using @R0/@R1	A15	A14	A13	A12	A11	A10	A9	A8	00000000	RW
93	5	Reserved											
98	1	SCON0	Serial Port 0 Control (bit addressable)	SM0_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0	00000000	RW
99	1	SBUF0	Serial Port 0 Data Buffer	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
9A	1	AUTOPTRH1 <sup>[12]</sup>	Autopointer 1 Address H	A15	A14	A13	A12	A11	A10	A9	A8	00000000	RW
9B	1	AUTOPTRL1 <sup>[12]</sup>	Autopointer 1 Address L	A7	A6	A5	A4	A3	A2	A1	A0	00000000	RW
9C	1	Reserved											
9D	1	AUTOPTRH2 <sup>[12]</sup>	Autopointer 2 Address H	A15	A14	A13	A12	A11	A10	A9	A8	00000000	RW
9E	1	AUTOPTRL2 <sup>[12]</sup>	Autopointer 2 Address L	A7	A6	A5	A4	A3	A2	A1	A0	00000000	RW
9F	1	Reserved											
A0	1	IOC <sup>[12]</sup>	Port C (bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
A1	1	INT2CLR <sup>[12]</sup>	Interrupt 2 clear	x	x	x	x	x	x	x	x	xxxxxxx	W
A2	1	Reserved		x	x	x	x	x	x	x	x	xxxxxxx	W
A3	5	Reserved											
A8	1	IE	Interrupt Enable (bit addressable)	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0	00000000	RW
A9	1	Reserved											
AA	1	EP2468STAT <sup>[12]</sup>	Endpoint 2,4,6,8 status flags	EP8F	EP8E	EP6F	EP6E	EP4F	EP4E	EP2F	EP2E	01011010	R
AB	1	EP24FIFOFLGS <sup>[12]</sup>	Endpoint 2,4 slave FIFO status flags	0	EP4PF	EP4EF	EP4FF	0	EP2PF	EP2EF	EP2FF	00100010	R
AC	1	EP68FIFOFLGS <sup>[12]</sup>	Endpoint 6,8 slave FIFO status flags	0	EP8PF	EP8EF	EP8FF	0	EP6PF	EP6EF	EP6FF	01100110	R
AD	2	Reserved											
AF	1	AUTOPTRSETUP <sup>[12]</sup>	Autopointer 1&2 set-up	0	0	0	0	0	APTR2INC	APTR1INC	APTREN	00000110	RW
B0	1	IOD <sup>[12]</sup>	Port D (bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
B1	1	IOE <sup>[12]</sup>	Port E (NOT bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
B2	1	OEA <sup>[12]</sup>	Port A Output Enable	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
B3	1	OEB <sup>[12]</sup>	Port B Output Enable	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
B4	1	OEC <sup>[12]</sup>	Port C Output Enable	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
B5	1	OED <sup>[12]</sup>	Port D Output Enable	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
B6	1	OEE <sup>[12]</sup>	Port E Output Enable	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
B7	1	Reserved											
B8	1	IP	Interrupt Priority (bit addressable)	1	PS1	PT2	PS0	PT1	PX1	PT0	PX0	10000000	RW
B9	1	Reserved											
BA	1	EP01STAT <sup>[12]</sup>	Endpoint 0&1 Status	0	0	0	0	0	EP1INBSY	EP1OUTBSY	EP0BSY	00000000	R
BB	1	GPIFTRIG <sup>[12, 10]</sup>	Endpoint 2,4,6,8 GPIF slave FIFO Trigger	DONE	0	0	0	0	RW	EP1	EP0	10000xxx	brrrbbb
BC	1	Reserved											
BD	1	GPIFGLDATH <sup>[12]</sup>	GPIF Data H (16-bit mode only)	D15	D14	D13	D12	D11	D10	D9	D8	xxxxxxx	RW
BE	1	GPIFGLDATLX <sup>[12]</sup>	GPIF Data L w/Trigger	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	RW
BF	1	GPIFGLDATL-NOX <sup>[12]</sup>	GPIF Data L w/No Trigger	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxx	R
C0	1	SCON1 <sup>[12]</sup>	Serial Port 1 Control (bit addressable)	SM0_1	SM1_1	SM2_1	REN_1	TB8_1	RB8_1	TI_1	RI_1	00000000	RW
C1	1	SBUF1 <sup>[12]</sup>	Serial Port 1 Data Buffer	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
C2	6	Reserved											
C8	1	T2CON	Timer/Counter 2 Control (bit addressable)	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	CT2	CPRL2	00000000	RW

**Table 5-1. FX2LP18 Register Summary (continued)**

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
C9	1	Reserved											
CA	1	RCAP2L	Capture for Timer 2, auto-reload, up-counter	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
CB	1	RCAP2H	Capture for Timer 2, auto-reload, up-counter	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
CC	1	TL2	Timer 2 reload L	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
CD	1	TH2	Timer 2 reload H	D15	D14	D13	D12	D11	D10	D9	D8	00000000	RW
CE	2	Reserved											
D0	1	PSW	Program Status Word (bit addressable)	CY	AC	F0	RS1	RS0	OV	F1	P	00000000	RW
D1	7	Reserved											
D8	1	EICON <sup>[12]</sup>	External Interrupt Control	SMOD1	1	ERESI	RESI	INT6	0	0	0	01000000	RW
D9	7	Reserved											
E0	1	ACC	Accumulator (bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
E1	7	Reserved											
E8	1	EIE <sup>[12]</sup>	External Interrupt Enable(s)	1	1	1	EX6	EX5	EX4	EI <sup>PC</sup>	EUSB	11100000	RW
E9	7	Reserved											
F0	1	B	B (bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
F1	7	Reserved											
F8	1	EIP <sup>[12]</sup>	External Interrupt Priority Control	1	1	1	PX6	PX5	PX4	PI <sup>PC</sup>	PUSB	11100000	RW
F9	7	Reserved											

**Ledgend**

R = all bits read-only  
 W = all bits write-only  
 r = read-only bit  
 w = write-only bit  
 b = both read/write bit

**6.0 Absolute Maximum Ratings**

Storage Temperature .....	- 65°C to +150°C
Ambient Temperature with Power Supplied	
Industrial .....	- 40°C to +85°C
Supply Voltage to Ground Potential	
For 3.3V Power domain .....	- 0.5V to +4.0V
For 1.8V Power domain .....	- 0.5V to +2.0V
DC Input Voltage to Any Input Pin	
For pins under 3.3V Power Domain .....	3.6V <sup>[14]</sup>
For pins under 1.8V - 3.3V Power Domain (GPIO's) .....	1.89V to 3.6V <sup>[14]</sup>
(The GPIO's are not over voltage tolerant, except the SCL and SDA pins, which are 3.3V tolerant)	
DC Voltage Applied to Outputs in High Z State .....	- 0.5V to VCC + 0.5V
Maximum Power Dissipation	
From AVcc Supply .....	90 mW
From IO Supply .....	36 mW
From Core Supply .....	95 mW
Static Discharge Voltage .....	> 2000V
(I2C SCL and SDA pins only ... >1500V)	
Maximum Output Current, per I/O port .....	10 mA

