# EZ-USB FX1™ USB Microcontroller
# Full-speed USB Peripheral Controller

## 1.0 Features

- **Single-chip integrated USB transceiver, SIE, and enhanced 8051 microprocessor**
- **Fit, form and function upgradable to the FX2LP (CY7C68013A)**
  - **Pin-compatible**
  - **Object-code-compatible**
  - **Functionally-compatible (FX1 functionality is a Subset of the FX2LP)**
- **Draws no more than 65 mA in any mode making the FX1 suitable for bus powered applications**
- **Software: 8051 runs from internal RAM, which is:**
  - **Downloaded via USB**
  - **Loaded from EEPROM**
  - **External memory device (128-pin configuration only)**
- **16 KBytes of on-chip Code/Data RAM**
- **Four programmable BULK/INTERRUPT/ISOCHRONOUS 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**
  - **Allows direct connection to most parallel interfaces; 8- and 16-bit**
  - **Programmable waveform descriptors and configuration registers to define waveforms**
  - **Supports multiple Ready (RDY) inputs and Control (CTL) outputs**
- **Integrated, industry standard 8051 with enhanced features**
  - **Up to 48-MHz clock rate**
  - **Four clocks per instruction cycle**
  - **Two USARTS**
  - **Three counter/timers**
  - **Expanded interrupt system**
  - **Two data pointers**
- **3.3V operation with 5V tolerant inputs**
- **Smart SIE**
- **Vectored USB interrupts**
- **Separate data buffers for the Setup and DATA portions of a CONTROL transfer**
- **Integrated $I^2C$ controller, runs at 100 or 400 KHz**
- **48-MHz, 24-MHz, or 12-MHz 8051 operation**
- **Four integrated FIFOs**
  - **Brings glue and FIFOs inside for lower system cost**
  - **Automatic conversion to and from 16-bit buses**
  - **Master or slave operation**
  - **FIFOs can use externally supplied clock or asynchronous strobes**
  - **Easy interface to ASIC and DSP ICs**
- **Vectored for FIFO and GPIF interrupts**
- **Up to 40 general purpose I/Os**
- **Three package options—128-pin TQFP, 100-pin TQFP, and 56-pin QFN Lead-free**

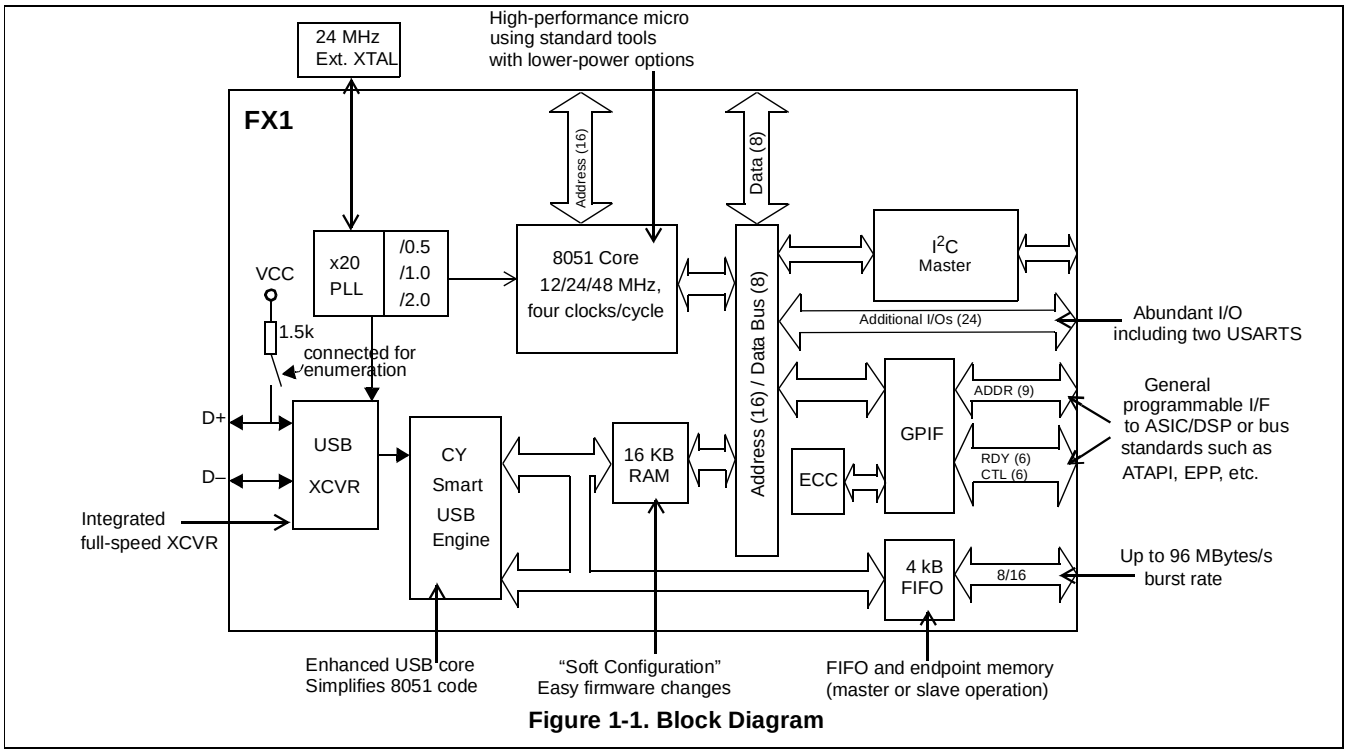



**Figure 1-1. Block Diagram**

## 2.0 Functional Description

EZ-USB FX1™ (CY7C64713/4) is a full-speed highly integrated, USB microcontroller. By integrating the USB 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.

Because it incorporates the USB transceiver, the EZ-USB FX1 is more economical, providing a smaller footprint solution than USB SIE or external transceiver implementations. With EZ-USB FX1, the Cypress Smart SIE handles most of the USB 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) provides an easy and glueless interface to popular interfaces such as ATA, UTOPIA, EPP, PCMCIA, and most DSP/processors.

Three lead-free packages are defined for the family: 56 QFN, 100 TQFP, and 128 TQFP.

## 3.0 Applications

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

The “Reference Designs” section of the cypress website provides additional tools for typical USB applications. Each reference design comes complete with firmware source and object code, schematics, and documentation. Please visit http://www.cypress.com for more information.

## 4.0 Functional Overview

### 4.1 USB Signaling Speed

FX1 operates at one 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.

FX1 does not support the low-speed signaling mode of 1.5 Mbps or the high-speed mode of 480 Mbps.

### 4.2 8051 Microprocessor

The 8051 microprocessor embedded in the FX1 family has 256 bytes of register RAM, an expanded interrupt system, three timer/counters, and two USARTs.

#### *4.2.1 8051 Clock Frequency*

FX1 has an on-chip oscillator circuit that uses an external 24-MHz (±100 ppm) crystal with the following characteristics:

- Parallel resonant
- Fundamental mode
- 500-∝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, and 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.

The CLKOUT pin, which can be three-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.

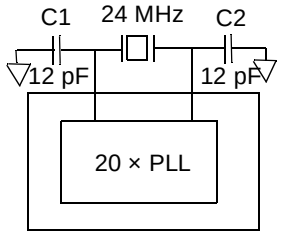
#### *4.2.2 USARTS*

FX1 contains two standard 8051 USARTs, addressed via Special Function Register (SFR) bits. The USART interface pins are available on separate I/O pins, and are not multiplexed with port pins.

UART0 and UART1 can operate using an internal clock at 230 KBaud with no more than 1% baud rate error. 230-KBaud operation is achieved by an internally derived clock source that generates overflow pulses at the appropriate time. The internal clock adjusts for the 8051 clock rate (48, 24, 12 MHz) such that it always presents the correct frequency for 230-KBaud operation.[1]

#### *4.2.3 Special Function Registers*

Certain 8051 SFR addresses are populated to provide fast access to critical FX1 functions. These SFR additions are shown in *Table 4-1*. Bold type indicates non-standard, enhanced 8051 registers. The two SFR rows that end with “0” and “8” contain bit-addressable registers. The four I/O ports A–D use the SFR addresses used in the standard 8051 for ports 0–3, which are not implemented in FX1. Because of the faster and more efficient SFR addressing, the FX1 I/O ports are not addressable in external RAM space (using the MOVX instruction).




12-pF capacitor values assumes a trace capacitance of 3 pF per side on a four-layer FR4 PCA

**Figure 4-1. Crystal Configuration**

**Note:**
1. 115-KBaud operation is also possible by programming the 8051 SMOD0 or SMOD1 bits to a “1” for UART0 and/or UART1, respectively.

**Table 4-1. 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** | **INT4CLR** | **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** | **AUTOPTRSETUP** | **GPIFSGLDATLNOX** | | | | |

## 4.3 I²C Bus

FX1 supports the I²C 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 3.3V, even if no I²C device is connected.

## 4.4 Buses

All packages: 8- or 16-bit “FIFO” bidirectional data bus, multiplexed on I/O ports B and D. 128-pin package: adds 16-bit output-only 8051 address bus, 8-bit bidirectional data bus.

## 4.5 USB Boot Methods

During the power-up sequence, internal logic checks the I²C port for the connection of an EEPROM whose first byte is either 0xC0 or 0xC2. If found, it uses the VID/PID/DID values in the EEPROM in place of the internally stored values (0xC0), or it boot-loads the EEPROM contents into internal RAM (0xC2). If no EEPROM is detected, FX1 enumerates using internally stored descriptors. The default ID values for FX1 are VID/PID/DID (0x04B4, 0x6473, 0xAxxx where xxx=Chip revision).[2]

**Table 4-2. Default ID Values for FX1**

| Default VID/PID/DID | | |
|---|---|---|
| Vendor ID | 0x04B4 | Cypress Semiconductor |
| Product ID | 0x6473 | EZ-USB FX1 |
| Device release | 0xAnnn | Depends chip revision (nnn = chip revision where first silicon = 001) |

## 4.6 ReNumeration™

Because the FX1’s configuration is soft, one chip can take on the identities of multiple distinct USB devices.

When first plugged into USB, the FX1 enumerates automatically and downloads firmware and USB descriptor tables over the USB cable. Next, the FX1 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 will handle device requests over endpoint zero: if RENUM = 0, the Default USB Device will handle device requests; if RENUM = 1, the firmware will.

## 4.7 Bus-powered Applications

The FX1 fully supports bus-powered designs by enumerating with less than 100 mA as required by the USB specification.

## 4.8 Interrupt System

### *4.8.1 INT2 Interrupt Request and Enable Registers*

FX1 implements an autovector feature for INT2 and INT4. There are 27 INT2 (USB) vectors, and 14 INT4 (FIFO/GPIF) vectors. See EZ-USB Technical Reference Manual (TRM) for more details.

**Note:**

2. The I²C bus SCL and SDA pins must be pulled up, even if an EEPROM is not connected. Otherwise this detection method does not work properly.

### 4.8.2 USB-Interrupt Autovectors

The main USB interrupt is shared by 27 interrupt sources. To save the code and processing time that normally would be required to identify the individual USB interrupt source, the FX1 provides a second level of interrupt vectoring, called Autovectoring. When a USB interrupt is asserted, the FX1 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 FX1 jump instruction is encoded as shown in *Table 4-3*.

If Autovectoring is enabled (AV2EN = 1 in the INTSETUP register), the FX1 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 will direct the jump to the correct address out of the 27 addresses within the page.

### 4.8.3 FIFO/GPIF Interrupt (INT4)

Just as the USB Interrupt is shared among 27 individual USB-interrupt sources, the FIFO/GPIF interrupt is shared among 14 individual FIFO/GPIF sources. The FIFO/GPIF Interrupt, like the USB Interrupt, can employ autovectoring. *Table 4-4* shows the priority and INT4VEC values for the 14 FIFO/GPIF interrupt sources.

**Table 4-3. INT2 USB Interrupts**

| USB INTERRUPT TABLE FOR INT2 | | | |
|---|---|---|---|
| **Priority** | **INT2VEC Value** | **Source** | **Notes** |
| 1 | 00 | SUDAV | Setup Data Available |
| 2 | 04 | SOF | Start of Frame |
| 3 | 08 | SUTOK | Setup Token Received |
| 4 | 0C | SUSPEND | USB Suspend request |
| 5 | 10 | USB RESET | Bus reset |
| 6 | 14 | | reserved |
| 7 | 18 | EP0ACK | FX1 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 |

**Table 4-4. Individual FIFO/GPIF Interrupt Sources**

| Priority | INT4VEC Value | Source | Notes |
|---|---|---|---|
| 1 | 80 | EP2PF | Endpoint 2 Programmable Flag |
| 2 | 84 | EP4PF | Endpoint 4 Programmable Flag |
| 3 | 88 | EP6PF | Endpoint 6 Programmable Flag |
| 4 | 8C | EP8PF | Endpoint 8 Programmable Flag |
| 5 | 90 | EP2EF | Endpoint 2 Empty Flag |
| 6 | 94 | EP4EF | Endpoint 4 Empty Flag |
| 7 | 98 | EP6EF | Endpoint 6 Empty Flag |
| 8 | 9C | EP8EF | Endpoint 8 Empty Flag |
| 9 | A0 | EP2FF | Endpoint 2 Full Flag |
| 10 | A4 | EP4FF | Endpoint 4 Full Flag |
| 11 | A8 | EP6FF | Endpoint 6 Full Flag |
| 12 | AC | EP8FF | Endpoint 8 Full Flag |
| 13 | B0 | GPIFDONE | GPIF Operation Complete |
| 14 | B4 | GPIFWF | GPIF Waveform |

If Autovectoring is enabled (AV4EN = 1 in the INTSETUP register), the FX1 substitutes its INT4VEC byte. Therefore, if the high byte (“page”) of a jump-table address is preloaded at location 0x0054, the automatically-inserted INT4VEC byte at 0x0055 will direct the jump to the correct address out of the 14 addresses within the page. When the ISR occurs, the FX1 pushes the program counter onto its stack then jumps to address 0x0053, where it expects to find a “jump” instruction to the ISR Interrupt service routine.

## 4.9 Reset and Wakeup

### *4.9.1 Reset Pin*

The input pin, RESET#, will reset the FX1 when asserted. This pin has hysteresis and is active LOW. When a crystal is used with the CY7C64713/4 the reset period must allow for the stabilization of the crystal and the PLL. This reset period should be approximately 5 ms after VCC has reached 3.0 Volts. If the crystal input pin is driven by a clock signal the internal PLL stabilizes in 200 ∝s after VCC has reached 3.0V[3]. *Figure 4-2* 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 FX1 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. While the application note discusses the FX2, the information provided applies also to the FX1. For more information on reset implementation for the FX2 family of products visit the http://www.cypress.com.

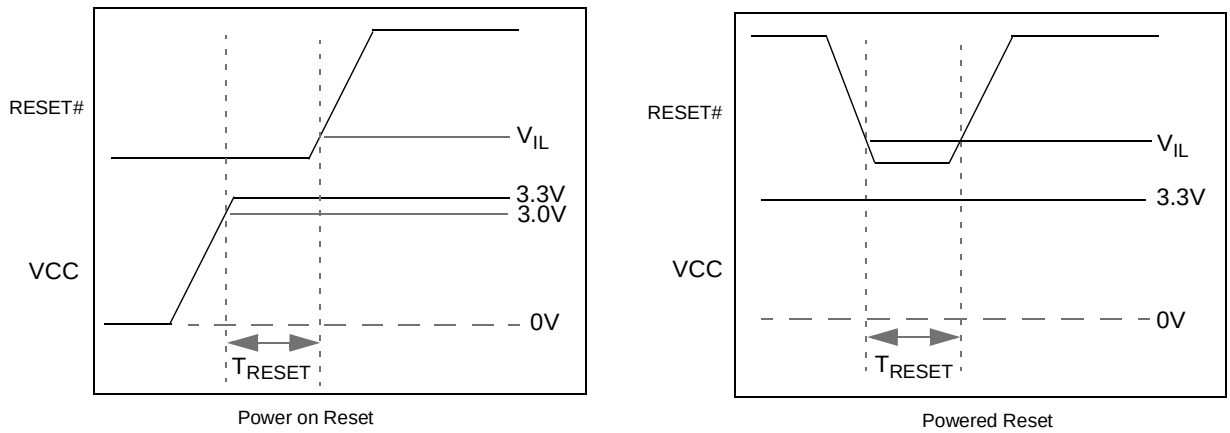



**Figure 4-2. Reset Timing Plots**

**Note:**
3. If the external clock is powered at the same time as the CY7C64713/4 and has a stabilization wait period, it must be added to the 200 ∝s.

**Table 4-5. Reset Timing Values**

| Condition | $T_{RESET}$ |
|---|---|
| Power-On Reset with crystal | 5 ms |
| Power-On Reset with external clock | 200 ∝s + Clock stability time |
| Powered Reset | 200 ∝s |

*4.9.2 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 FX1 is connected to the USB.

The FX1 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 FX1 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 I/O 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.

## 4.10 Program/Data RAM

*4.10.1 Size*

The FX1 has 16 KBytes of internal program/data RAM, where PSEN#/RD# signals are internally ORed to allow the 8051 to access it as both program and data memory. No USB control registers appear in this space.

Two memory maps are shown in the following diagrams:

*Figure 4-3* Internal Code Memory, EA = 0

*Figure 4-4* External Code Memory, EA = 1.

*4.10.2 Internal Code Memory, EA = 0*

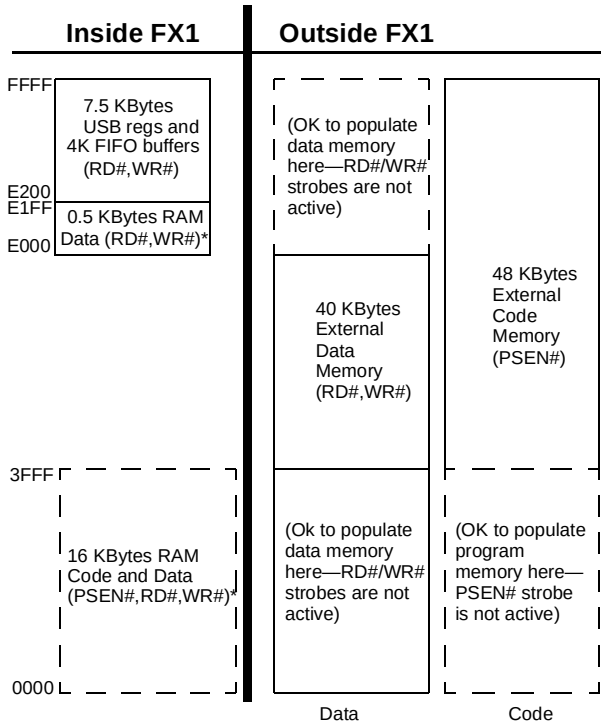
This mode implements the internal 16-KByte block of RAM (starting at 0) as combined code and data memory. When external RAM or ROM is added, the external read and write strobes are suppressed for memory spaces that exist inside the chip. This allows the user to connect a 64-KByte memory without requiring address decodes to keep clear of internal memory spaces.

Only the **internal** 16 KBytes and **scratch pad** 0.5 KBytes RAM spaces have the following access:

- USB download
- USB upload
- Setup data pointer
- $I^2C$ interface boot load.

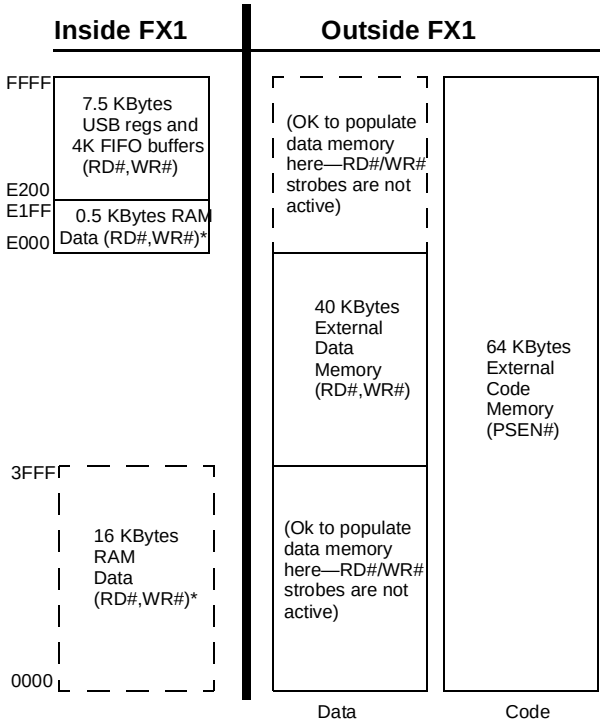
*4.10.3 External Code Memory, EA = 1*

The bottom 16 KBytes of program memory is external, and therefore the bottom 16 KBytes of internal RAM is accessible only as data memory.




*SUDPTR, USB upload/download, $I^2C$ interface boot access

**Figure 4-3. Internal Code Memory, EA = 0**




*SUDPTR, USB upload/download, $I^2C$ interface boot access

**Figure 4-4. External Code Memory, EA = 1**

## 4.11 Register Addresses

| | |
|---|---|
| FFFF<br>F000 | 4 KBytes EP2-EP8 buffers<br>(8 x 512)<br>Not all Space is available for all transfer types |
| EFFF<br>E800 | 2 KBytes RESERVED |
| E7FF<br>E7C0 | 64 Bytes EP1IN |
| E7BF<br>E780 | 64 Bytes EP1OUT |
| E77F<br>E740 | 64 Bytes EP0 IN/OUT |
| E73F<br>E700 | 64 Bytes RESERVED |
| E6FF<br>E500 | 8051 Addressable Registers (512) |
| E4FF<br>E480 | Reserved (128) |
| E47F<br>E400 | 128 bytes GPIF Waveforms |
| E3FF<br>E200 | Reserved (512) |
| E1FF<br>E000 | 512 bytes<br>8051 xdata RAM |

## 4.12 Endpoint RAM

### 4.12.1 Size

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

### 4.12.2 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, of which only the transfer size is available. EP4 and EP8 can be double buffered, while EP2 and 6 can be either double, triple, or quad buffered. Regardless of the physical size of the buffer, each endpoint buffer accommodates only one full-speed packet. For bulk endpoints the maximum number of bytes it can accommodate is 64, even though the physical buffer size is 512 or 1024. For an ISOCHRONOUS endpoint the maximum number of bytes it can accommodate is 1023. For endpoint configuration options, see *Figure 4-5*.

### 4.12.3 Setup Data Buffer

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

### 4.12.4 Endpoint Configurations

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 1 of the 12 configurations shown in the vertical columns. In full-speed, BULK mode uses only the first 64 bytes of each buffer, even though memory exists for the allocation of the isochronous transfers in BULK mode the unused endpoint buffer space is not available for other operations. An example endpoint configuration would be:

EP2—1023 double buffered; EP6—64 quad buffered (column 8).

### 4.12.5 Default Alternate Settings

**Table 4-6. Default Alternate Settings**[4, 5]

| 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×) |

## 4.13 External FIFO Interface

### 4.13.1 Architecture

The FX1 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). The usable size of these buffers depend on the USB transfer mode as described in Section 4.12.2.

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

### 4.13.2 Master/Slave Control Signals

The FX1 endpoint FIFOS 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-I/O Unit domain. This switching is done virtually

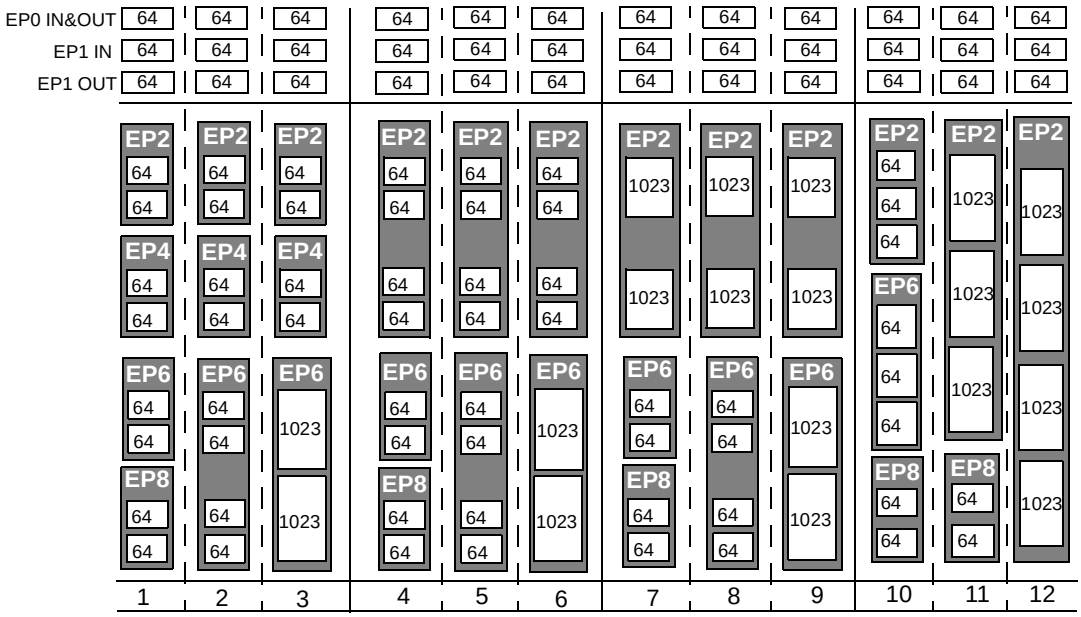



**Figure 4-5. Endpoint Configuration**

**Notes:**
4. "0" means "not implemented."
5. "2×" means "double buffered."

instantaneously, giving essentially zero transfer time between “USB FIFOS” and “Slave FIFOS.” Since they are physically the same memory, no bytes are actually transferred between buffers.

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

The I/O 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 RDY pins (two in the 56-pin package, six in the 100-pin and 128-pin packages) can be used as flag inputs from an external FIFO or other logic if desired. 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 FX1 accepts either an internally derived clock or externally supplied clock (IFCLK, max. 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#.

#### *4.13.3 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 FIFOs are internally clocked. An output enable bit in the IFCONFIG register turns this clock output off, if desired. Another bit within the IFCONFIG register will invert the IFCLK signal whether internally or externally sourced.

### 4.14 GPIF

The GPIF is a flexible 8- or 16-bit parallel interface driven by a user-programmable finite state machine. It allows the CY7C64713/4 to perform local bus mastering, and can implement a wide variety of protocols such as ATA interface, printer parallel port, and Utopia.

The GPIF has six programmable control outputs (CTL), nine address outputs (GPIFADRx), and six 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, etc. A sequence of the GPIF vectors make up a single waveform that will be executed to perform the desired data move between the FX1 and the external device.

#### *4.14.1 Six Control OUT Signals*

The 100- and 128-pin packages bring out all six Control Output pins (CTL0-CTL5). The 8051 programs the GPIF unit to define the CTL waveforms. The 56-pin package brings out three of these signals, CTL0–CTL2. CTLx waveform edges can be programmed to make transitions as fast as once per clock (20.8 ns using a 48-MHz clock).

#### *4.14.2 Six Ready IN Signals*

The 100- and 128-pin packages bring out all six Ready inputs (RDY0–RDY5). The 8051 programs the GPIF unit to test the RDY pins for GPIF branching. The 56-pin package brings out two of these signals, RDY0–1.

#### *4.14.3 Nine GPIF Address OUT Signals*

Nine GPIF address lines are available in the 100- and 128-pin packages, GPIFADR[8..0]. The GPIF address lines allow indexing through up to a 512-byte block of RAM. If more address lines are needed, I/O port pins can be used.

#### *4.14.4 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.

### 4.15 ECC Generation

The EZ-USB FX1 can calculate ECCs (Error-Correcting Codes) on data that passes across its GPIF or Slave FIFO interfaces. There are two ECC configurations: Two ECCs, each calculated over 256 bytes (SmartMedia™ Standard); and one ECC calculated over 512 bytes.

The ECC can correct any one-bit error or detect any two-bit error.

**Note:** To use the ECC logic, the GPIF or Slave FIFO interface must be configured for byte-wide operation.

#### *4.15.1 ECC Implementation*

The two ECC configurations are selected by the ECCM bit:

##### *4.15.1.1 ECCM = 0*

Two 3-byte ECCs, each calculated over a 256-byte block of data. This configuration conforms to the SmartMedia Standard.

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

##### *4.15.1.2 ECCM = 1*

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

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

### 4.16 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).[6]

### 4.17 Autopointer Access

FX1 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. This capability is available to and from both internal and external RAM. The autopointers are available in external FX1 registers, under control of a mode bit (AUTOPTRSETUP.0). Using the external FX1 autopointer access (at 0xE67B – 0xE67C) allows the autopointer to access all RAM, internal and external to the part. Also, the autopointers can point to any FX1 register or endpoint buffer space. When autopointer access to external memory is enabled, location 0xE67B and 0xE67C in XDATA and code space cannot be used.

### 4.18 $I^2C$ Controller

FX1 has one $I^2C$ port that is driven by two internal controllers, one that automatically operates at boot time to load VID/PID/DID and configuration information, and another that the 8051, once running, uses to control external $I^2C$ devices. The $I^2C$ port operates in master mode only.

#### *4.18.1 $I^2C$ Port Pins*

The $I^2C$ pins SCL and SDA must have external 2.2-kΩ pull-up resistors even if no EEPROM is connected to the FX1. External EEPROM device address pins must be configured properly. See *Table 4-7* for configuring the device address pins.

**Table 4-7. Strap Boot EEPROM Address Lines to These Values**

| Bytes | Example EEPROM | A2 | A1 | A0 |
|---|---|---|---|---|
| 16 | 24LC00[7] | N/A | N/A | N/A |
| 128 | 24LC01 | 0 | 0 | 0 |
| 256 | 24LC02 | 0 | 0 | 0 |
| 4K | 24LC32 | 0 | 0 | 1 |
| 8K | 24LC64 | 0 | 0 | 1 |
| 16K | 24LC128 | 0 | 0 | 1 |

#### *4.18.2 $I^2C$ Interface Boot Load Access*

At power-on reset the $I^2C$ interface boot loader will load the VID/PID/DID 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 will be in reset. $I^2C$ interface boot loads only occur after power-on reset.

#### *4.18.3 $I^2C$ Interface General Purpose Access*

The 8051 can control peripherals connected to the $I^2C$ bus using the I2CTL and I2DAT registers. FX1 provides $I^2C$ master control only, it is never an $I^2C$ slave.

### 4.19 Compatible with Previous Generation EZ-USB FX2

The EZ-USB FX1 is fit/form/function-upgradable to the EZ-USB FX2LP. This makes for a easy transition for designers wanting to upgrade their systems from full-speed to the high-speed designs. The pinout and package selection are identical, and all of the firmware developed for the FX1 will function in the FX2LP with proper addition of High Speed descriptors and speed switching code.

## 5.0 Pin Assignments

*Figure 5-1* identifies all signals for the three package types. The following pages illustrate the individual pin diagrams, plus a combination diagram showing which of the full set of signals are available in the 128-, 100-, and 56-pin packages.

The signals on the left edge of the 56-pin package in *Figure 5-1* are common to all versions in the FX1 family. Three modes are available in all package versions: 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.

The 100-pin package adds functionality to the 56-pin package by adding these pins:

- PORTC or alternate GPIFADR[7:0] address signals
- PORTE or alternate GPIFADR[8] address signal and seven additional 8051 signals
- Three GPIF Control signals
- Four GPIF Ready signals
- Nine 8051 signals (two USARTs, three timer inputs, INT4,and INT5#)
- BKPT, RD#, WR#.

The 128-pin package adds the 8051 address and data buses plus control signals. Note that two of the required signals, RD# and WR#, are present in the 100-pin version. In the 100-pin and 128-pin versions, an 8051 control bit can be set to pulse the RD# and WR# pins when the 8051 reads from/writes to PORTC.