**7.0 Operating Conditions**

T <sub>A</sub> (Ambient Temperature Under Bias)	
Industrial .....	- 40°C to +85°C
Supply Voltage	
3.3V Power Supply .....	3.0V to 3.6V
1.8V Power Supply .....	1.71V to 1.89V
Ground Voltage .....	0V
F <sub>OSC</sub> (Oscillator or Crystal Frequency) .....	24 MHz ± 100 ppm
.....	Parallel Resonant
.....	500 μW drive level
.....	Load capacitors 12 pF

**Note**

14. It is recommended to not power I/O when chip power is off.

## 8.0 DC Characteristics

**Table 8-1. DC Characteristics**

Parameter	Description	Conditions	Min.	Typ.	Max.	Unit
$V_{CC}$	3.3 V supply (to Osc. and PHY) <sup>[15]</sup>		3.00	3.3	3.60	V
$V_{CC\_IO}$	1.8V to 3.3V supply (to I/O)		1.71	1.8	3.60	V
$V_{CC\_A}$	1.8 V supply to Analog Core		1.71	1.8	1.89	V
$V_{CC\_D}$	1.8 V supply to Digital Core		1.71	1.8	1.89	V
$V_{IH}$	Input HIGH Voltage		$0.6 \cdot V_{CC\_IO}$		$V_{CC\_IO} + 10\%$	V
$V_{IL}$	Input LOW Voltage		0		$0.3 \cdot V_{CC\_IO}$	V
$V_{IH\_X}$	Crystal Input HIGH Voltage		2.0		3.60	V
$V_{IL\_X}$	Crystal Input LOW Voltage		-0.5		0.8	V
	Hysteresis		50			mV
$I_I$	Input Leakage Current	$0 < V_{IN} < V_{CC\_IO}$			$\pm 10$	$\mu A$
$V_{OH}$	Output Voltage HIGH	$I_{OUT} = 4 \text{ mA}$	$V_{CC\_IO} - 0.4$			V
$V_{OL}$	Output LOW Voltage	$I_{OUT} = -4 \text{ mA}$			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	Connected		220	$380^{[16]}$	$\mu A$
		Disconnected		20	$150^{[16]}$	$\mu A$
$I_{CC\_AVCC}$	Supply Current ( $V_{CC}$ )	8051 running, connected to USB HS		15	25	mA
		8051 running, connected to USB FS		10	20	mA
$I_{CC\_IO}$	Supply Current ( $V_{CC\_IO}$ )	8051 running, connected to USB HS		3	10	mA
		8051 running, connected to USB FS		1	5	mA
$I_{CC\_CORE}$	Supply Current ( $V_{CC\_CORE}$ )	8051 running, connected to USB HS		32	50	mA
		8051 running, connected to USB FS		24	40	mA
$T_{RESET}$	Reset Time after Valid Power	$V_{CC} \text{ min} = 3.0V$	5.0			ms
	Pin Reset after powered on		200			$\mu s$

**Notes**

15. The pins for this supply can be floated in low-power mode.  
 16. Measured at Maximum  $V_{CC}$ , 25°C.

## 9.0 AC Electrical Characteristics

### 9.1 USB Transceiver

USB 2.0-compliant in full- and high-speed modes.

### 9.2 GPIF Synchronous Signals

Figure 9-1. GPIF Synchronous Signals Timing Diagram<sup>[17]</sup>

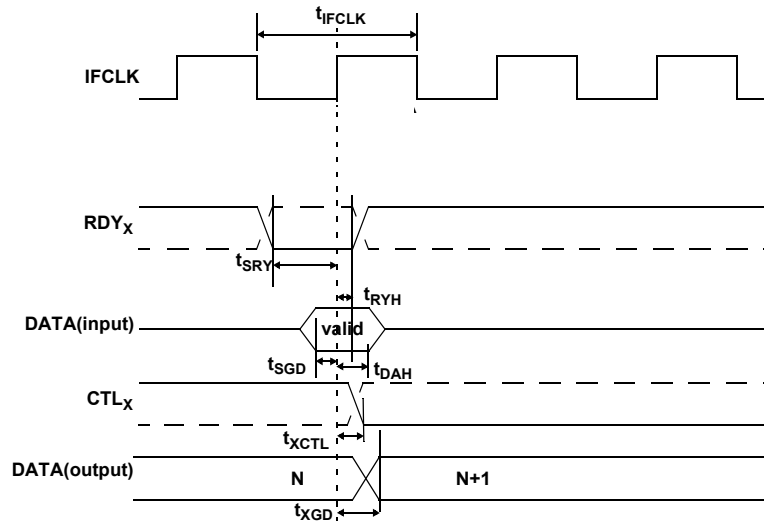


Table 9-1. GPIF Synchronous Signals Parameters with Internally Sourced IFCLK<sup>[17,18]</sup>

Parameter	Description	Min.	Max.	Unit
$t_{IFCLK}$	IFCLK Period	20.83		ns
$t_{SRY}$	RDY <sub>x</sub> to Clock Set-up Time	8.9		ns
$t_{RYH}$	Clock to RDY <sub>x</sub>	0		ns
$t_{SGD}$	GPIF Data to Clock Set-up Time	9.2		ns
$t_{DAH}$	GPIF Data Hold Time	0		ns
$t_{XGD}$	Clock to GPIF Data Output Propagation Delay		11	ns
$t_{XCTL}$	Clock to CTL <sub>x</sub> Output Propagation Delay		6.7	ns

Table 9-2. GPIF Synchronous Signals Parameters with Externally Sourced IFCLK<sup>[18]</sup>

Parameter	Description	Min.	Max.	Unit
$t_{IFCLK}$	IFCLK Period <sup>[19]</sup>	20.83	200	ns
$t_{SRY}$	RDY <sub>x</sub> to Clock Set-up Time	2.9		ns
$t_{RYH}$	Clock to RDY <sub>x</sub>	3.7		ns
$t_{SGD}$	GPIF Data to Clock Set-up Time	3.2		ns
$t_{DAH}$	GPIF Data Hold Time	4.5		ns
$t_{XGD}$	Clock to GPIF Data Output Propagation Delay		15	ns
$t_{XCTL}$	Clock to CTL <sub>x</sub> Output Propagation Delay		13.06	ns

**Notes**

- 17. Dashed lines denote signals with programmable polarity.
- 18. GPIF asynchronous RDY<sub>x</sub> signals have a minimum set-up time of 50 ns when using internal 48 MHz IFCLK.
- 19. IFCLK must not exceed 48 MHz.

9.3 Slave FIFO Synchronous Read

Figure 9-2. Slave FIFO Synchronous Read Timing Diagram<sup>[17]</sup>

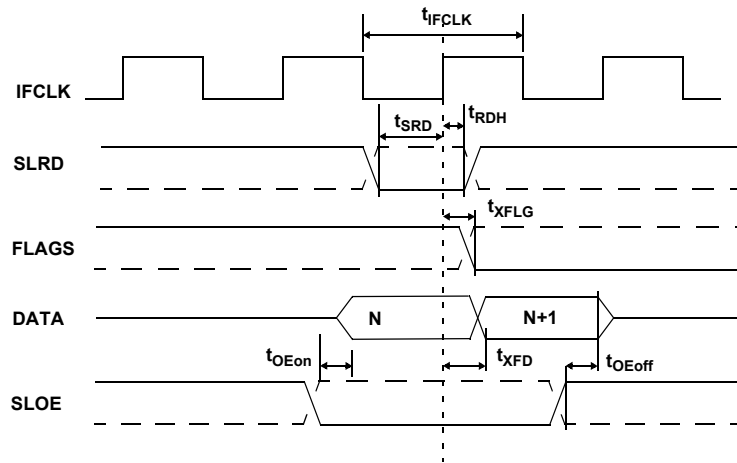


Table 9-3. Slave FIFO Synchronous Read Parameters with Internally Sourced IFCLK<sup>[18]</sup>

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	2.15	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 9-4. Slave FIFO Synchronous Read Parameters with Externally Sourced IFCLK<sup>[18]</sup>

Parameter	Description	Min.	Max.	Unit
$t_{IFCLK}$	IFCLK Period	20.83	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	2.15	10.5	ns
$t_{XFLG}$	Clock to FLAGS Output Propagation Delay		13.5	ns
$t_{XFD}$	Clock to FIFO Data Output Propagation Delay		17.31	ns



9.4 Slave FIFO Asynchronous Read

Figure 9-3. Slave FIFO Asynchronous Read Timing Diagram<sup>[17]</sup>

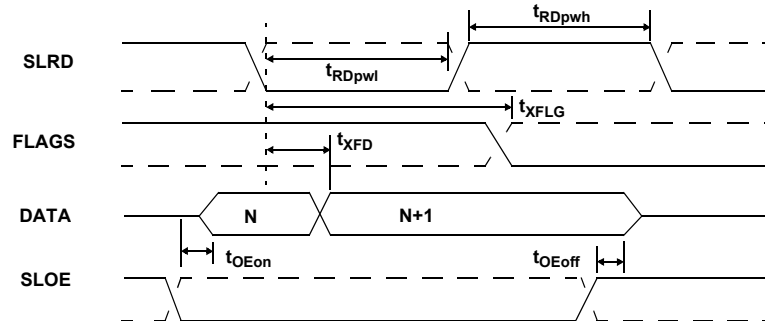


Table 9-5. Slave FIFO Asynchronous Read Parameters<sup>[20]</sup>

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	2.15	10.5	ns

Note

20. Slave FIFO asynchronous parameter values use internal IFCLK setting at 48 MHz.

9.5 Slave FIFO Synchronous Write

Figure 9-4. Slave FIFO Synchronous Write Timing Diagram<sup>[17]</sup>

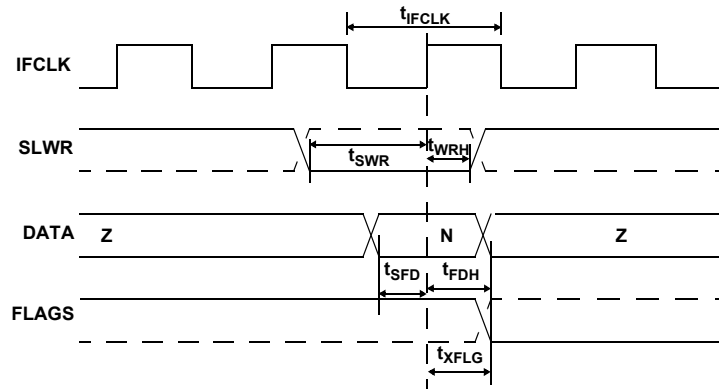


Table 9-6. Slave FIFO Synchronous Write Parameters with Internally Sourced IFCLK<sup>[18]</sup>

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	10.64		ns
$t_{FDH}$	Clock to FIFO Data Hold Time	0		ns
$t_{XFLG}$	Clock to FLAGS Output Propagation Time		9.5	ns

Table 9-7. Slave FIFO Synchronous Write Parameters with Externally Sourced IFCLK<sup>[10]</sup>

Parameter	Description	Min.	Max.	Unit
$t_{IFCLK}$	IFCLK Period	20.83	200	ns
$t_{SWR}$	SLWR to Clock Set-up Time	12.1		ns
$t_{WRH}$	Clock to SLWR Hold Time	3.6		ns
$t_{SFD}$	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

9.6 Slave FIFO Asynchronous Write

Figure 9-5. Slave FIFO Asynchronous Write Timing Diagram<sup>[17]</sup>

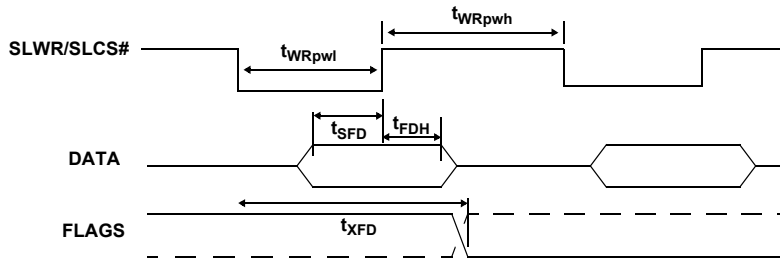


Table 9-8. Slave FIFO Asynchronous Write Parameters with Internally Sourced IFCLK<sup>[20]</sup>

Parameter	Description	Min.	Max.	Unit
tWRpwl	SLWR Pulse LOW	50		ns
tWRpwh	SLWR Pulse HIGH	70		ns
tSFD	SLWR to FIFO DATA Set-up Time	10		ns
tFDH	FIFO DATA to SLWR Hold Time	10		ns
tXFD	SLWR to FLAGS Output Propagation Delay		70	ns

9.7 Slave FIFO Synchronous Packet End Strobe

Figure 9-6. Slave FIFO Synchronous Packet End Strobe Timing Diagram<sup>[17]</sup>

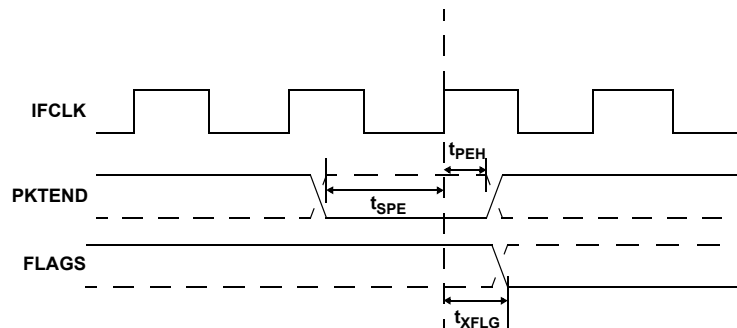


Table 9-9. Slave FIFO Synchronous Packet End Strobe Parameters with Internally Sourced IFCLK<sup>[10]</sup>

Parameter	Description	Min.	Max.	Unit
tIFCLK	IFCLK Period	20.83		ns
tSPE	PKTEND to Clock Set-up Time	14.6		ns
tPEH	Clock to PKTEND Hold Time	0		ns
tXFLG	Clock to FLAGS Output Propagation Delay		9.5	ns

Table 9-10. Slave FIFO Synchronous Packet End Strobe Parameters with Externally Sourced IFCLK<sup>[10]</sup>

Parameter	Description	Min.	Max.	Unit
tIFCLK	IFCLK Period	20.83	200	ns
tSPE	PKTEND to Clock Set-up Time	8.6		ns
tPEH	Clock to PKTEND Hold Time	3.04		ns
tXFLG	Clock to FLAGS Output Propagation Delay		13.5	ns