**Notes:**
6. 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.
7. This EEPROM does not have address pins.

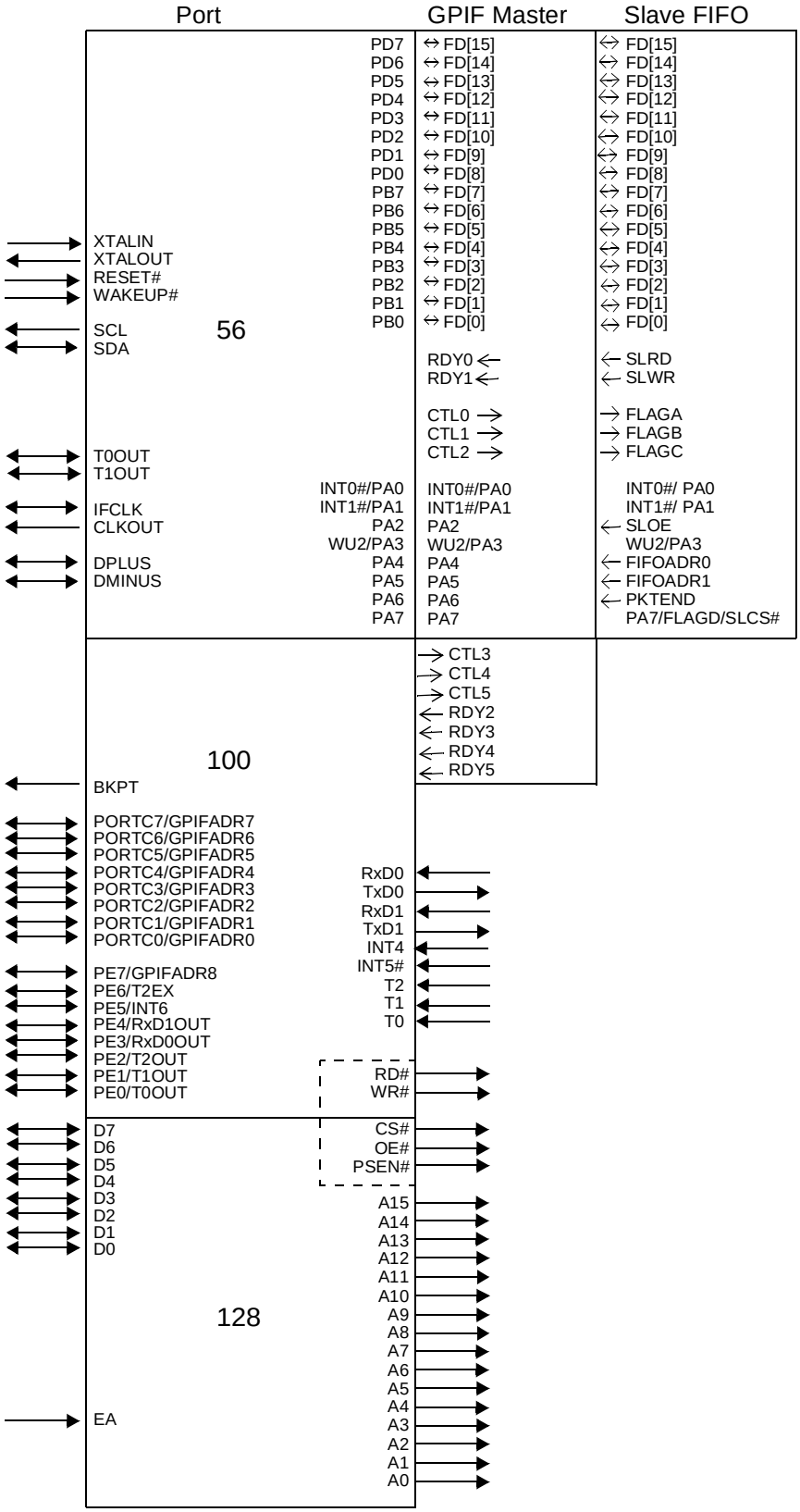



**Figure 5-1. Signals**

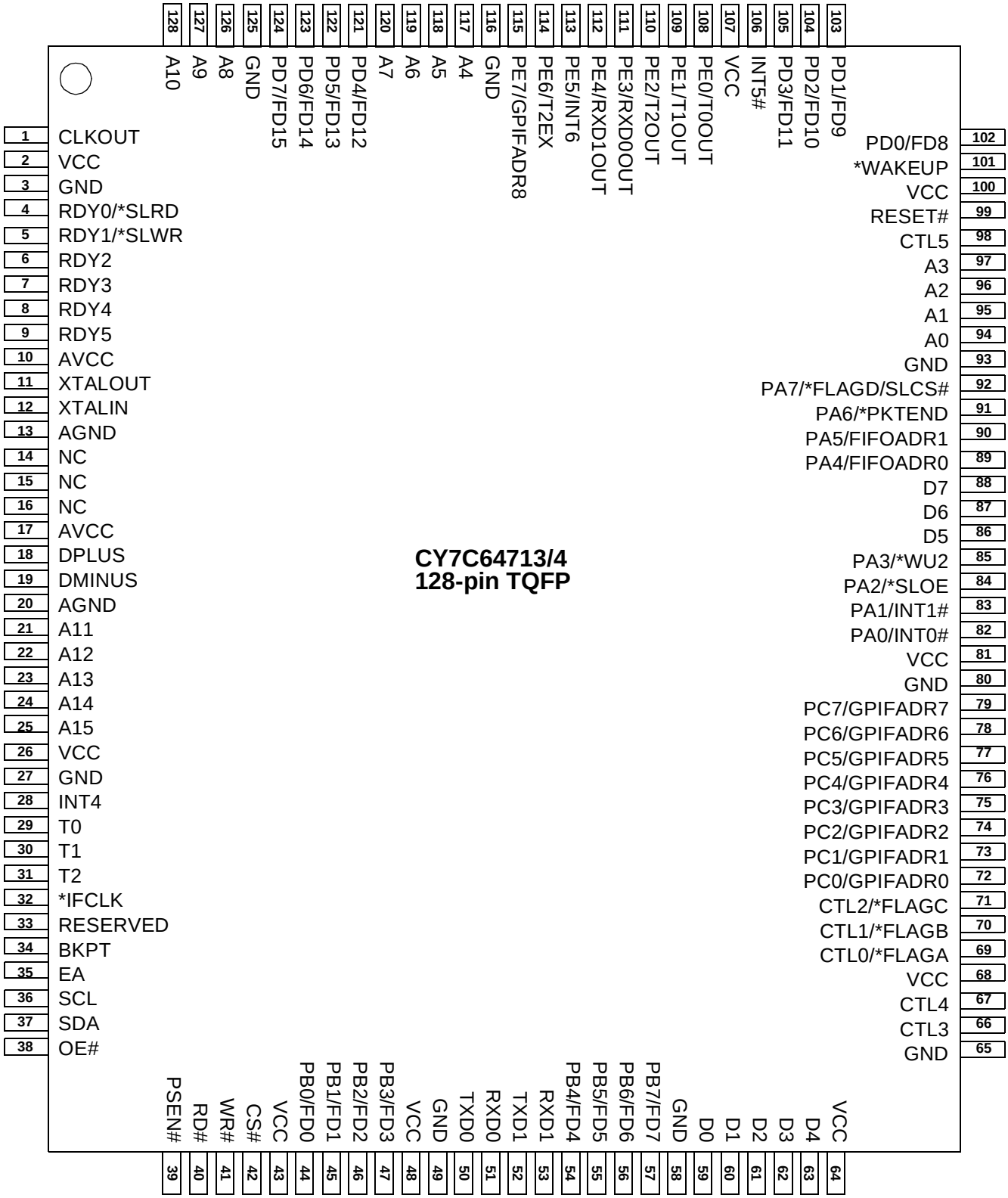



**Figure 5-2. CY7C64713/4 128-pin TQFP Pin Assignment**

* denotes programmable polarity

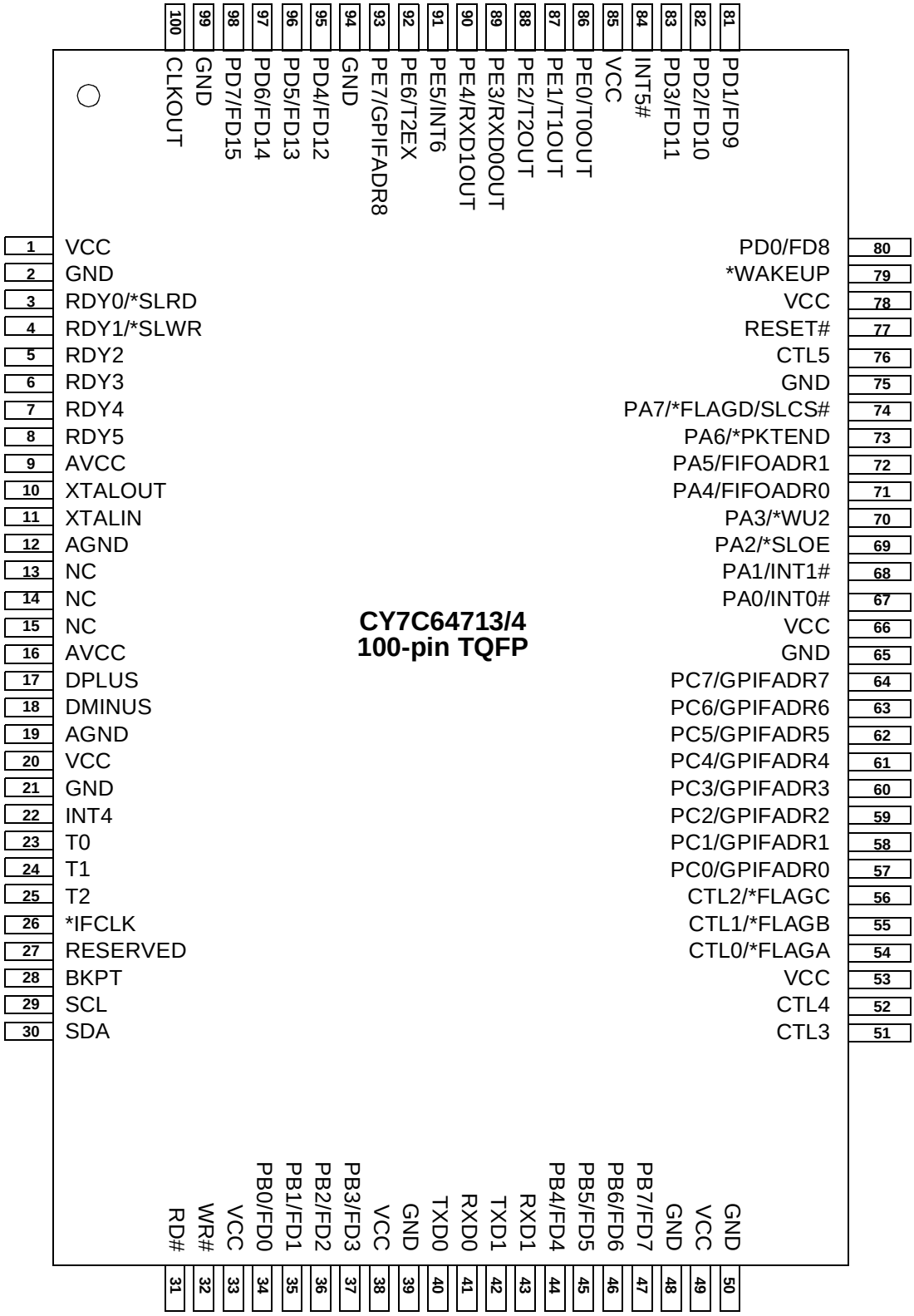



**Figure 5-3. CY7C64713/4 100-pin TQFP Pin Assignment**

* denotes programmable polarity

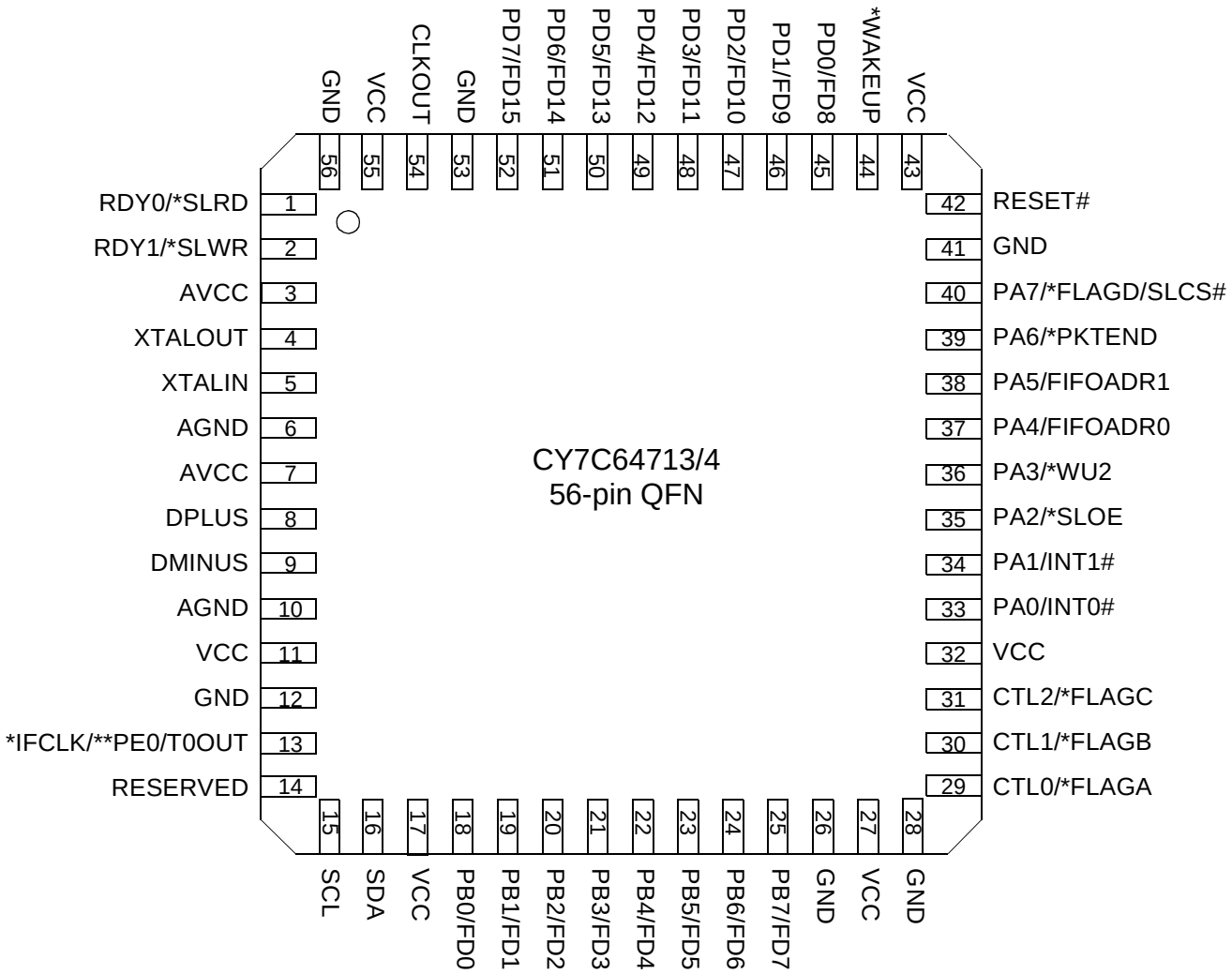



**Figure 5-4. CY7C64713/4 56-pin QFN Pin Assignment**

* denotes programmable polarity

## 5.1 CY7C64713/4 Pin Definitions

**Table 5-1. FX1 Pin Definitions** [8]

| 128 TQFP | 100 TQFP | 56 QFN | Name | Type | Default | Description |
|---|---|---|---|---|---|---|
| 10 | 9 | 3 | AVCC | Power | N/A | **Analog VCC**. Connect this pin to 3.3V power source. This signal provides power to the analog section of the chip. |
| 17 | 16 | 7 | AVCC | Power | N/A | **Analog VCC**. Connect this pin to 3.3V power source. This signal provides power to the analog section of the chip. |
| 13 | 12 | 6 | AGND | Ground | N/A | **Analog Ground**. Connect to ground with as short a path as possible. |
| 20 | 19 | 10 | AGND | Ground | N/A | **Analog Ground**. Connect to ground with as short a path as possible. |
| 19 | 18 | 9 | DMINUS | I/O/Z | Z | **USB D– Signal**. Connect to the USB D– signal. |
| 18 | 17 | 8 | DPLUS | I/O/Z | Z | **USB D+ Signal**. Connect to the USB D+ signal. |
| 94 | | | A0 | Output | L | **8051 Address Bus**. This bus is driven at all times. When the 8051 is addressing internal RAM it reflects the internal address. |
| 95 | | | A1 | Output | L | |
| 96 | | | A2 | Output | L | |
| 97 | | | A3 | Output | L | |
| 117 | | | A4 | Output | L | |
| 118 | | | A5 | Output | L | |
| 119 | | | A6 | Output | L | |
| 120 | | | A7 | Output | L | |
| 126 | | | A8 | Output | L | |
| 127 | | | A9 | Output | L | |
| 128 | | | A10 | Output | L | |
| 21 | | | A11 | Output | L | |
| 22 | | | A12 | Output | L | |
| 23 | | | A13 | Output | L | |
| 24 | | | A14 | Output | L | |
| 25 | | | A15 | Output | L | |
| 59 | | | D0 | I/O/Z | Z | **8051 Data Bus**. This bidirectional bus is high-impedance when inactive, input for bus reads, and output for bus writes. The data bus is used for external 8051 program and data memory. The data bus is active only for external bus accesses, and is driven LOW in suspend. |
| 60 | | | D1 | I/O/Z | Z | |
| 61 | | | D2 | I/O/Z | Z | |
| 62 | | | D3 | I/O/Z | Z | |
| 63 | | | D4 | I/O/Z | Z | |
| 86 | | | D5 | I/O/Z | Z | |
| 87 | | | D6 | I/O/Z | Z | |
| 88 | | | D7 | I/O/Z | Z | |
| 39 | | | PSEN# | Output | H | **Program Store Enable**. This active-LOW signal indicates an 8051 code fetch from external memory. It is active for program memory fetches from 0x4000–0xFFFF when the EA pin is LOW, or from 0x0000–0xFFFF when the EA pin is HIGH. |
| 34 | 28 | | BKPT | Output | L | **Breakpoint**. This pin goes active (HIGH) when the 8051 address bus matches the BPADDRH/L registers and breakpoints are enabled in the BREAKPT register (BPEN = 1). If the BPPULSE bit in the BREAKPT register is HIGH, this signal pulses HIGH for eight 12-/24-/48-MHz clocks. If the BPPULSE bit is LOW, the signal remains HIGH until the 8051 clears the BREAK bit (by writing 1 to it) in the BREAKPT register. |