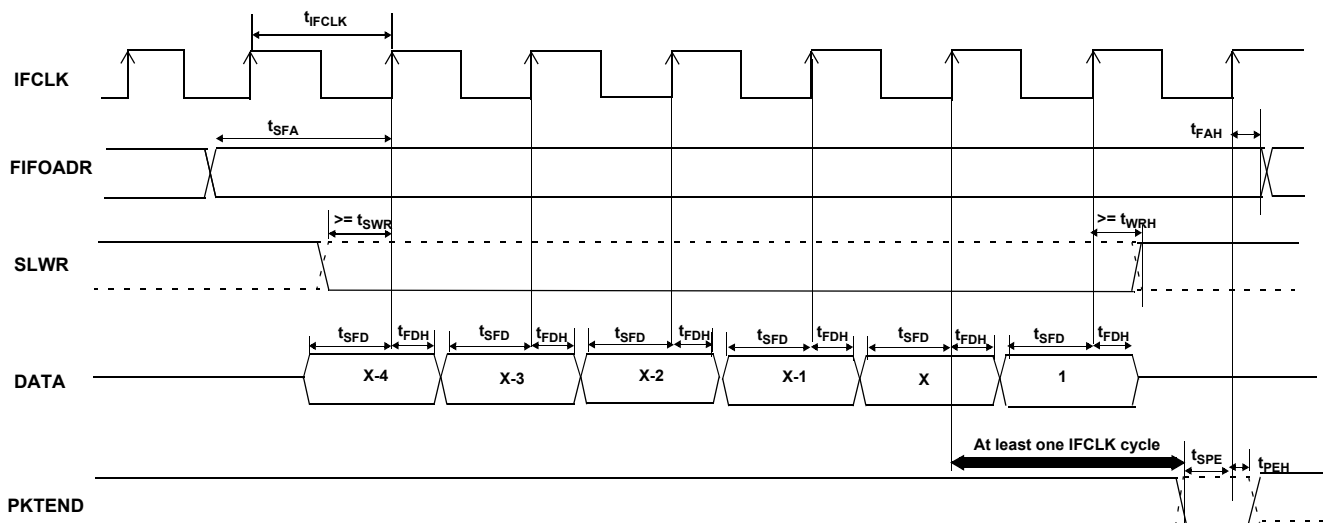
There is no specific timing requirement that needs to be met for asserting the PKTEND pin with regards to asserting SLWR. PKTEND can be asserted with the last data value clocked into the FIFO's or thereafter. The only consideration is that the setup time  $t_{SPE}$  and the hold time  $t_{PEH}$  must be met.

Although there are no specific timing requirements for the PKTEND assertion, there is a specific corner case condition that needs attention while using the PKTEND to commit a one byte/word packet. There is an additional timing requirement that needs to be met when the FIFO is configured to operate in auto mode and you want to send two packets back to back: a full packet (full defined as the number of bytes in the FIFO meeting the level set in AUTOINLEN register) committed automatically followed by a short one byte/word packet committed manually using the PKTEND pin. In this particular scenario, the user must make sure to assert PKTEND at least

one clock cycle after the rising edge that caused the last byte/word to be clocked into the previous auto committed packet. *Figure 9-7* shows this scenario. X is the value the AUTOINLEN register is set to when the IN endpoint is configured to be in auto mode.

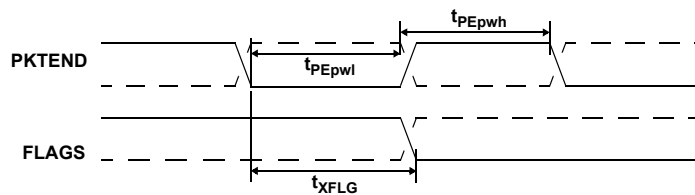
*Figure 9-7* shows a scenario where two packets are being committed. The first packet gets committed automatically when the number of bytes in the FIFO reaches X (value set in AUTOINLEN register) and the second one byte/word short packet is committed manually using PKTEND. Note that there is at least one IFCLK cycle timing between the assertion of PKTEND and clocking of the last byte of the previous packet (causing the packet to be committed automatically). Failing to adhere to this timing, results in the FX2LP18 failing to send the one byte/word short packet.

**Figure 9-7. Slave FIFO Synchronous Write Sequence and Timing Diagram<sup>[17]</sup>**



**9.8 Slave FIFO Asynchronous Packet End Strobe**

**Figure 9-8. Slave FIFO Asynchronous Packet End Strobe Timing Diagram<sup>[17]</sup>**



**Table 9-11. Slave FIFO Asynchronous Packet End Strobe Parameters<sup>[20]</sup>**

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		115	ns

### 9.9 Slave FIFO Output Enable

Figure 9-9. Slave FIFO Output Enable Timing Diagram<sup>[17]</sup>

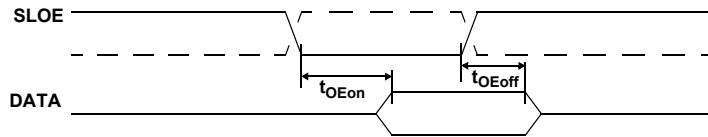


Table 9-12. Slave FIFO Output Enable Parameters

Parameter	Description	Min.	Max.	Unit
$t_{OEon}$	SLOE Assert to FIFO DATA Output		10.5	ns
$t_{OEoff}$	SLOE Deassert to FIFO DATA Hold	2.15	10.5	ns

### 9.10 Slave FIFO Address to Flags/Data

Figure 9-10. Slave FIFO Address to Flags/Data Timing Diagram<sup>[17]</sup>

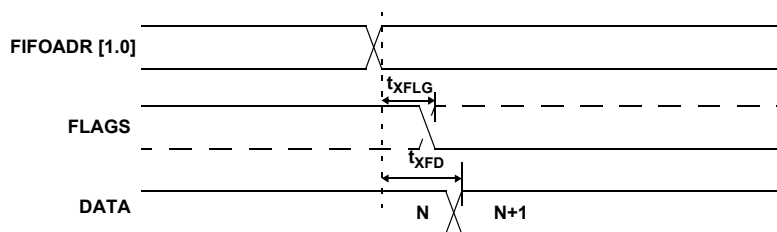


Table 9-13. Slave FIFO Address to Flags/Data Parameters

Parameter	Description	Min.	Max.	Unit
$t_{XFLG}$	FIFOADR[1:0] to FLAGS Output Propagation Delay		10.7	ns
$t_{XFD}$	FIFOADR[1:0] to FIFODATA Output Propagation Delay		14.3	ns

9.11 Slave FIFO Synchronous Address

Figure 9-11. Slave FIFO Synchronous Address Timing Diagram<sup>[17]</sup>

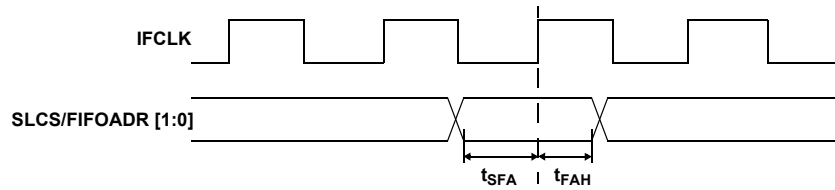
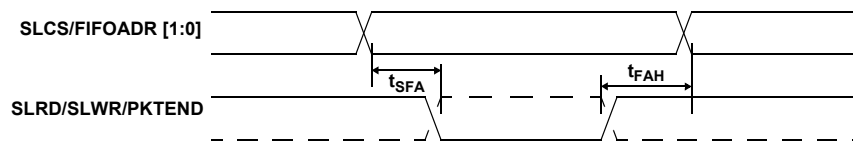


Table 9-14. Slave FIFO Synchronous Address Parameters<sup>[10]</sup>

Parameter	Description	Min.	Max.	Unit
$t_{IFCLK}$	Interface Clock Period	20.83	200	ns
$t_{SFA}$	FIFOADR[1:0] to Clock Set-up Time	25		ns
$t_{FAH}$	Clock to FIFOADR[1:0] Hold Time	10		ns

9.12 Slave FIFO Asynchronous Address

Figure 9-12. Slave FIFO Asynchronous Address Timing Diagram<sup>[17]</sup>



Slave FIFO Asynchronous Address Parameters<sup>[20]</sup>

Parameter	Description	Min.	Max.	Unit
$t_{SFA}$	FIFOADR[1:0] to SLRD/SLWR/PKTEND Set-up Time	10		ns
$t_{FAH}$	RD/WR/PKTEND to FIFOADR[1:0] Hold Time	10		ns



### 9.13 Sequence Diagram

Various sequence diagrams and examples are presented in this section.

#### 9.13.1 Single and Burst Synchronous Read Example

Figure 9-13. Slave FIFO Synchronous Read Sequence and Timing Diagram<sup>[17]</sup>

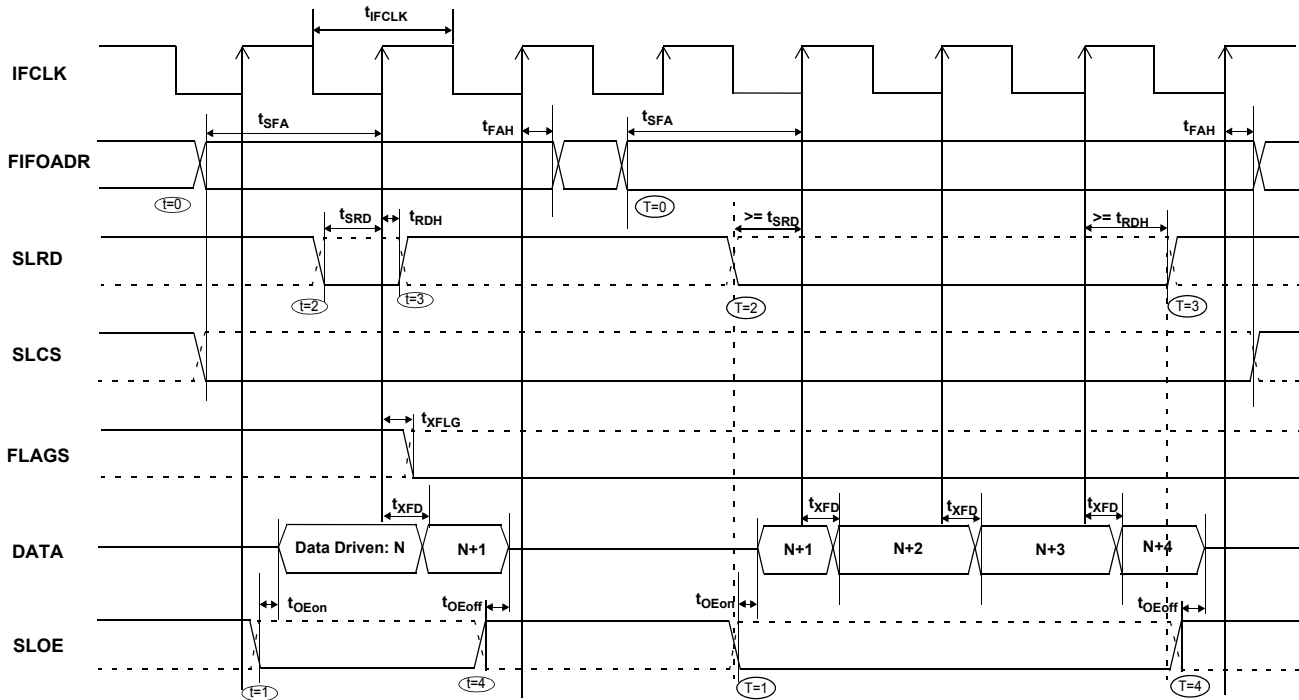


Figure 9-14. Slave FIFO Synchronous Sequence of Events Diagram

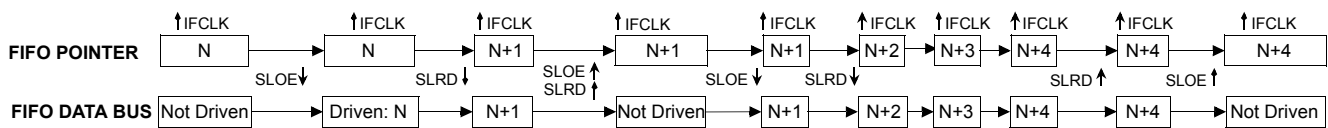


Figure 9-13 shows the timing relationship of the SLAVE FIFO signals during a synchronous FIFO read using IFCLK as the synchronizing clock. The diagram illustrates a single read followed by a burst read.

- At  $t = 0$  the FIFO address is stable and the signal SLCS is asserted (SLCS may be tied low in some applications). **Note**  $t_{SFA}$  has a minimum of 25 ns. This means that when IFCLK is running at 48 MHz, the FIFO address set-up time is more than one IFCLK cycle.
- At  $t = 1$ , SLOE is asserted. SLOE is an output enable only whose sole function is to drive the data bus. The data that is driven on the bus is the data that the internal FIFO pointer is currently pointing to. In this example it is the first data value in the FIFO. **Note** The data is pre-fetched and is driven on the bus when SLOE is asserted.
- At  $t = 2$ , SLRD is asserted. SLRD must meet the set-up time of  $t_{SRD}$  (time from asserting the SLRD signal to the rising edge of the IFCLK) and maintain a minimum hold time of  $t_{RDH}$  (time from the IFCLK edge to the deassertion of the SLRD signal). If the SLCS signal is used, it must be asserted

with SLRD, or before SLRD is asserted (for example, the SLCS and SLRD signals must both be asserted to start a valid read condition).

- The FIFO pointer is updated on the rising edge of the IFCLK while SLRD is asserted. This starts the propagation of data from the newly addressed location to the data bus. After a propagation delay of  $t_{XFD}$  (measured from the rising edge of IFCLK) the new data value is present. N is the first data value read from the FIFO. In order to have data on the FIFO data bus, SLOE MUST also be asserted.

The same sequence of events is shown for a burst read and is marked with the time indicators of T = 0 through 5. **Note** For the burst mode, the SLRD and SLOE are left asserted during the entire duration of the read. In the burst read mode, when SLOE is asserted, data indexed by the FIFO pointer is on the data bus. During the first read cycle on the rising edge of the clock, the FIFO pointer is updated and increments to point to address N+1. For each subsequent rising edge of IFCLK while the SLRD is asserted, the FIFO pointer is incremented and the next data value is placed on the data bus.