**Note:**

8. Unused inputs should 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 5-1. FX1 Pin Definitions** (continued)[8]

| 128 TQFP | 100 TQFP | 56 QFN | Name | Type | Default | Description |
|---|---|---|---|---|---|---|
| 99 | 77 | 42 | RESET# | Input | N/A | **Active LOW Reset**. Resets the entire chip. See section 4.9 ”Reset and Wakeup” on page 5 for more details. |
| 35 | | | EA | Input | N/A | **External Access**. This pin determines where the 8051 fetches code between addresses 0x0000 and 0x3FFF. If EA = 0 the 8051 fetches this code from its internal RAM. IF EA = 1 the 8051 fetches this code from external memory. |
| 12 | 11 | 5 | XTALIN | Input | N/A | **Crystal Input**. Connect this signal to a 24-MHz parallel-resonant, fundamental mode crystal and load capacitor to GND.<br>It is also correct to drive XTALIN with an external 24 MHz square wave derived from another clock source. When driving from an external source, the driving signal should be a 3.3V square wave. |
| 11 | 10 | 4 | XTALOUT | Output | N/A | **Crystal Output**. Connect this signal to a 24-MHz parallel-resonant, fundamental mode crystal and load capacitor to GND.<br>If an external clock is used to drive XTALIN, leave this pin open. |
| 1 | 100 | 54 | CLKOUT | O/Z | 12 MHz | **CLKOUT:** 12-, 24- or 48-MHz clock, phase locked to the 24-MHz input clock. The 8051 defaults to 12-MHz operation. The 8051 may three-state this output by setting CPUCS.1 = 1. |
| **Port A** | | | | | | |
| 82 | 67 | 33 | PA0 or INT0# | I/O/Z | I (PA0) | Multiplexed pin whose function is selected by PORTACFG.0<br>**PA0** is a bidirectional IO port pin.<br>**INT0#** is the active-LOW 8051 INT0 interrupt input signal, which is either edge triggered (IT0 = 1) or level triggered (IT0 = 0). |
| 83 | 68 | 34 | PA1 or INT1# | I/O/Z | I (PA1) | Multiplexed pin whose function is selected by:<br>PORTACFG.1<br>**PA1** is a bidirectional IO port pin.<br>**INT1#** is the active-LOW 8051 INT1 interrupt input signal, which is either edge triggered (IT1 = 1) or level triggered (IT1 = 0). |
| 84 | 69 | 35 | PA2 or SLOE | I/O/Z | I (PA2) | Multiplexed pin whose function is selected by two bits:<br>IFCONFIG[1:0].<br>**PA2** is a bidirectional IO port pin.<br>**SLOE** is an input-only output enable with programmable polarity (FIFOPINPOLAR.4) for the slave FIFOs connected to FD[7..0] or FD[15..0]. |
| 85 | 70 | 36 | PA3 or WU2 | I/O/Z | I (PA3) | Multiplexed pin whose function is selected by:<br>WAKEUP.7 and OEA.3<br>**PA3** is a bidirectional I/O port pin.<br>**WU2** is an alternate source for **USB Wakeup,** 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. |
| 89 | 71 | 37 | PA4 or FIFOADR0 | I/O/Z | I (PA4) | Multiplexed pin whose function is selected by:<br>IFCONFIG[1..0].<br>**PA4** is a bidirectional I/O port pin.<br>**FIFOADR0** is an input-only address select for the slave FIFOs connected to FD[7..0] or FD[15..0]. |
| 90 | 72 | 38 | PA5 or FIFOADR1 | I/O/Z | I (PA5) | Multiplexed pin whose function is selected by:<br>IFCONFIG[1..0].<br>**PA5** is a bidirectional I/O port pin.<br>**FIFOADR1** is an input-only address select for the slave FIFOs connected to FD[7..0] or FD[15..0]. |
| 91 | 73 | 39 | PA6 or PKTEND | I/O/Z | I (PA6) | Multiplexed pin whose function is selected by the IFCONFIG[1:0] bits.<br>**PA6** is a bidirectional I/O port pin.<br>**PKTEND** is an input used to commit the FIFO packet data to the endpoint and whose polarity is programmable via FIFOPINPOLAR.5. |

**Table 5-1. FX1 Pin Definitions** (continued)[8]

| 128 TQFP | 100 TQFP | 56 QFN | Name | Type | Default | Description |
|---|---|---|---|---|---|---|
| 92 | 74 | 40 | 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.<br>**PA7** is a bidirectional I/O port pin.<br>**FLAGD** is a programmable slave-FIFO output status flag signal.<br>**SLCS#** gates all other slave FIFO enable/strobes |
| **Port B** | | | | | | |
| 44 | 34 | 18 | PB0 or FD[0] | I/O/Z | I (PB0) | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**PB0** is a bidirectional I/O port pin.<br>**FD[0]** is the bidirectional FIFO/GPIF data bus. |
| 45 | 35 | 19 | PB1 or FD[1] | I/O/Z | I (PB1) | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**PB1** is a bidirectional I/O port pin.<br>**FD[1]** is the bidirectional FIFO/GPIF data bus. |
| 46 | 36 | 20 | PB2 or FD[2] | I/O/Z | I (PB2) | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**PB2** is a bidirectional I/O port pin.<br>**FD[2]** is the bidirectional FIFO/GPIF data bus. |
| 47 | 37 | 21 | PB3 or FD[3] | I/O/Z | I (PB3) | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**PB3** is a bidirectional I/O port pin.<br>**FD[3]** is the bidirectional FIFO/GPIF data bus. |
| 54 | 44 | 22 | PB4 or FD[4] | I/O/Z | I (PB4) | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**PB4** is a bidirectional I/O port pin.<br>**FD[4]** is the bidirectional FIFO/GPIF data bus. |
| 55 | 45 | 23 | PB5 or FD[5] | I/O/Z | I (PB5) | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**PB5** is a bidirectional I/O port pin.<br>**FD[5]** is the bidirectional FIFO/GPIF data bus. |
| 56 | 46 | 24 | PB6 or FD[6] | I/O/Z | I (PB6) | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**PB6** is a bidirectional I/O port pin.<br>**FD[6]** is the bidirectional FIFO/GPIF data bus. |
| 57 | 47 | 25 | PB7 or FD[7] | I/O/Z | I (PB7) | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**PB7** is a bidirectional I/O port pin.<br>**FD[7]** is the bidirectional FIFO/GPIF data bus. |
| **PORT C** | | | | | | |
| 72 | 57 | | PC0 or GPIFADR0 | I/O/Z | I (PC0) | Multiplexed pin whose function is selected by PORTCCFG.0<br>**PC0** is a bidirectional I/O port pin.<br>**GPIFADR0** is a GPIF address output pin. |
| 73 | 58 | | PC1 or GPIFADR1 | I/O/Z | I (PC1) | Multiplexed pin whose function is selected by PORTCCFG.1<br>**PC1** is a bidirectional I/O port pin.<br>**GPIFADR1** is a GPIF address output pin. |
| 74 | 59 | | PC2 or GPIFADR2 | I/O/Z | I (PC2) | Multiplexed pin whose function is selected by PORTCCFG.2<br>**PC2** is a bidirectional I/O port pin.<br>**GPIFADR2** is a GPIF address output pin. |
| 75 | 60 | | PC3 or GPIFADR3 | I/O/Z | I (PC3) | Multiplexed pin whose function is selected by PORTCCFG.3<br>**PC3** is a bidirectional I/O port pin.<br>**GPIFADR3** is a GPIF address output pin. |
| 76 | 61 | | PC4 or GPIFADR4 | I/O/Z | I (PC4) | Multiplexed pin whose function is selected by PORTCCFG.4<br>**PC4** is a bidirectional I/O port pin.<br>**GPIFADR4** is a GPIF address output pin. |

**Table 5-1. FX1 Pin Definitions** (continued)[8]

| 128 TQFP | 100 TQFP | 56 QFN | Name | Type | Default | Description |
|---|---|---|---|---|---|---|
| 77 | 62 | | PC5 or GPIFADR5 | I/O/Z | I (PC5) | Multiplexed pin whose function is selected by PORTCCFG.5<br>**PC5** is a bidirectional I/O port pin.<br>**GPIFADR5** is a GPIF address output pin. |
| 78 | 63 | | PC6 or GPIFADR6 | I/O/Z | I (PC6) | Multiplexed pin whose function is selected by PORTCCFG.6<br>**PC6** is a bidirectional I/O port pin.<br>**GPIFADR6** is a GPIF address output pin. |
| 79 | 64 | | PC7 or GPIFADR7 | I/O/Z | I (PC7) | Multiplexed pin whose function is selected by PORTCCFG.7<br>**PC7** is a bidirectional I/O port pin.<br>**GPIFADR7** is a GPIF address output pin. |
| **PORT D** | | | | | | |
| 102 | 80 | 45 | 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.<br>**FD[8]** is the bidirectional FIFO/GPIF data bus. |
| 103 | 81 | 46 | 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.<br>**FD[9]** is the bidirectional FIFO/GPIF data bus. |
| 104 | 82 | 47 | 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.<br>**FD[10]** is the bidirectional FIFO/GPIF data bus. |
| 105 | 83 | 48 | 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.<br>**FD[11]** is the bidirectional FIFO/GPIF data bus. |
| 121 | 95 | 49 | 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.<br>**FD[12]** is the bidirectional FIFO/GPIF data bus. |
| 122 | 96 | 50 | 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.<br>**FD[13]** is the bidirectional FIFO/GPIF data bus. |
| 123 | 97 | 51 | 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.<br>**FD[14]** is the bidirectional FIFO/GPIF data bus. |
| 124 | 98 | 52 | 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.<br>**FD[15]** is the bidirectional FIFO/GPIF data bus. |
| **Port E** | | | | | | |
| 108 | 86 | | PE0 or T0OUT | I/O/Z | I (PE0) | Multiplexed pin whose function is selected by the PORTECFG.0 bit.<br>**PE0** is a bidirectional I/O port pin.<br>**T0OUT** is an active-HIGH signal from 8051 Timer-counter0. T0OUT outputs a high level for one CLKOUT clock cycle when Timer0 overflows. If Timer0 is operated in Mode 3 (two separate timer/counters), T0OUT is active when the low byte timer/counter overflows. |
| 109 | 87 | | PE1 or T1OUT | I/O/Z | I (PE1) | Multiplexed pin whose function is selected by the PORTECFG.1 bit.<br>**PE1** is a bidirectional I/O port pin.<br>**T1OUT** is an active-HIGH signal from 8051 Timer-counter1. T1OUT outputs a high level for one CLKOUT clock cycle when Timer1 overflows. If Timer1 is operated in Mode 3 (two separate timer/counters), T1OUT is active when the low byte timer/counter overflows. |
| 110 | 88 | | PE2 or T2OUT | I/O/Z | I (PE2) | Multiplexed pin whose function is selected by the PORTECFG.2 bit.<br>**PE2** is a bidirectional I/O port pin.<br>**T2OUT** is the active-HIGH output signal from 8051 Timer2. T2OUT is active (HIGH) for one clock cycle when Timer/Counter 2 overflows. |

**Table 5-1. FX1 Pin Definitions** (continued)[8]

| 128 TQFP | 100 TQFP | 56 QFN | Name | Type | Default | Description |
|---|---|---|---|---|---|---|
| 111 | 89 | | PE3 or RXD0OUT | I/O/Z | I (PE3) | Multiplexed pin whose function is selected by the PORTECFG.3 bit.<br>**PE3** is a bidirectional I/O port pin.<br>**RXD0OUT** is an active-HIGH signal from 8051 UART0. If RXD0OUT is selected and UART0 is in Mode 0, this pin provides the output data for UART0 only when it is in sync mode. Otherwise it is a 1. |
| 112 | 90 | | PE4 or RXD1OUT | I/O/Z | I (PE4) | Multiplexed pin whose function is selected by the PORTECFG.4 bit.<br>**PE4** is a bidirectional I/O port pin.<br>**RXD1OUT** is an active-HIGH output from 8051 UART1. When RXD1OUT is selected and UART1 is in Mode 0, this pin provides the output data for UART1 only when it is in sync mode. In Modes 1, 2, and 3, this pin is HIGH. |
| 113 | 91 | | PE5 or INT6 | I/O/Z | I (PE5) | Multiplexed pin whose function is selected by the PORTECFG.5 bit.<br>**PE5** is a bidirectional I/O port pin.<br>**INT6** is the 8051 INT6 interrupt request input signal. The INT6 pin is edge-sensitive, active HIGH. |
| 114 | 92 | | PE6 or T2EX | I/O/Z | I (PE6) | Multiplexed pin whose function is selected by the PORTECFG.6 bit.<br>**PE6** is a bidirectional I/O port pin.<br>**T2EX** is an active-high input signal to the 8051 Timer2. T2EX reloads timer 2 on its falling edge. T2EX is active only if the EXEN2 bit is set in T2CON. |
| 115 | 93 | | PE7 or GPIFADR8 | I/O/Z | I (PE7) | Multiplexed pin whose function is selected by the PORTECFG.7 bit.<br>**PE7** is a bidirectional I/O port pin.<br>**GPIFADR8** is a GPIF address output pin. |
| 4 | 3 | 1 | RDY0 or SLRD | Input | N/A | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**RDY0** is a GPIF input signal.<br>**SLRD** is the input-only read strobe with programmable polarity (FIFOPIN-POLAR.3) for the slave FIFOs connected to FD[7..0] or FD[15..0]. |
| 5 | 4 | 2 | RDY1 or SLWR | Input | N/A | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**RDY1** is a GPIF input signal.<br>**SLWR** is the input-only write strobe with programmable polarity (FIFOPIN-POLAR.2) for the slave FIFOs connected to FD[7..0] or FD[15..0]. |
| 6 | 5 | | RDY2 | Input | N/A | **RDY2** is a GPIF input signal. |
| 7 | 6 | | RDY3 | Input | N/A | **RDY3** is a GPIF input signal. |
| 8 | 7 | | RDY4 | Input | N/A | **RDY4** is a GPIF input signal. |
| 9 | 8 | | RDY5 | Input | N/A | **RDY5** is a GPIF input signal. |
| 69 | 54 | 29 | CTL0 or FLAGA | O/Z | H | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**CTL0** is a GPIF control output.<br>**FLAGA** is a programmable slave-FIFO output status flag signal. Defaults to programmable for the FIFO selected by the FIFOADR[1:0] pins. |
| 70 | 55 | 30 | CTL1 or FLAGB | O/Z | H | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**CTL1** is a GPIF control output.<br>**FLAGB** is a programmable slave-FIFO output status flag signal. Defaults to FULL for the FIFO selected by the FIFOADR[1:0] pins. |
| 71 | 56 | 31 | CTL2 or FLAGC | O/Z | H | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0].<br>**CTL2** is a GPIF control output.<br>**FLAGC** is a programmable slave-FIFO output status flag signal. Defaults to EMPTY for the FIFO selected by the FIFOADR[1:0] pins. |
| 66 | 51 | | CTL3 | O/Z | H | **CTL3** is a GPIF control output. |
| 67 | 52 | | CTL4 | Output | H | **CTL4** is a GPIF control output. |
| 98 | 76 | | CTL5 | Output | H | **CTL5** is a GPIF control output. |

**Table 5-1. FX1 Pin Definitions** (continued)[8]

| 128 TQFP | 100 TQFP | 56 QFN | Name | Type | Default | Description |
|---|---|---|---|---|---|---|
| 32 | 26 | 13 | IFCLK | I/O/Z | Z | Interface Clock, used for synchronously clocking data into or out of the slave FIFOs. IFCLK also serves as a timing reference for all slave FIFO control signals and GPIF. When internal clocking 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. |
| 28 | 22 | | INT4 | Input | N/A | **INT4** is the 8051 INT4 interrupt request input signal. The INT4 pin is edge-sensitive, active HIGH. |
| 106 | 84 | | INT5# | Input | N/A | **INT5#** is the 8051 INT5 interrupt request input signal. The INT5 pin is edge-sensitive, active LOW. |
| 31 | 25 | | T2 | Input | N/A | **T2** is the active-HIGH T2 input signal to 8051 Timer2, which provides the input to Timer2 when C/T2 = 1. When C/T2 = 0, Timer2 does not use this pin. |
| 30 | 24 | | T1 | Input | N/A | **T1** is the active-HIGH T1 signal for 8051 Timer1, which provides the input to Timer1 when C/T1 is 1. When C/T1 is 0, Timer1 does not use this bit. |
| 29 | 23 | | T0 | Input | N/A | **T0** is the active-HIGH T0 signal for 8051 Timer0, which provides the input to Timer0 when C/T0 is 1. When C/T0 is 0, Timer0 does not use this bit. |
| 53 | 43 | | RXD1 | Input | N/A | **RXD1**is an active-HIGH input signal for 8051 UART1, which provides data to the UART in all modes. |
| 52 | 42 | | TXD1 | Output | H | **TXD1**is an active-HIGH output pin from 8051 UART1, which provides the output clock in sync mode, and the output data in async mode. |
| 51 | 41 | | RXD0 | Input | N/A | **RXD0** is the active-HIGH RXD0 input to 8051 UART0, which provides data to the UART in all modes. |
| 50 | 40 | | TXD0 | Output | H | **TXD0** is the active-HIGH TXD0 output from 8051 UART0, which provides the output clock in sync mode, and the output data in async mode. |
| 42 | | | CS# | Output | H | **CS#** is the active-LOW chip select for external memory. |
| 41 | 32 | | WR# | Output | H | **WR#** is the active-LOW write strobe output for external memory. |
| 40 | 31 | | RD# | Output | H | **RD#** is the active-LOW read strobe output for external memory. |
| 38 | | | OE# | Output | H | **OE#** is the active-LOW output enable for external memory. |
| 33 | 27 | 14 | Reserved | Input | N/A | **Reserved**. Connect to ground. |
| | | | | | | |
| 101 | 79 | 44 | WAKEUP | Input | N/A | **USB Wakeup**. 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 EZ-USB FX1 chip from suspending. This pin has programmable polarity (WAKEUP.4). |
| 36 | 29 | 15 | SCL | OD | Z | **Clock** for the $I^2C$ interface. Connect to VCC with a 2.2K resistor, even if no $I^2C$ peripheral is attached. |
| 37 | 30 | 16 | SDA | OD | Z | **Data** for **$I^2C$ interface**. Connect to VCC with a 2.2K resistor, **even if no $I^2C$ peripheral is attached**. |
| | | | | | | |
| 2 | 1 | 55 | VCC | Power | N/A | **VCC**. Connect to 3.3V power source. |
| 26 | 20 | 11 | VCC | Power | N/A | **VCC**. Connect to 3.3V power source. |
| 43 | 33 | 17 | VCC | Power | N/A | **VCC**. Connect to 3.3V power source. |
| 48 | 38 | | VCC | Power | N/A | **VCC**. Connect to 3.3V power source. |
| 64 | 49 | 27 | VCC | Power | N/A | **VCC**. Connect to 3.3V power source. |
| 68 | 53 | | VCC | Power | N/A | **VCC**. Connect to 3.3V power source. |
| 81 | 66 | 32 | VCC | Power | N/A | **VCC**. Connect to 3.3V power source. |
| 100 | 78 | 43 | VCC | Power | N/A | **VCC**. Connect to 3.3V power source. |
| 107 | 85 | | VCC | Power | N/A | **VCC**. Connect to 3.3V power source. |

**Table 5-1. FX1 Pin Definitions** (continued)[8]

| 128 TQFP | 100 TQFP | 56 QFN | Name | Type | Default | Description |
|---|---|---|---|---|---|---|
| 3 | 2 | 56 | GND | Ground | N/A | **Ground**. |
| 27 | 21 | 12 | GND | Ground | N/A | **Ground**. |
| 49 | 39 | | GND | Ground | N/A | **Ground**. |
| 58 | 48 | 26 | GND | Ground | N/A | **Ground**. |
| 65 | 50 | 28 | GND | Ground | N/A | **Ground**. |
| 80 | 65 | | GND | Ground | N/A | **Ground**. |
| 93 | 75 | 41 | GND | Ground | N/A | **Ground**. |
| 116 | 94 | | GND | Ground | N/A | **Ground**. |
| 125 | 99 | 53 | GND | Ground | N/A | **Ground**. |
| | | | | | | |
| 14 | 13 | | NC | N/A | N/A | **No Connect**. This pin must be left open. |
| 15 | 14 | | NC | N/A | N/A | **No Connect**. This pin must be left open. |
| 16 | 15 | | NC | N/A | N/A | **No-connect**. This pin must be left open. |

## 6.0 Register Summary

FX1 register bit definitions are described in the EZ-USB TRM in greater detail.