9.13.2 Single and Burst Synchronous Write

Figure 9-15. Slave FIFO Synchronous Write Sequence and Timing Diagram<sup>[17]</sup>

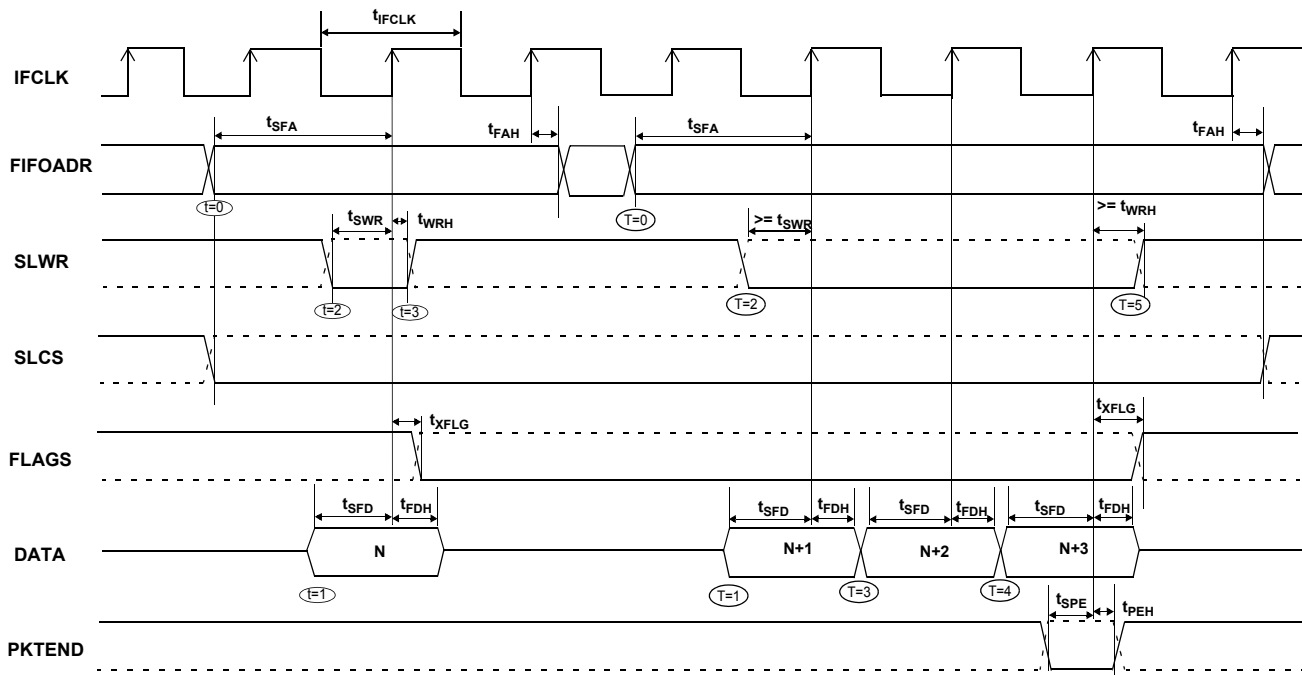


Figure 9-15 shows the timing relationship of the SLAVE FIFO signals during a synchronous write using IFCLK as the synchronizing clock. The diagram illustrates a single write followed by burst write of 3 bytes and committing all 4 bytes as a short packet using the PKTEND pin.

- At  $t = 0$  the FIFO address is stable and the signal SLCS is asserted. (SLCS may be tied low in some applications)  
**Note**  $t_{SFA}$  has a minimum of 25 ns. This means that when IFCLK is running at 48 MHz, the FIFO address set-up time is more than one IFCLK cycle.
- At  $t = 1$ , the external master/peripheral must output the data value onto the data bus with a minimum set-up time of  $t_{SFD}$  before the rising edge of IFCLK.
- At  $t = 2$ , SLWR is asserted. The SLWR must meet the set-up time of  $t_{SWR}$  (time from asserting the SLWR signal to the rising edge of IFCLK) and maintain a minimum hold time of  $t_{WRH}$  (time from the IFCLK edge to the deassertion of the SLWR signal). If the SLCS signal is used, it must be asserted with SLWR or before SLWR is asserted. (for example, the SLCS and SLWR signals must both be asserted to start a valid write condition).
- While the SLWR is asserted, data is written to the FIFO and on the rising edge of the IFCLK, the FIFO pointer is incremented. The FIFO flag is also updated after a delay of  $t_{XFLG}$  from the rising edge of the clock.

The same sequence of events is also shown for a burst write and is marked with the time indicators of  $T = 0$  through 5. **Note** For the burst mode, SLWR and SLCS are left asserted for the entire duration of writing all the required data values. 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 IFCLK. The FIFO pointer is updated on each rising edge of IFCLK. In Figure 9-15, once the four bytes are written to the FIFO, SLWR is deasserted. The short 4-byte packet can be committed to the host by asserting the PKTEND signal.

There is no specific timing requirement that needs to be met for asserting the PKTEND signal with regards to asserting the SLWR signal. PKTEND can be asserted with the last data value or thereafter. The only requirement is that the set-up time  $t_{SPE}$  and the hold time  $t_{PEH}$  must be met. In the scenario of Figure 9-15, the number of data values committed includes the last value written to the FIFO. In this example, both the data value and the PKTEND signal are clocked on the same rising edge of IFCLK. PKTEND can also be asserted in subsequent clock cycles. The FIFOADDR lines must be held constant during the PKTEND assertion.

Although there are no specific timing requirements for the PKTEND assertion, there is a specific corner case condition that needs attention while using the PKTEND to commit a one byte/word packet. Additional timing requirements exist when the FIFO is configured to operate in auto mode and you want to send two packets: a full packet (full defined as the number of bytes in the FIFO meeting the level set in AUTOINLEN register) committed automatically followed by a short one byte/word packet committed manually using the PKTEND pin. In this case, the external master must make sure to assert the PKTEND pin at least one clock cycle after the rising edge that caused the last byte/word to be clocked into the previous auto committed packet (the packet with the number of bytes equal to what is set in the AUTOINLEN register). Refer to Figure 9-7 for further details on this timing.

9.13.3 Sequence Diagram of a Single and Burst Asynchronous Read

Figure 9-16. Slave FIFO Asynchronous Read Sequence and Timing Diagram<sup>[17]</sup>