**Table 6-1. FX1 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 | xxxxxxxx | RW |
| E480 | 128 | reserved | | | | | | | | | | | |
| | | GENERAL CONFIGURATION | | | | | | | | | | | |
| E600 | 1 | CPUCS | CPU Control & Status | 0 | 0 | PORTCSTB | CLKSPD1 | CLKSPD0 | CLKINV | CLKOE | 8051RES | 00000010 | rrbbbbbr |
| E601 | 1 | IFCONFIG | Interface Configuration (Ports, GPIF, slave FIFOs) | IFCLKSRC | 3048MHZ | IFCLKOE | IFCLKPOL | ASYNC | GSTATE | IFCFG1 | IFCFG0 | 10000000 | RW |
| E602 | 1 | PINFLAGSAB[9] | Slave FIFO FLAGA and FLAGB Pin Configuration | FLAGB3 | FLAGB2 | FLAGB1 | FLAGB0 | FLAGA3 | FLAGA2 | FLAGA1 | FLAGA0 | 00000000 | RW |
| E603 | 1 | PINFLAGSCD[9] | Slave FIFO FLAGC and FLAGD Pin Configuration | FLAGD3 | FLAGD2 | FLAGD1 | FLAGD0 | FLAGC3 | FLAGC2 | FLAGC1 | FLAGC0 | 00000000 | RW |
| E604 | 1 | FIFORESET[9] | Restore FIFOS to default state | NAKALL | 0 | 0 | 0 | EP3 | EP2 | EP1 | EP0 | xxxxxxxx | 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 | xxxxxxxx | RW |
| E607 | 1 | BPADDRL | Breakpoint Address L | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | xxxxxxxx | RW |
| E608 | 1 | UART230 | 230 Kbaud internally generated ref. clock | 0 | 0 | 0 | 0 | 0 | 0 | 230UART1 | 230UART0 | 00000000 | rrrrrrbb |
| E609 | 1 | FIFOPINPOLAR[9] | Slave FIFO Interface pins polarity | 0 | 0 | PKTEND | SLOE | SLRD | SLWR | EF | FF | 00000000 | rrbbbbbb |
| E60A | 1 | REVID | Chip Revision | rv7 | rv6 | rv5 | rv4 | rv3 | rv2 | rv1 | rv0 | RevA 00000001 | R |
| E60B | 1 | REVCTL[9] | 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 | brbbrrrr |
| E611 | 1 | EP1INCFG | Endpoint 1-IN Configuration | VALID | 0 | TYPE1 | TYPE0 | 0 | 0 | 0 | 0 | 10100000 | brbbrrrr |
| 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[9] | Endpoint 2 / slave FIFO configuration | 0 | INFM1 | OEP1 | AUTOOUT | AUTOIN | ZEROLENIN | 0 | WORDWIDE | 00000101 | rbbbbbrb |
| E619 | 1 | EP4FIFOCFG[9] | Endpoint 4 / slave FIFO configuration | 0 | INFM1 | OEP1 | AUTOOUT | AUTOIN | ZEROLENIN | 0 | WORDWIDE | 00000101 | rbbbbbrb |
| E61A | 1 | EP6FIFOCFG[9] | Endpoint 6 / slave FIFO configuration | 0 | INFM1 | OEP1 | AUTOOUT | AUTOIN | ZEROLENIN | 0 | WORDWIDE | 00000101 | rbbbbbrb |
| E61B | 1 | EP8FIFOCFG[9] | Endpoint 8 / slave FIFO configuration | 0 | INFM1 | OEP1 | AUTOOUT | AUTOIN | ZEROLENIN | 0 | WORDWIDE | 00000101 | rbbbbbrb |
| E61C | 4 | reserved | | | | | | | | | | | |
| E620 | 1 | EP2AUTOINLENH[9] | Endpoint 2 AUTOIN Packet Length H | 0 | 0 | 0 | 0 | 0 | PL10 | PL9 | PL8 | 00000010 | rrrrrbbb |
| E621 | 1 | EP2AUTOINLENL[9] | Endpoint 2 AUTOIN Packet Length L | PL7 | PL6 | PL5 | PL4 | PL3 | PL2 | PL1 | PL0 | 00000000 | RW |
| E622 | 1 | EP4AUTOINLENH[9] | Endpoint 4 AUTOIN Packet Length H | 0 | 0 | 0 | 0 | 0 | 0 | PL9 | PL8 | 00000010 | rrrrrrbb |
| E623 | 1 | EP4AUTOINLENL[9] | Endpoint 4 AUTOIN Packet Length L | PL7 | PL6 | PL5 | PL4 | PL3 | PL2 | PL1 | PL0 | 00000000 | RW |
| E624 | 1 | EP6AUTOINLENH[9] | Endpoint 6 AUTOIN Packet Length H | 0 | 0 | 0 | 0 | 0 | PL10 | PL9 | PL8 | 00000010 | rrrrrbbb |
| E625 | 1 | EP6AUTOINLENL[9] | Endpoint 6 AUTOIN Packet Length L | PL7 | PL6 | PL5 | PL4 | PL3 | PL2 | PL1 | PL0 | 00000000 | RW |
| E626 | 1 | EP8AUTOINLENH[9] | Endpoint 8 AUTOIN Packet Length H | 0 | 0 | 0 | 0 | 0 | 0 | PL9 | PL8 | 00000010 | rrrrrrbb |
| E627 | 1 | EP8AUTOINLENL[9] | 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 | rrrrrrrb |
| 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 | 11111111 | R |
| E62B | 1 | ECC1B1 | ECC1 Byte 1 Address | LINE7 | LINE6 | LINE5 | LINE4 | LINE3 | LINE2 | LINE1 | LINE0 | 11111111 | R |
| E62C | 1 | ECC1B2 | ECC1 Byte 2 Address | COL5 | COL4 | COL3 | COL2 | COL1 | COL0 | LINE17 | LINE16 | 11111111 | R |
| E62D | 1 | ECC2B0 | ECC2 Byte 0 Address | LINE15 | LINE14 | LINE13 | LINE12 | LINE11 | LINE10 | LINE9 | LINE8 | 11111111 | R |
| E62E | 1 | ECC2B1 | ECC2 Byte 1 Address | LINE7 | LINE6 | LINE5 | LINE4 | LINE3 | LINE2 | LINE1 | LINE0 | 11111111 | R |

**Note:**

9. Read and writes to these register may require synchronization delay, see Technical Reference Manual for “Synchronization Delay.”

**Table 6-1. FX1 Register Summary** (continued)

| Hex | Size | Name | Description | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | Default | Access |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E62F | 1 | ECC2B2 | ECC2 Byte 2 Address | COL5 | COL4 | COL3 | COL2 | COL1 | COL0 | 0 | 0 | 11111111 | R |
| | | | | | | | | | | | | | |
| E630 | 1 | EP2FIFOPFH[9] | Endpoint 2 / slave FIFO Programmable Flag H ISO Mode | DECIS | PKTSTAT | IN: PKTS[2] OUT:PFC12 | IN: PKTS[1] OUT:PFC11 | IN: PKTS[0] OUT:PFC10 | 0 | PFC9 | PFC8 | 10001000 | bbbbbrbb |
| E630 | 1 | EP2FIFOPFH[9] | Endpoint 2 / slave FIFO Programmable Flag H Non-ISO Mode | DECIS | PKTSTAT | OUT:PFC12 | OUT:PFC11 | OUT:PFC10 | 0 | PFC9 | IN:PKTS[2] OUT:PFC8 | 10001000 | bbbbbrbb |
| | | | | | | | | | | | | | |
| E631 | 1 | EP2FIFOPFL[9] | 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 | 1 | EP4FIFOPFH[9] | Endpoint 4 / slave FIFO Programmable Flag H ISO Mode | DECIS | PKTSTAT | 0 | IN: PKTS[1] OUT:PFC10 | IN: PKTS[0] OUT:PFC9 | 0 | 0 | PFC8 | 10001000 | bbrbbrrb |
| E632 | 1 | EP4FIFOPFH[9] | Endpoint 4 / slave FIFO Programmable Flag H Non-ISO Mode | DECIS | PKTSTAT | 0 | OUT:PFC10 | OUT:PFC9 | 0 | 0 | PFC8 | 10001000 | bbrbbrrb |
| | | | | | | | | | | | | | |
| E633 | 1 | EP4FIFOPFL[9] | 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 | 1 | EP6FIFOPFH[9] | Endpoint 6 / slave FIFO Programmable Flag H ISO Mode | DECIS | PKTSTAT | INPKTS[2] OUT:PFC12 | IN: PKTS[1] OUT:PFC11 | IN: PKTS[0] OUT:PFC10 | 0 | PFC9 | PFC8 | 00001000 | bbbbbrbb |
| E634 | 1 | EP6FIFOPFH[9] | Endpoint 6 / slave FIFO Programmable Flag H Non-ISO Mode | DECIS | PKTSTAT | OUT:PFC12 | OUT:PFC11 | OUT:PFC10 | 0 | PFC9 | IN:PKTS[2] OUT:PFC8 | 00001000 | bbbbbrbb |
| | | | | | | | | | | | | | |
| E635 | 1 | EP6FIFOPFL[9] | 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 | 1 | EP8FIFOPFH[9] | Endpoint 8 / slave FIFO Programmable Flag H ISO Mode | DECIS | PKTSTAT | 0 | IN: PKTS[1] OUT:PFC10 | IN: PKTS[0] OUT:PFC9 | 0 | 0 | PFC8 | 00001000 | bbrbbrrb |
| E636 | 1 | EP8FIFOPFH[9] | Endpoint 8 / slave FIFO Programmable Flag H Non-ISO Mode | DECIS | PKTSTAT | 0 | OUT:PFC10 | OUT:PFC9 | 0 | 0 | PFC8 | 00001000 | bbrbbrrb |
| | | | | | | | | | | | | | |
| E637 | 1 | EP8FIFOPFL[9] ISO Mode | Endpoint 8 / slave FIFO Programmable Flag L | PFC7 | PFC6 | PFC5 | PFC4 | PFC3 | PFC2 | PFC1 | PFC0 | 00000000 | RW |
| E637 | 1 | EP8FIFOPFL[9] Non-ISO Mode | 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 | reserved | | | | | | | | | | | |
| E641 | 1 | reserved | | | | | | | | | | | |
| E642 | 1 | reserved | | | | | | | | | | | |
| E643 | 1 | reserved | | | | | | | | | | | |
| E644 | 4 | reserved | | | | | | | | | | | |
| E648 | 1 | INPKTEND[9] | Force IN Packet End | Skip | 0 | 0 | 0 | EP3 | EP2 | EP1 | EP0 | xxxxxxxx | W |
| E649 | 7 | OUTPKTEND[9] | Force OUT Packet End | Skip | 0 | 0 | 0 | EP3 | EP2 | EP1 | EP0 | xxxxxxxx | W |
| | | INTERRUPTS | | | | | | | | | | | |
| E650 | 1 | EP2FIFOIE[9] | Endpoint 2 slave FIFO Flag Interrupt Enable | 0 | 0 | 0 | 0 | EDGEPF | PF | EF | FF | 00000000 | RW |
| E651 | 1 | EP2FIFOIRQ[9,10] | Endpoint 2 slave FIFO Flag Interrupt Request | 0 | 0 | 0 | 0 | 0 | PF | EF | FF | 00000111 | rrrrrbbb |
| E652 | 1 | EP4FIFOIE[9] | Endpoint 4 slave FIFO Flag Interrupt Enable | 0 | 0 | 0 | 0 | EDGEPF | PF | EF | FF | 00000000 | RW |
| E653 | 1 | EP4FIFOIRQ[9,10] | Endpoint 4 slave FIFO Flag Interrupt Request | 0 | 0 | 0 | 0 | 0 | PF | EF | FF | 00000111 | rrrrrbbb |
| E654 | 1 | EP6FIFOIE[9] | Endpoint 6 slave FIFO Flag Interrupt Enable | 0 | 0 | 0 | 0 | EDGEPF | PF | EF | FF | 00000000 | RW |
| E655 | 1 | EP6FIFOIRQ[9,10] | Endpoint 6 slave FIFO Flag Interrupt Request | 0 | 0 | 0 | 0 | 0 | PF | EF | FF | 00000110 | rrrrrbbb |
| E656 | 1 | EP8FIFOIE[9] | Endpoint 8 slave FIFO Flag Interrupt Enable | 0 | 0 | 0 | 0 | EDGEPF | PF | EF | FF | 00000000 | RW |
| E657 | 1 | EP8FIFOIRQ[9,10] | Endpoint 8 slave FIFO Flag Interrupt Request | 0 | 0 | 0 | 0 | 0 | PF | EF | FF | 00000110 | rrrrrbbb |
| E658 | 1 | IBNIE | IN-BULK-NAK Interrupt Enable | 0 | 0 | EP8 | EP6 | EP4 | EP2 | EP1 | EP0 | 00000000 | RW |

**Note:**
10. SFRs not part of the standard 8051 architecture.
The register can only be reset, it cannot be set.

**Table 6-1. FX1 Register Summary** (continued)

| Hex | Size | Name | Description | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | Default | Access |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E659 | 1 | IBNIRQ[10] | IN-BULK-NAK interrupt Request | 0 | 0 | EP8 | EP6 | EP4 | EP2 | EP1 | EP0 | 00xxxxxx | rrbbbbbb |
| E65A | 1 | NAKIE | Endpoint Ping-NAK / IBN Interrupt Enable | EP8 | EP6 | EP4 | EP2 | EP1 | EP0 | 0 | IBN | 00000000 | RW |
| E65B | 1 | NAKIRQ[10] | Endpoint Ping-NAK / IBN Interrupt Request | EP8 | EP6 | EP4 | EP2 | EP1 | EP0 | 0 | IBN | xxxxxx0x | bbbbbbrb |
| E65C | 1 | USBIE | USB Int Enables | 0 | EP0ACK | 0 | URES | SUSP | SUTOK | SOF | SUDAV | 00000000 | RW |
| E65D | 1 | USBIRQ[10] | USB Interrupt Requests | 0 | EP0ACK | 0 | URES | SUSP | SUTOK | SOF | SUDAV | 0xxxxxxx | rbbbbbbb |
| E65E | 1 | EPIE | Endpoint Interrupt Enables | EP8 | EP6 | EP4 | EP2 | EP1OUT | EP1IN | EP0OUT | EP0IN | 00000000 | RW |
| E65F | 1 | EPIRQ[10] | Endpoint Interrupt Requests | EP8 | EP6 | EP4 | EP2 | EP1OUT | EP1IN | EP0OUT | EP0IN | 0 | RW |
| E660 | 1 | GPIFIE[9] | GPIF Interrupt Enable | 0 | 0 | 0 | 0 | 0 | 0 | GPIFWF | GPIFDONE | 00000000 | RW |
| E661 | 1 | GPIFIRQ[9] | 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[10] | USB Error Interrupt Requests | ISOEP8 | ISOEP6 | ISOEP4 | ISOEP2 | 0 | 0 | 0 | ERRLIMIT | 0000000x | bbbbrrrb |
| 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 | xxxxxxxx | W |
| E666 | 1 | INT2IVEC | Interrupt 2 (USB) Autovector | 0 | I2V4 | I2V3 | I2V2 | I2V1 | I2V0 | 0 | 0 | 00000000 | R |
| E667 | 1 | INT4IVEC | Interrupt 4 (slave FIFO & GPIF) Autovector | 1 | 0 | I4V3 | I4V2 | I4V1 | I4V0 | 0 | 0 | 10000000 | R |
| E668 | 1 | INTSETUP | Interrupt 2&4 setup | 0 | 0 | 0 | 0 | AV2EN | 0 | INT4SRC | 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 | XTALINSRC | XTALIN Clock Source | 0 | 0 | 0 | 0 | 0 | 0 | 0 | EXTCLK | 00000000 | rrrrrrrb |
| E677 | 1 | reserved | | | | | | | | | | | |
| E678 | 1 | I2CS | I²C Bus Control & Status | START | STOP | LASTRD | ID1 | ID0 | BERR | ACK | DONE | 000xx000 | bbbrrrrr |
| E679 | 1 | I2DAT | I²C Bus Data | d7 | d6 | d5 | d4 | d3 | d2 | d1 | d0 | xxxxxxxx | RW |
| E67A | 1 | I2CTL | I²C 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 | xxxxxxxx | RW |
| E67C | 1 | XAUTODAT2 | Autoptr2 MOVX access, when APTREN=1 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| | | UDMA CRC | | | | | | | | | | | |
| E67D | 1 | UDMACRCH[9] | UDMA CRC MSB | CRC15 | CRC14 | CRC13 | CRC12 | CRC11 | CRC10 | CRC9 | CRC8 | 01001010 | RW |
| E67E | 1 | UDMACRCL[9] | 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 | QSIGNAL2 | QSIGNAL1 | QSIGNAL0 | 00000000 | brrrbbbb |
| | | USB CONTROL | | | | | | | | | | | |
| E680 | 1 | USBCS | USB Control & Status | 0 | 0 | 0 | 0 | DISCON | NOSYNSOF | RENUM | SIGRSUME | x0000000 | rrrrbbbb |
| E681 | 1 | SUSPEND | Put chip into suspend | x | x | x | x | x | x | x | x | xxxxxxxx | W |
| E682 | 1 | WAKEUPCS | Wakeup Control & Status | WU2 | WU | WU2POL | WUPOL | 0 | DPEN | WU2EN | WUEN | xx000101 | bbbbrbbb |
| E683 | 1 | TOGCTL | Toggle Control | Q | S | R | IO | EP3 | EP2 | EP1 | EP0 | x0000000 | rrrbbbbb |
| 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 | xxxxxxxx | R |
| E686 | 1 | reserved | | | | | | | | | | | |
| E687 | 1 | FNADDR | USB Function address | 0 | FA6 | FA5 | FA4 | FA3 | FA2 | FA1 | FA0 | 0xxxxxxx | R |
| E688 | 2 | reserved | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | ENDPOINTS | | | | | | | | | | | |
| E68A | 1 | EP0BCH[9] | Endpoint 0 Byte Count H | (BC15) | (BC14) | (BC13) | (BC12) | (BC11) | (BC10) | (BC9) | (BC8) | xxxxxxxx | RW |
| E68B | 1 | EP0BCL[9] | Endpoint 0 Byte Count L | (BC7) | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 | xxxxxxxx | RW |
| E68C | 1 | reserved | | | | | | | | | | | |
| E68D | 1 | EP1OUTBC | Endpoint 1 OUT Byte Count | 0 | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 | xxxxxxxx | RW |
| E68E | 1 | reserved | | | | | | | | | | | |
| E68F | 1 | EP1INBC | Endpoint 1 IN Byte Count | 0 | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 | xxxxxxxx | RW |
| E690 | 1 | EP2BCH[9] | Endpoint 2 Byte Count H | 0 | 0 | 0 | 0 | 0 | BC10 | BC9 | BC8 | xxxxxxxx | RW |
| E691 | 1 | EP2BCL[9] | Endpoint 2 Byte Count L | BC7/SKIP | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 | xxxxxxxx | RW |
| E692 | 2 | reserved | | | | | | | | | | | |
| E694 | 1 | EP4BCH[9] | Endpoint 4 Byte Count H | 0 | 0 | 0 | 0 | 0 | 0 | BC9 | BC8 | xxxxxxxx | RW |
| E695 | 1 | EP4BCL[9] | Endpoint 4 Byte Count L | BC7/SKIP | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 | xxxxxxxx | RW |
| E696 | 2 | reserved | | | | | | | | | | | |

**Table 6-1. FX1 Register Summary** (continued)

| Hex | Size | Name | Description | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | Default | Access |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E698 | 1 | EP6BCH[9] | Endpoint 6 Byte Count H | 0 | 0 | 0 | 0 | 0 | BC10 | BC9 | BC8 | xxxxxxxx | RW |
| E699 | 1 | EP6BCL[9] | Endpoint 6 Byte Count L | BC7/SKIP | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 | xxxxxxxx | RW |
| E69A | 2 | reserved | | | | | | | | | | | |
| E69C | 1 | EP8BCH[9] | Endpoint 8 Byte Count H | 0 | 0 | 0 | 0 | 0 | 0 | BC9 | BC8 | xxxxxxxx | RW |
| E69D | 1 | EP8BCL[9] | Endpoint 8 Byte Count L | BC7/SKIP | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 | xxxxxxxx | RW |
| E69E | 2 | reserved | | | | | | | | | | | |
| E6A0 | 1 | EP0CS | Endpoint 0 Control and Status | HSNAK | 0 | 0 | 0 | 0 | 0 | BUSY | STALL | 10000000 | bbbbbbrb |
| E6A1 | 1 | EP1OUTCS | Endpoint 1 OUT Control and Status | 0 | 0 | 0 | 0 | 0 | 0 | BUSY | STALL | 00000000 | bbbbbbrb |
| E6A2 | 1 | EP1INCS | Endpoint 1 IN Control and Status | 0 | 0 | 0 | 0 | 0 | 0 | BUSY | STALL | 00000000 | bbbbbbrb |
| E6A3 | 1 | EP2CS | Endpoint 2 Control and Status | 0 | NPAK2 | NPAK1 | NPAK0 | FULL | EMPTY | 0 | STALL | 00101000 | rrrrrrrb |
| E6A4 | 1 | EP4CS | Endpoint 4 Control and Status | 0 | 0 | NPAK1 | NPAK0 | FULL | EMPTY | 0 | STALL | 00101000 | rrrrrrrb |
| E6A5 | 1 | EP6CS | Endpoint 6 Control and Status | 0 | NPAK2 | NPAK1 | NPAK0 | FULL | EMPTY | 0 | STALL | 00000100 | rrrrrrrb |
| E6A6 | 1 | EP8CS | Endpoint 8 Control and Status | 0 | 0 | NPAK1 | NPAK0 | FULL | EMPTY | 0 | STALL | 00000100 | rrrrrrrb |
| 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 | Setup Data Pointer high address byte | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | xxxxxxxx | RW |
| E6B4 | 1 | SUDPTRL | Setup Data Pointer low address byte | A7 | A6 | A5 | A4 | A3 | A2 | A1 | 0 | xxxxxxx0 | bbbbbbbr |
| E6B5 | 1 | SUDPTRCTL | Setup Data Pointer Auto Mode | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SDPAUTO | 00000001 | RW |
| | 2 | reserved | | | | | | | | | | | |
| E6B8 | 8 | SETUPDAT | 8 bytes of setup data | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | R |
| | | | SETUPDAT[0] = bmRequestType | | | | | | | | | | |
| | | | SETUPDAT[1] = bmRequest | | | | | | | | | | |
| | | | SETUPDAT[2:3] = wValue | | | | | | | | | | |
| | | | SETUPDAT[4:5] = wIndex | | | | | | | | | | |
| | | | SETUPDAT[6:7] = wLength | | | | | | | | | | |
| | | GPIF | | | | | | | | | | | |
| E6C0 | 1 | GPIFWFSELECT | 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 | CTL5 | CTL4 | CTL3 | CTL2 | CTL1 | CTL0 | 11111111 | RW |
| E6C3 | 1 | GPIFCTLCFG | CTL Drive Type | TRICTL | 0 | CTL5 | CTL4 | CTL3 | CTL2 | CTL1 | CTL0 | 00000000 | RW |
| E6C4 | 1 | GPIFADRH[9] | GPIF Address H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | GPIFA8 | 00000000 | RW |
| E6C5 | 1 | GPIFADRL[9] | GPIF Address L | GPIFA7 | GPIFA6 | GPIFA5 | GPIFA4 | GPIFA3 | GPIFA2 | GPIFA1 | GPIFA0 | 00000000 | RW |
| | | FLOWSTATE | | | | | | | | | | | |
| E6C6 | 1 | FLOWSTATE | Flowstate Enable and Selector | FSE | 0 | 0 | 0 | 0 | FS2 | FS1 | FS0 | 00000000 | brrrrbbb |
| 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/ CTL5 | CTL0E0/ CTL4 | CTL3 | CTL2 | CTL1 | CTL0 | 00000000 | RW |
| E6C9 | 1 | FLOWEQ1CTL | CTL-Pin States in Flowstate (when Logic = 1) | CTL0E3 | CTL0E2 | CTL0E1/ CTL5 | CTL0E0/ CTL4 | CTL3 | CTL2 | CTL1 | CTL0 | 00000000 | RW |
| E6CA | 1 | FLOWHOLDOFF | Holdoff Configuration | HOPERIOD3 | HOPERIOD2 | HOPERIOD1 | HOPERIOD 0 | HOSTATE | HOCTL2 | HOCTL1 | HOCTL0 | 00000000 | RW |

**Table 6-1. FX1 Register Summary** (continued)

| Hex | Size | Name | Description | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | Default | Access |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | 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[9] | GPIF Transaction Count Byte 3 | TC31 | TC30 | TC29 | TC28 | TC27 | TC26 | TC25 | TC24 | 00000000 | RW |
| E6CF | 1 | GPIFTCB2[9] | GPIF Transaction Count Byte 2 | TC23 | TC22 | TC21 | TC20 | TC19 | TC18 | TC17 | TC16 | 00000000 | RW |
| E6D0 | 1 | GPIFTCB1[9] | GPIF Transaction Count Byte 1 | TC15 | TC14 | TC13 | TC12 | TC11 | TC10 | TC9 | TC8 | 00000000 | RW |
| E6D1 | 1 | GPIFTCB0[9] | GPIF Transaction Count Byte 0 | TC7 | TC6 | TC5 | TC4 | TC3 | TC2 | TC1 | TC0 | 00000001 | RW |
| | 2 | reserved | | | | | | | | | | 00000000 | RW |
| | | reserved | | | | | | | | | | | |
| | | reserved | | | | | | | | | | | |
| E6D2 | 1 | EP2GPIFFLGSEL[9] | Endpoint 2 GPIF Flag select | 0 | 0 | 0 | 0 | 0 | 0 | FS1 | FS0 | 00000000 | RW |
| E6D3 | 1 | EP2GPIFPFSTOP | Endpoint 2 GPIF stop transaction on prog. flag | 0 | 0 | 0 | 0 | 0 | 0 | 0 | FIFO2FLAG | 00000000 | RW |
| E6D4 | 1 | EP2GPIFTRIG[9] | Endpoint 2 GPIF Trigger | x | x | x | x | x | x | x | x | xxxxxxxx | W |
| | 3 | reserved | | | | | | | | | | | |
| | | reserved | | | | | | | | | | | |
| | | reserved | | | | | | | | | | | |
| E6DA | 1 | EP4GPIFFLGSEL[9] | Endpoint 4 GPIF Flag select | 0 | 0 | 0 | 0 | 0 | 0 | FS1 | FS0 | 00000000 | RW |
| E6DB | 1 | EP4GPIFPFSTOP | Endpoint 4 GPIF stop transaction on GPIF Flag | 0 | 0 | 0 | 0 | 0 | 0 | 0 | FIFO4FLAG | 00000000 | RW |
| E6DC | 1 | EP4GPIFTRIG[9] | Endpoint 4 GPIF Trigger | x | x | x | x | x | x | x | x | xxxxxxxx | W |
| | 3 | reserved | | | | | | | | | | | |
| | | reserved | | | | | | | | | | | |
| | | reserved | | | | | | | | | | | |
| E6E2 | 1 | EP6GPIFFLGSEL[9] | Endpoint 6 GPIF Flag select | 0 | 0 | 0 | 0 | 0 | 0 | FS1 | FS0 | 00000000 | RW |
| E6E3 | 1 | EP6GPIFPFSTOP | Endpoint 6 GPIF stop transaction on prog. flag | 0 | 0 | 0 | 0 | 0 | 0 | 0 | FIFO6FLAG | 00000000 | RW |
| E6E4 | 1 | EP6GPIFTRIG[9] | Endpoint 6 GPIF Trigger | x | x | x | x | x | x | x | x | xxxxxxxx | W |
| | 3 | reserved | | | | | | | | | | | |
| | | reserved | | | | | | | | | | | |
| | | reserved | | | | | | | | | | | |
| E6EA | 1 | EP8GPIFFLGSEL[9] | Endpoint 8 GPIF Flag select | 0 | 0 | 0 | 0 | 0 | 0 | FS1 | FS0 | 00000000 | RW |
| E6EB | 1 | EP8GPIFPFSTOP | Endpoint 8 GPIF stop transaction on prog. flag | 0 | 0 | 0 | 0 | 0 | 0 | 0 | FIFO8FLAG | 00000000 | RW |
| E6EC | 1 | EP8GPIFTRIG[9] | Endpoint 8 GPIF Trigger | x | x | x | x | x | x | x | x | xxxxxxxx | W |
| | 3 | reserved | | | | | | | | | | | |
| E6F0 | 1 | XGPIFSGLDATH | GPIF Data H (16-bit mode only) | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | xxxxxxxx | RW |
| E6F1 | 1 | XGPIFSGLDATLX | Read/Write GPIF Data L & trigger transaction | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| E6F2 | 1 | XGPIFSGLDATL-NOX | Read GPIF Data L, no transaction trigger | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | R |
| E6F3 | 1 | GPIFREADYCFG | Internal RDY, Sync/Async, RDY pin states | INTRDY | SAS | TCXRDY5 | 0 | 0 | 0 | 0 | 0 | 00000000 | bbbrrrrr |
| | | | | | | | | | | | | | |
| E6F4 | 1 | GPIFREADYSTAT | GPIF Ready Status | 0 | 0 | RDY5 | RDY4 | RDY3 | RDY2 | RDY1 | RDY0 | 00xxxxxx | R |
| E6F5 | 1 | GPIFABORT | Abort GPIF Waveforms | x | x | x | x | x | x | x | x | xxxxxxxx | W |
| E6F6 | 2 | reserved | | | | | | | | | | | |
| | | ENDPOINT BUFFERS | | | | | | | | | | | |
| E740 | 64 | EP0BUF | EP0-IN/-OUT buffer | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| E780 | 64 | EP10UTBUF | EP1-OUT buffer | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| E7C0 | 64 | EP1INBUF | EP1-IN buffer | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| | 2048 | reserved | | | | | | | | | | | RW |
| F000 | 1023 | EP2FIFOBUF | 64/1023-byte EP 2 / slave FIFO buffer (IN or OUT) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| F400 | 64 | EP4FIFOBUF | 64 byte EP 4 / slave FIFO buffer (IN or OUT) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| F600 | 64 | reserved | | | | | | | | | | | |
| F800 | 1023 | EP6FIFOBUF | 64/1023-byte EP 6 / slave FIFO buffer (IN or OUT) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| FC00 | 64 | EP8FIFOBUF | 64 byte EP 8 / slave FIFO buffer (IN or OUT) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| FE00 | 64 | reserved | | | | | | | | | | | |
| xxxx | | I²C Configuration Byte | | 0 | DISCON | 0 | 0 | 0 | 0 | 0 | 400KHZ | xxxxxxxx [11] | n/a |
| | | Special Function Registers (SFRs) | | | | | | | | | | | |
| 80 | 1 | IOA[10] | Port A (bit addressable) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |

**Table 6-1. FX1 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 | 00000111 | 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[10] | Data Pointer 1 L | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | 00000000 | RW |
| 85 | 1 | DPH1[10] | Data Pointer 1 H | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | 00000000 | RW |
| 86 | 1 | DPS[10] | 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[10] | Clock Control | x | x | T2M | T1M | T0M | MD2 | MD1 | MD0 | 00000001 | RW |
| 8F | 1 | reserved | | | | | | | | | | | |
| 90 | 1 | IOB[10] | Port B (bit addressable) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| 91 | 1 | EXIF[10] | External Interrupt Flag(s) | IE5 | IE4 | I²CINT | USBNT | 1 | 0 | 0 | 0 | 00001000 | RW |
| 92 | 1 | MPAGE[10] | 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[10] | Autopointer 1 Address H | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | 00000000 | RW |
| 9B | 1 | AUTOPTRL1[10] | Autopointer 1 Address L | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | 00000000 | RW |
| 9C | 1 | reserved | | | | | | | | | | | |
| 9D | 1 | AUTOPTRH2[10] | Autopointer 2 Address H | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | 00000000 | RW |
| 9E | 1 | AUTOPTRL2[10] | Autopointer 2 Address L | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | 00000000 | RW |
| 9F | 1 | reserved | | | | | | | | | | | |
| A0 | 1 | IOC[10] | Port C (bit addressable) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| A1 | 1 | INT2CLR[10] | Interrupt 2 clear | x | x | x | x | x | x | x | x | xxxxxxxx | W |
| A2 | 1 | INT4CLR[10] | Interrupt 4 clear | x | x | x | x | x | x | x | x | xxxxxxxx | 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[10] | Endpoint 2,4,6,8 status flags | EP8F | EP8E | EP6F | EP6E | EP4F | EP4E | EP2F | EP2E | 01011010 | R |
| AB | 1 | EP24FIFOFLGS[10] | Endpoint 2,4 slave FIFO status flags | 0 | EP4PF | EP4EF | EP4FF | 0 | EP2PF | EP2EF | EP2FF | 00100010 | R |
| AC | 1 | EP68FIFOFLGS[10] | Endpoint 6,8 slave FIFO status flags | 0 | EP8PF | EP8EF | EP8FF | 0 | EP6PF | EP6EF | EP6FF | 01100110 | R |
| AD | 2 | reserved | | | | | | | | | | | |
| AF | 1 | AUTOPTRSETUP[10] | Autopointer 1&2 setup | 0 | 0 | 0 | 0 | 0 | APTR2INC | APTR1INC | APTREN | 00000110 | RW |
| B0 | 1 | IOD[10] | Port D (bit addressable) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| B1 | 1 | IOE[10] | Port E (NOT bit addressable) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| B2 | 1 | OEA[10] | Port A Output Enable | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00000000 | RW |
| B3 | 1 | OEB[10] | Port B Output Enable | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00000000 | RW |
| B4 | 1 | OEC[10] | Port C Output Enable | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00000000 | RW |
| B5 | 1 | OED[10] | Port D Output Enable | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00000000 | RW |
| B6 | 1 | OEE[10] | 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[10] | Endpoint 0&1 Status | 0 | 0 | 0 | 0 | 0 | EP1INBSY | EP1OUTBSY | EP0BSY | 00000000 | R |
| BB | 1 | GPIFTRIG[10] [9] | Endpoint 2,4,6,8 GPIF slave FIFO Trigger | DONE | 0 | 0 | 0 | 0 | RW | EP1 | EP0 | 10000xxx | brrrrbbb |
| BC | 1 | reserved | | | | | | | | | | | |
| BD | 1 | GPIFSGLDATH[10] | GPIF Data H (16-bit mode only) | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | xxxxxxxx | RW |
| BE | 1 | GPIFSGLDATLX[10] | GPIF Data L w/ Trigger | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| BF | 1 | GPIFSGLDATLNOX[10] | GPIF Data L w/ No Trigger | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | R |
| C0 | 1 | SCON1[10] | 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[10] | Serial Port 1 Data Buffer | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00000000 | RW |
| C2 | 6 | reserved | | | | | | | | | | | |