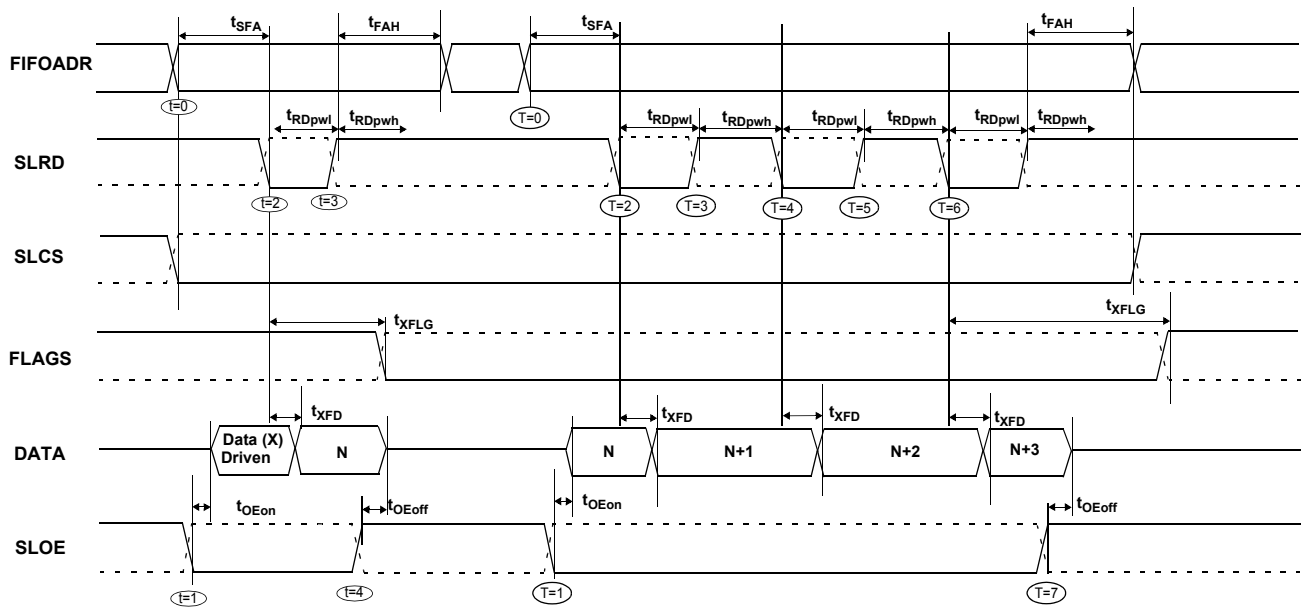


Figure 9-17. Slave FIFO Asynchronous Read Sequence of Events Diagram

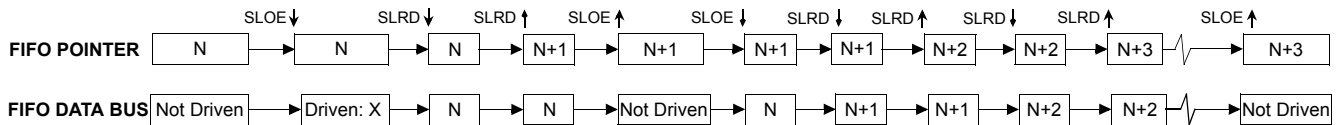


Figure 9-16 illustrates the timing relationship of the SLAVE FIFO signals during an asynchronous FIFO read. It shows a single read followed by a burst read.

- At  $t = 0$ , the FIFO address is stable and the SLCS signal is asserted.
- At  $t = 1$ , SLOE is asserted. This results in the data bus being driven. The data that is driven on to the bus is previous data; it is data that was in the FIFO from a prior read cycle.
- At  $t = 2$ , SLRD is asserted. 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 asserted with SLRD or before SLRD is asserted (for example, the SLCS and SLRD signals must both be asserted to start a valid read condition).

- The data that is driven, after asserting SLRD, is the updated data from the FIFO. This data is valid after a propagation delay of  $t_{XFD}$  from the activating edge of SLRD. In Figure 9-16, data N is the first valid data read from the FIFO. For data to appear on the data bus during the read cycle (for example, SLRD is asserted), SLOE MUST be in an asserted state. SLRD and SLOE can also be tied together.

The same sequence of events is also shown for a burst read marked with T = 0 through 5. **Note** In burst read mode, during SLOE assertion, the data bus is in a driven state and outputs the previous data. Once SLRD is asserted, the data from the FIFO is driven on the data bus (SLOE must also be asserted) and then the FIFO pointer is incremented.

9.13.4 Sequence Diagram of a Single and Burst Asynchronous Write

Figure 9-18. Slave FIFO Asynchronous Write Sequence and Timing Diagram<sup>[17]</sup>

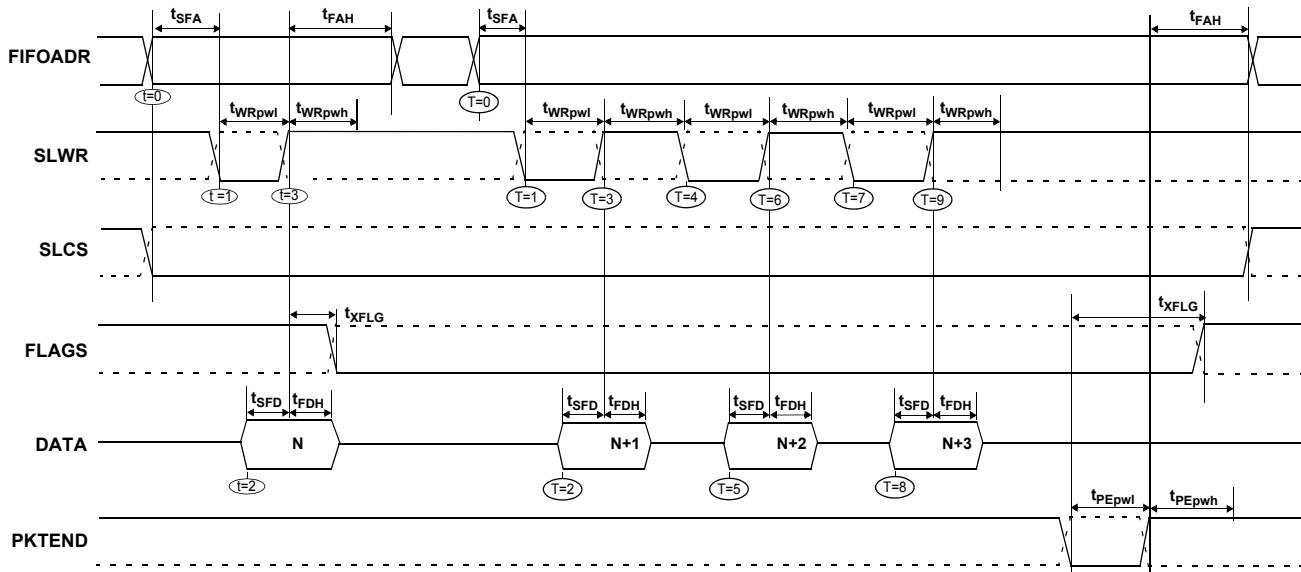


Figure 9-18 illustrates the timing relationship of the SLAVE FIFO write in an asynchronous mode. The diagram shows a single write followed by a burst write of 3 bytes and committing the 4-byte-short packet using PKTEND.

- At  $t = 0$  the FIFO address is applied, ensuring that it meets the set-up time of  $t_{SFA}$ . If SLCS is used, it must also be asserted (SLCS may be tied low in some applications).
- At  $t = 1$  SLWR is asserted. SLWR must meet the minimum active pulse of  $t_{WRpwl}$  and minimum de-active pulse width of  $t_{WRpwh}$ . If the SLCS is used, it must be asserted with SLWR or before SLWR is asserted.
- At  $t = 2$ , data must be present on the bus  $t_{SFD}$  before the deasserting edge of SLWR.
- At  $t = 3$ , deasserting SLWR causes the data to be written from the data bus to the FIFO and then the FIFO pointer is

incremented. The FIFO flag is also updated after  $t_{XFLG}$  from the deasserting edge of SLWR.

The same sequence of events is shown for a burst write and is indicated by the timing marks of  $T = 0$  through 5. **Note** In the burst write mode, once SLWR is deasserted, the data is written to the FIFO and then the FIFO pointer is incremented to the next byte in the FIFO. The FIFO pointer is post incremented.