**Notes:**

11. If no EEPROM is detected by the SIE then the default is 00000000.

**Table 6-1. FX1 Register Summary** (continued)

| Hex | Size | Name | Description | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | Default | Access |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C8 | 1 | T2CON | Timer/Counter 2 Control (bit addressable) | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | CT2 | CPRL2 | 00000000 | RW |
| 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[10] | 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[10] | External Interrupt Enable(s) | 1 | 1 | 1 | EX6 | EX5 | EX4 | EI²C | 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[10] | External Interrupt Priority Control | 1 | 1 | 1 | PX6 | PX5 | PX4 | PI²C | PUSB | 11100000 | RW |
| F9 | 7 | reserved | | | | | | | | | | | |

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

## 7.0 Absolute Maximum Ratings

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

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

Supply Voltage to Ground Potential ..... –0.5V to +4.0V

DC Input Voltage to Any Input Pin ..... 5.25V[12]

DC Voltage Applied to Outputs in High-Z State ..... –0.5V to VCC + 0.5V

Power Dissipation ..... 235 mW

Static Discharge Voltage ..... > 2000V

Max Output Current, per I/O port ..... 10 mA

Max Output Current, all five I/O ports (128- and 100-pin packages) ..... 50 mA

## 8.0 Operating Conditions

$T_A$ (Ambient Temperature Under Bias) ..... 0°C to +70°C

Supply Voltage ..... +3.15V to +3.45V

Ground Voltage ..... 0V

$F_{OSC}$ (Oscillator or Crystal Frequency) ... 24 MHz ± 100 ppm ..... Parallel Resonant

## 9.0 DC Characteristics

**Table 9-1. DC Characteristics**

| Parameter | Description | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| VCC | Supply Voltage | | 3.15 | 3.3 | 3.45 | V |
| VCC Ramp Up | 0 to 3.3V | | 200 | | | ∝s |
| $V_{IH}$ | Input HIGH Voltage | | 2 | | 5.25 | V |
| $V_{IL}$ | Input LOW Voltage | | –0.5 | | 0.8 | V |
| $V_{IH_X}$ | Crystal input HIGH Voltage | | 2 | | 5.25 | V |
| $V_{IL_X}$ | Crystal input LOW Voltage | | –0.05 | | 0.8 | V |
| $I_I$ | Input Leakage Current | 0< $V_{IN}$ < VCC | | | ±10 | ∝A |
| $V_{OH}$ | Output Voltage HIGH | $I_{OUT}$ = 4 mA | 2.4 | | | V |
| $V_{OL}$ | Output LOW Voltage | $I_{OUT}$ = –4 mA | | | 0.4 | V |
| $I_{OH}$ | Output Current HIGH | | | | 4 | mA |
| $I_{OL}$ | Output Current LOW | | | | 4 | mA |
| $C_{IN}$ | Input Pin Capacitance | Except D+/D– | | 3.29 | 10 | pF |
| | | D+/D– | | 12.96 | 15 | pF |
| $I_{SUSP}$ | Suspend Current CY7C64714 | Connected | | 300 | 380[13] | ∝A |
| | | Disconnected | | 100 | 150[13] | ∝A |
| | Suspend Current CY7C64713 | Connected | | .5 | 1.2 | mA |
| | | Disconnected | | .3 | 1.0 | mA |
| $I_{CC}$ | Supply Current | 8051 running, connected to USB | | 35 | 65 | mA |
| $T_{RESET}$ | Reset Time after Valid Power | VCC min = 3.0V | 5.0 | | | ms |
| | Pin Reset after powered on | | 200 | | | ∝s |

### 9.1 USB Transceiver

USB 2.0-compliant in full-speed mode.

**Note:**

12. It is recommended to not power I/O when chip power is off.
13. Measured at Max VCC, 25ºC.

## 10.0 AC Electrical Characteristics

### 10.1 USB Transceiver

USB 2.0-compliant in full-speed mode.

### 10.2 Program Memory Read

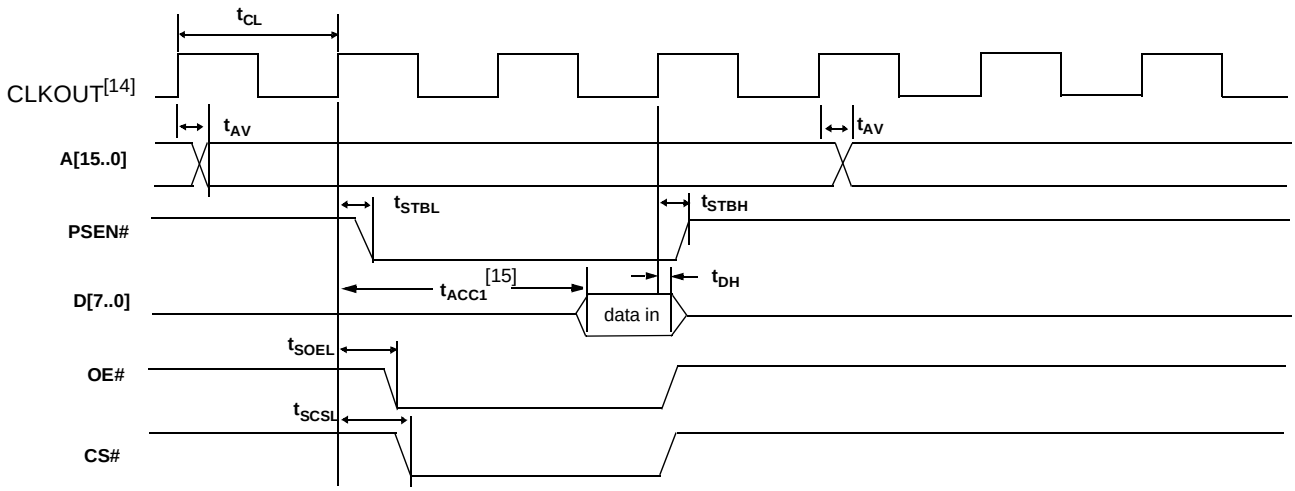



**Figure 10-1. Program Memory Read Timing Diagram**

**Table 10-1. Program Memory Read Parameters**

| Parameter | Description | Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|
| $t_{CL}$ | 1/CLKOUT Frequency | | 20.83 | | ns | 48 MHz |
| | | | 41.66 | | ns | 24 MHz |
| | | | 83.2 | | ns | 12 MHz |
| $t_{AV}$ | Delay from Clock to Valid Address | 0 | | 10.7 | ns | |
| $t_{STBL}$ | Clock to PSEN Low | 0 | | 8 | ns | |
| $t_{STBH}$ | Clock to PSEN High | 0 | | 8 | ns | |
| $t_{SOEL}$ | Clock to OE Low | | | 11.1 | ns | |
| $t_{SCSL}$ | Clock to CS Low | | | 13 | ns | |
| $t_{DSU}$ | Data Setup to Clock | 9.6 | | | ns | |
| $t_{DH}$ | Data Hold Time | 0 | | | ns | |

**Notes:**

14. CLKOUT is shown with positive polarity.
15. $t_{ACC1}$ is computed from the above parameters as follows:
    $t_{ACC1}$(24 MHz) = 3*$t_{CL}$ – $t_{AV}$ – $t_{DSU}$ = 106 ns
    $t_{ACC1}$(48 MHz) = 3*$t_{CL}$ – $t_{AV}$ – $t_{DSU}$ = 43 ns.

## 10.3 Data Memory Read

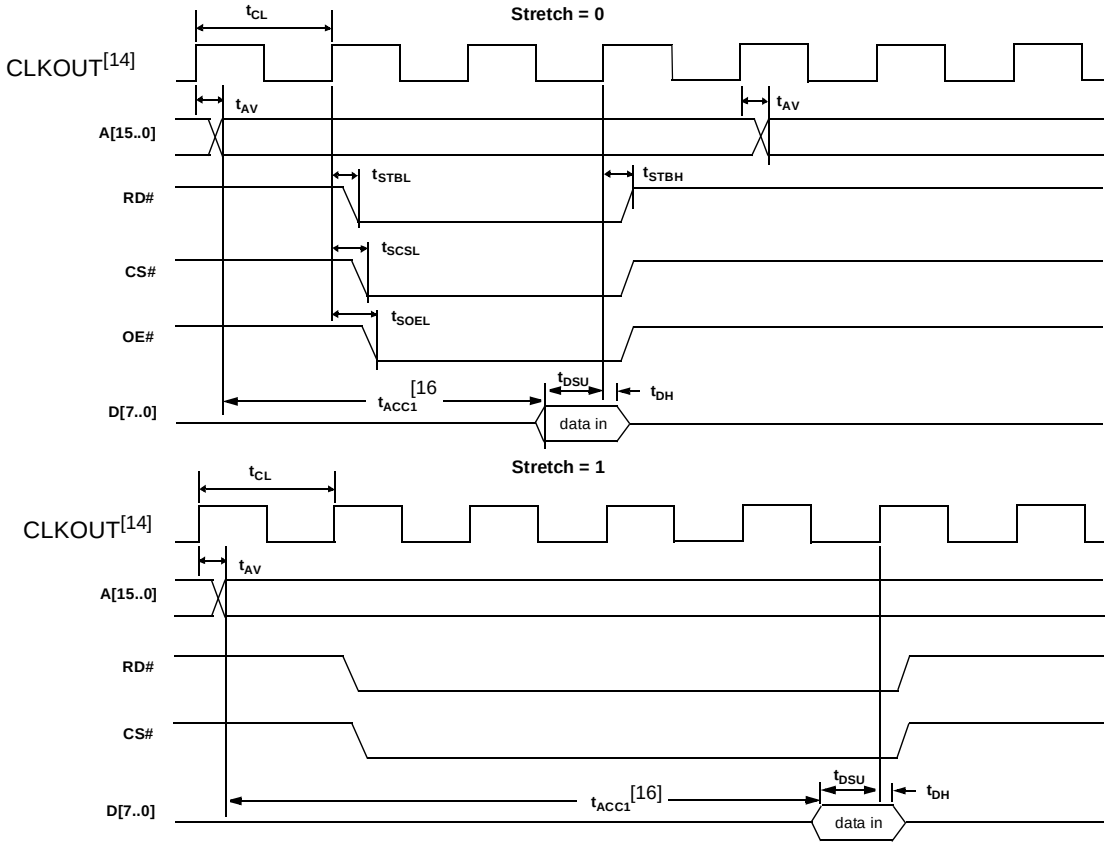



**Figure 10-2. Data Memory Read Timing Diagram**

**Table 10-2. Data Memory Read Parameters**

| Parameter | Description | Min. | Typ. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|
| $t_{CL}$ | 1/CLKOUT Frequency | | 20.83 | | ns | 48 MHz |
| | | | 41.66 | | ns | 24 MHz |
| | | | 83.2 | | ns | 12 MHz |
| $t_{AV}$ | Delay from Clock to Valid Address | | | 10.7 | ns | |
| $t_{STBL}$ | Clock to RD LOW | | | 11 | ns | |
| $t_{STBH}$ | Clock to RD HIGH | | | 11 | ns | |
| $t_{SCSL}$ | Clock to CS LOW | | | 13 | ns | |
| $t_{SOEL}$ | Clock to OE LOW | | | 11.1 | ns | |
| $t_{DSU}$ | Data Setup to Clock | 9.6 | | | ns | |
| $t_{DH}$ | Data Hold Time | 0 | | | ns | |

When using the AUTPOPTR1 or AUTOPTR2 to address external memory, the address of AUTOPTR1 will only be active while either RD# or WR# are active. The address of AUTOPTR2 will be active throughout the cycle and meet the above address valid time for which is based on the stretch value.

**Note:**

16. $t_{ACC2}$ and $t_{ACC3}$ are computed from the above parameters as follows:
$t_{ACC2}$(24 MHz) = 3*$t_{CL}$ – $t_{AV}$ – $t_{DSU}$ = 106 ns
$t_{ACC2}$(48 MHz) = 3*$t_{CL}$ – $t_{AV}$ – $t_{DSU}$ = 43 ns

$t_{ACC3}$(24 MHz) = 5*$t_{CL}$ – $t_{AV}$ – $t_{DSU}$ = 190 ns
$t_{ACC3}$(48 MHz) = 5*$t_{CL}$ – $t_{AV}$ – $t_{DSU}$ = 86 ns.

### 10.4 Data Memory Write

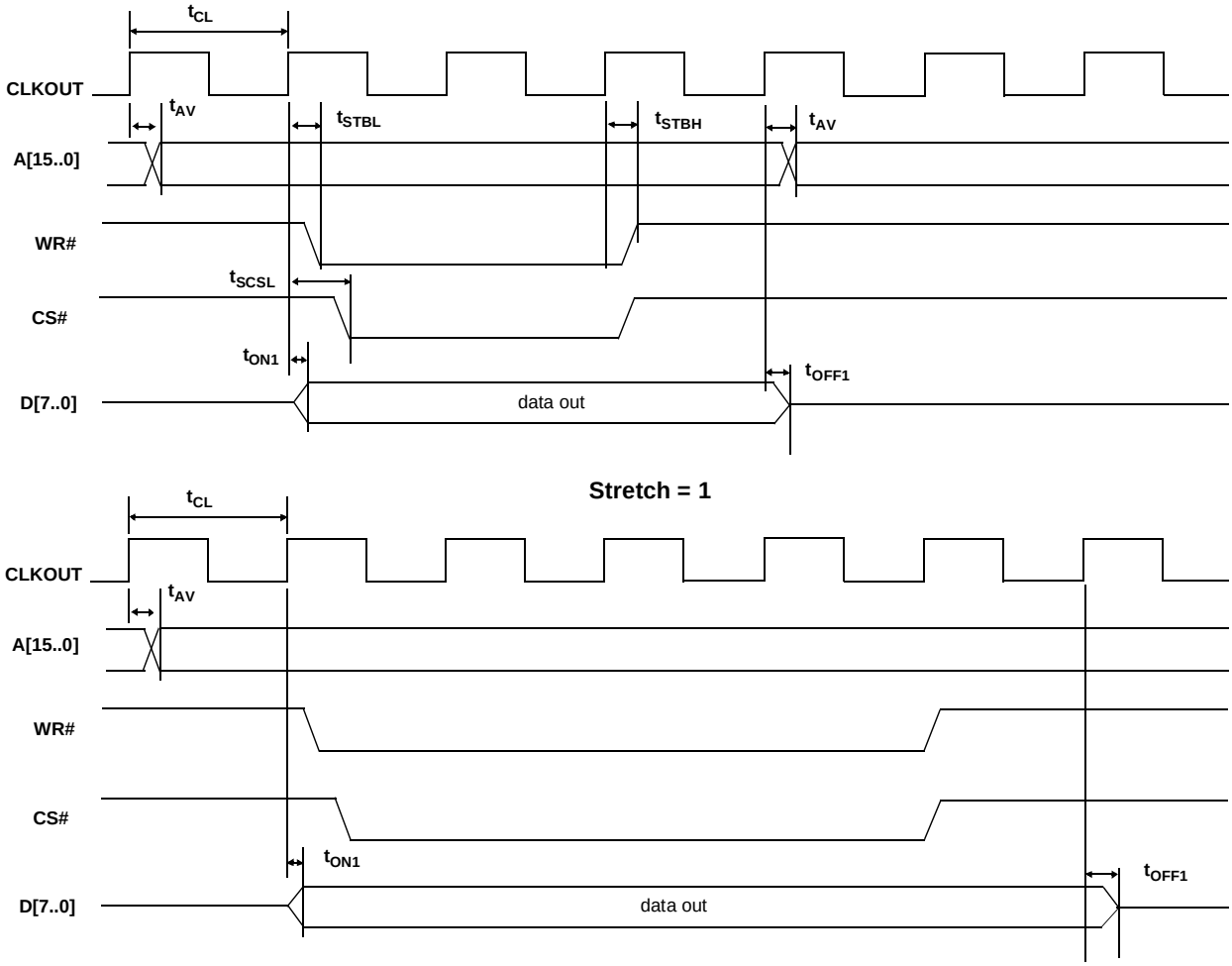



**Figure 10-3. Data Memory Write Timing Diagram**

**Table 10-3. Data Memory Write Parameters**

| Parameter | Description | Min. | Max. | Unit | Notes |
|---|---|---|---|---|---|
| $t_{AV}$ | Delay from Clock to Valid Address | 0 | 10.7 | ns | |
| $t_{STBL}$ | Clock to WR Pulse LOW | 0 | 11.2 | ns | |
| $t_{STBH}$ | Clock to WR Pulse HIGH | 0 | 11.2 | ns | |
| $t_{SCSL}$ | Clock to CS Pulse LOW | | 13.0 | ns | |
| $t_{ON1}$ | Clock to Data Turn-on | 0 | 13.1 | ns | |
| $t_{OFF1}$ | Clock to Data Hold Time | 0 | 13.1 | ns | |

When using the AUTPOPTR1 or AUTOPTR2 to address external memory, the address of AUTOPTR1 will only be active while either RD# or WR# are active. The address of AUTOPTR2 will be active throughout the cycle and meet the above address valid time for which is based on the stretch value.