In Figure 9-18 once the four bytes are written to the FIFO and SLWR is deasserted, the short 4-byte packet can be committed to the host using the PKTEND. The external device must be designed to not assert SLWR and the PKTEND signal at the same time. It must be designed to assert the PKTEND after SLWR is deasserted and meet the minimum deasserted pulse width. The FIFOADDR lines are to be held constant during the PKTEND assertion.

### 10.0 Ordering Information

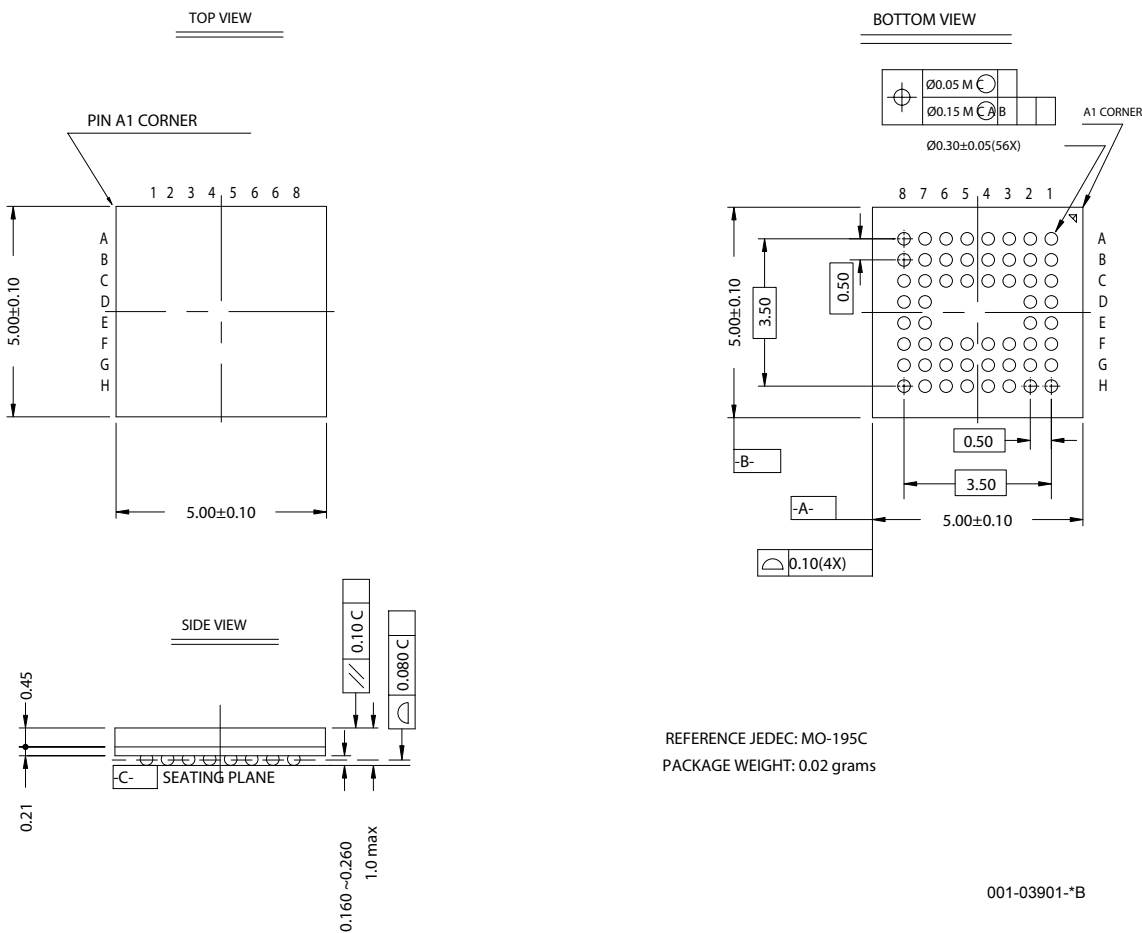
Table 10-1. Ordering Information

Ordering Code	Package Type	RAM Size	# Prog I/Os	8051 Address/Data Busses
CY7C68053-56BAXI	56 VFBGA– Lead-Free	16K	24	–
<b>Development Tool Kit</b>				
CY3687	MoBL-USB FX2LP18 Development Kit			

### 11.0 Package Diagram

The FX2LP18 is available in a 56-pin VFBGA package.

Figure 11-1. 56 VFBGA (5 x 5 x 1.0 mm) 0.50 Pitch, 0.30 Ball BZ56



## 12.0 PCB Layout Recommendations

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

- At least a four-layer impedance controlled board is 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 to within specifications.
- 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 must 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 preferable 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.

Purchase of I<sup>2</sup>C components from Cypress, or one of its sublicensed Associated Companies, conveys a license under the Philips I<sup>2</sup>C Patent Rights to use these components in an I<sup>2</sup>C system, provided that the system conforms to the I<sup>2</sup>C Standard Specification as defined by Philips. MoBL-USB FX2LP18, EZ-USB FX2LP and ReNumeration are trademarks, and MoBL-USB is a registered trademark, of Cypress Semiconductor Corporation. All product and company names mentioned in this document are the trademarks of their respective holders.

**Document History Page**

Document Title: CY7C68053 MoBL-USB FX2LP18 USB Microcontroller				
Document Number: 001-06120				
REV.	ECN NO.	Issue Date	Orig. of Change	Description of Change
**	430449	03/03/06	OSG	New data sheet
*A	434754	03/24/06	OSG	In Section 3.3, stated that SCL and SDA pins can be connected to $V_{CC}$ or $V_{CC\_IO}$ Changed sections 3.5, 3.18.1 and pin descriptions of SCL, SDA to indicate that since $DISCON=1$ after reset, an EEPROM or EEPROM emulation is required on the I <sup>2</sup> C interface In pin description table, renamed pin 2H (Reserved) to Ground In Section 6, added statement "The GPIO's are not over voltage tolerant, except the SCL and SDA pins, which are 3.3V tolerant" In Section 8, added a footnote to the DC char table stating that AVcc can be floated in low power mode In Section 8, changed $V_{IHmax}$ in DC char table from 3.6V to $V_{CC\_IO} + 10\%$
*B	465471	See ECN	OSG	Changed the recommendation for the pull up resistors on I <sup>2</sup> C Split Icc into 4 different values, corresponding to the different voltage supplies Changed I <sub>sus</sub> typical to 20uA and 220uA Added section 3.9.3 on suspend current considerations
*C	484726	See ECN	ARI	Removed all references the part number CY7C68055. Corrected the bullet in Features to state that 24 GPIO's are available. Added the Test ID (TID#) to the Features on the front page. Made changes to the block diagram on the first page (this is now a Visio drawing instead of a Framemaker drawing). Corrected the Ambient Temperature with Power Supplied. Moved figure titles to meet the new template. Checked grammar. Took out 9-bit address bus from the block diagram on the first page. Corrected Figure 4.1
*D	492009	See ECN	OSG	Added Icc data in DC Characteristics and Maximum Power dissipation
*E	500408	See ECN	OSG	Changed ESD spec to 1500V
*F	502115	See ECN	OSG	Changed ESD spec to 2000V and 1500V only for SCL and SDA pins. Added min spec for $t_{OEOff}$ Changed Icc and power dissipation numbers