### 10.5 PORTC Strobe Feature Timings

The RD# and WR# are present in the 100-pin version and the 128-pin package. In these 100-pin and 128-pin versions, an 8051 control bit can be set to pulse the RD# and WR# pins when the 8051 reads from/writes to PORTC. This feature is enabled by setting PORTCSTB bit in CPUCS register.

The RD# and WR# strobes are asserted for two CLKOUT cycles when PORTC is accessed.

The WR# strobe will be asserted two clock cycles after PORTC is updated and will be active for two clock cycles after that as shown in *Figure 10-4*.

As for read, the value of PORTC three clock cycles before the assertion of RD# is the value that the 8051 reads in. The RD# is pulsed for 2 clock cycles after 3 clock cycles from the point when the 8051 has performed a read function on PORTC.

The way the feature is intended to work is that the RD# signal will prompt the external logic to prepare the next data byte. Nothing gets sampled internally on assertion of the RD# signal itself. It is just a “prefetch” type signal to get the next data byte prepared. So, using it with that in mind should easily meet the set-up time to the next read.

The purpose of this pulsing of RD# is to let the external peripheral know that the 8051 is done reading PORTC and the data was latched into PORTC three CLKOUT cycles prior to asserting the RD# signal. Once the RD# is pulsed the external logic may update the data on PORTC.

Following is the timing diagram of the read and write strobing function on accessing PORTC. Refer to Section 10.3 and Section 10.4 for details on propagation delay of RD# and WR# signals.

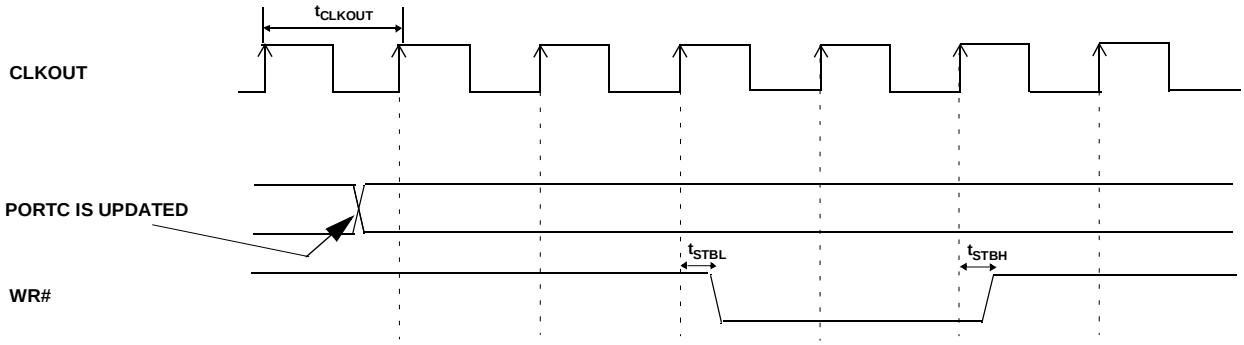



**Figure 10-4. WR# Strobe Function when PORTC is Accessed by 8051**

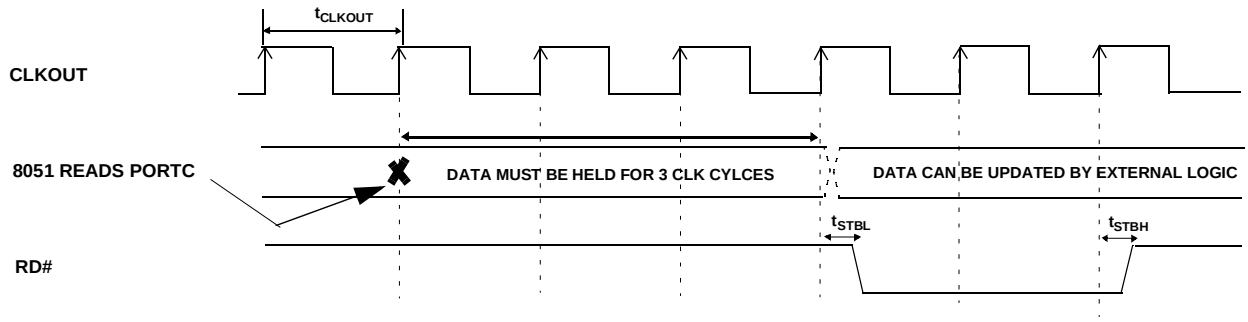



**Figure 10-5. RD# Strobe Function when PORTC is Accessed by 8051**

## 10.6 GPIF Synchronous Signals

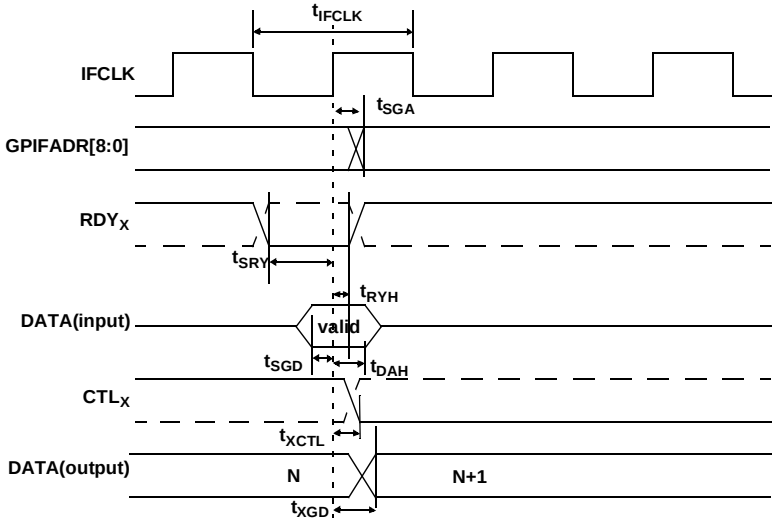



**Figure 10-6. GPIF Synchronous Signals Timing Diagram**[17]

**Table 10-4. GPIF Synchronous Signals Parameters with Internally Sourced IFCLK**[18, 19]

| Parameter | Description | Min. | Max. | Unit |
|---|---|---|---|---|
| $t_{IFCLK}$ | IFCLK Period | 20.83 | | ns |
| $t_{SRY}$ | $RDY_X$ to Clock Setup Time | 8.9 | | ns |
| $t_{RYH}$ | Clock to $RDY_X$ | 0 | | ns |
| $t_{SGD}$ | GPIF Data to Clock Setup Time | 9.2 | | ns |
| $t_{DAH}$ | GPIF Data Hold Time | 0 | | ns |
| $t_{SGA}$ | Clock to GPIF Address Propagation Delay | | 7.5 | ns |
| $t_{XGD}$ | Clock to GPIF Data Output Propagation Delay | | 11 | ns |
| $t_{XCTL}$ | Clock to $CTL_X$ Output Propagation Delay | | 6.7 | ns |

**Table 10-5. GPIF Synchronous Signals Parameters with Externally Sourced IFCLK**[19]

| Parameter | Description | Min. | Max. | Unit |
|---|---|---|---|---|
| $t_{IFCLK}$ | IFCLK Period | 20.83 | 200 | ns |
| $t_{SRY}$ | $RDY_X$ to Clock Setup Time | 2.9 | | ns |
| $t_{RYH}$ | Clock to $RDY_X$ | 3.7 | | ns |
| $t_{SGD}$ | GPIF Data to Clock Setup Time | 3.2 | | ns |
| $t_{DAH}$ | GPIF Data Hold Time | 4.5 | | ns |
| $t_{SGA}$ | Clock to GPIF Address Propagation Delay | | 11.5 | ns |
| $t_{XGD}$ | Clock to GPIF Data Output Propagation Delay | | 15 | ns |
| $t_{XCTL}$ | Clock to $CTL_X$ Output Propagation Delay | | 10.7 | ns |

**Notes:**
17. Dashed lines denote signals with programmable polarity.
18. GPIF asynchronous $RDY_x$ signals have a minimum Setup time of 50 ns when using internal 48-MHz IFCLK.
19. IFCLK must not exceed 48 MHz.

## 10.7 Slave FIFO Synchronous Read

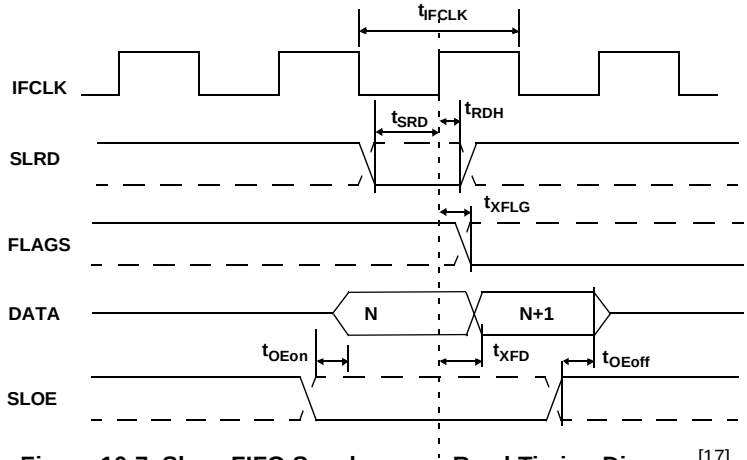



**Figure 10-7. Slave FIFO Synchronous Read Timing Diagram**[17]

**Table 10-6. Slave FIFO Synchronous Read Parameters with Internally Sourced IFCLK**[19]

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

**Table 10-7. Slave FIFO Synchronous Read Parameters with Externally Sourced IFCLK**[19]

| Parameter | Description | Min. | Max. | Unit |
|---|---|---|---|---|
| $t_{IFCLK}$ | IFCLK Period | 20.83 | 200 | ns |
| $t_{SRD}$ | SLRD to Clock Setup Time | 12.7 | | ns |
| $t_{RDH}$ | Clock to SLRD Hold Time | 3.7 | | ns |
| $t_{OEon}$ | SLOE Turn-on to FIFO Data Valid | | 10.5 | ns |
| $t_{OEoff}$ | SLOE Turn-off to FIFO Data Hold | | 10.5 | ns |
| $t_{XFLG}$ | Clock to FLAGS Output Propagation Delay | | 13.5 | ns |
| $t_{XFD}$ | Clock to FIFO Data Output Propagation Delay | | 15 | ns |

### 10.8 Slave FIFO Asynchronous Read

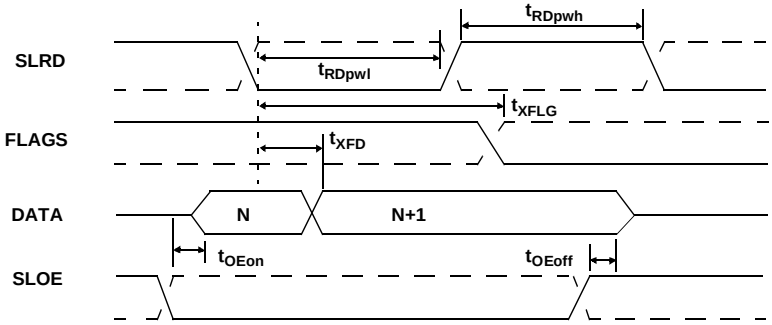



**Figure 10-8. Slave FIFO Asynchronous Read Timing Diagram**[17]

**Table 10-8. Slave FIFO Asynchronous Read Parameters**[20]

| Parameter | Description | Min. | Max. | Unit |
|---|---|---|---|---|
| $t_{RDpwl}$ | SLRD Pulse Width LOW | 50 | | ns |
| $t_{RDpwh}$ | SLRD Pulse Width HIGH | 50 | | ns |
| $t_{XFLG}$ | SLRD to FLAGS Output Propagation Delay | | 70 | ns |
| $t_{XFD}$ | SLRD to FIFO Data Output Propagation Delay | | 15 | ns |
| $t_{OEon}$ | SLOE Turn-on to FIFO Data Valid | | 10.5 | ns |
| $t_{OEoff}$ | SLOE Turn-off to FIFO Data Hold | | 10.5 | ns |

## 10.9 Slave FIFO Synchronous Write

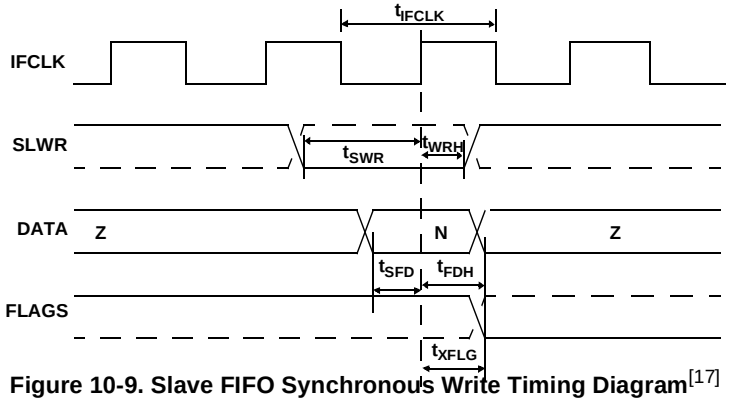



**Figure 10-9. Slave FIFO Synchronous Write Timing Diagram**[17]

**Table 10-9. Slave FIFO Synchronous Write Parameters with Internally Sourced IFCLK** [19]

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

**Table 10-10. Slave FIFO Synchronous Write Parameters with Externally Sourced IFCLK** [19]

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

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

### 10.10 Slave FIFO Asynchronous Write

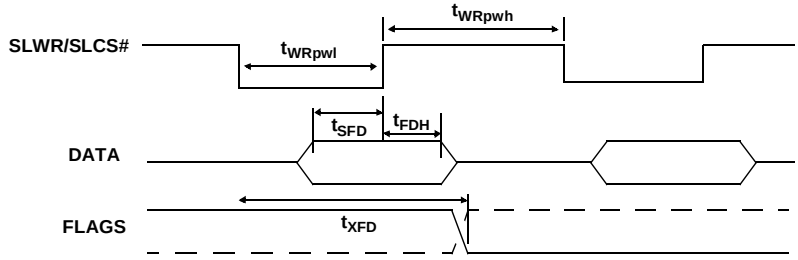



**Figure 10-10. Slave FIFO Asynchronous Write Timing Diagram**[17]

**Table 10-11. Slave FIFO Asynchronous Write Parameters with Internally Sourced IFCLK** [20]

| Parameter | Description | Min. | Max. | Unit |
|---|---|---|---|---|
| $t_{WRpwl}$ | SLWR Pulse LOW | 50 | | ns |
| $t_{WRpwh}$ | SLWR Pulse HIGH | 70 | | ns |
| $t_{SFD}$ | SLWR to FIFO DATA Setup Time | 10 | | ns |
| $t_{FDH}$ | FIFO DATA to SLWR Hold Time | 10 | | ns |
| $t_{XFD}$ | SLWR to FLAGS Output Propagation Delay | | 70 | ns |

### 10.11 Slave FIFO Synchronous Packet End Strobe

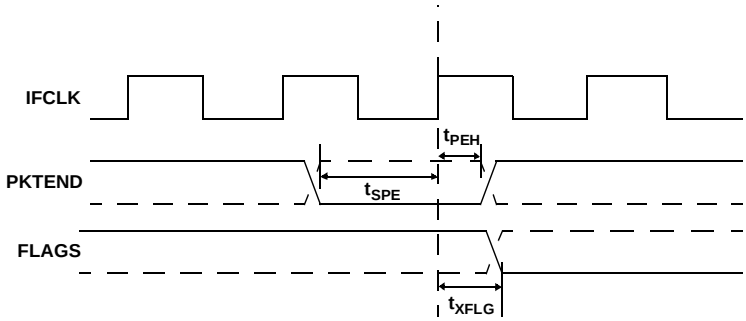



**Figure 10-11. Slave FIFO Synchronous Packet End Strobe Timing Diagram**[17]

**Table 10-12. Slave FIFO Synchronous Packet End Strobe Parameters with Internally Sourced IFCLK** [19]

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

**Table 10-13. Slave FIFO Synchronous Packet End Strobe Parameters with Externally Sourced IFCLK** [19]

| Parameter | Description | Min. | Max. | Unit |
|---|---|---|---|---|
| $t_{IFCLK}$ | IFCLK Period | 20.83 | 200 | ns |
| $t_{SPE}$ | PKTEND to Clock Setup Time | 8.6 | | ns |
| $t_{PEH}$ | Clock to PKTEND Hold Time | 2.5 | | ns |
| $t_{XFLG}$ | Clock to FLAGS Output Propagation Delay | | 13.5 | ns |

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

Although typically there are no specific timing requirements for asserting PKTEND in relation to SLWR, there exists a specific corner case condition that needs attention. While using the PKTEND to commit a one byte/word packet, an additional timing requirement needs to be met when the FIFO is

configured to operate in auto mode and it is desired 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 developer 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 10-12* below 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 10-12* 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 being committed manually using PKTEND. Note that there is at least one IFCLK cycle timing between asserting PKTEND and clocking of the last byte of the previous packet (causing the packet to be committed automatically). Failing to adhere to this timing, will result in the FX2 failing to send the one byte/word short packet.

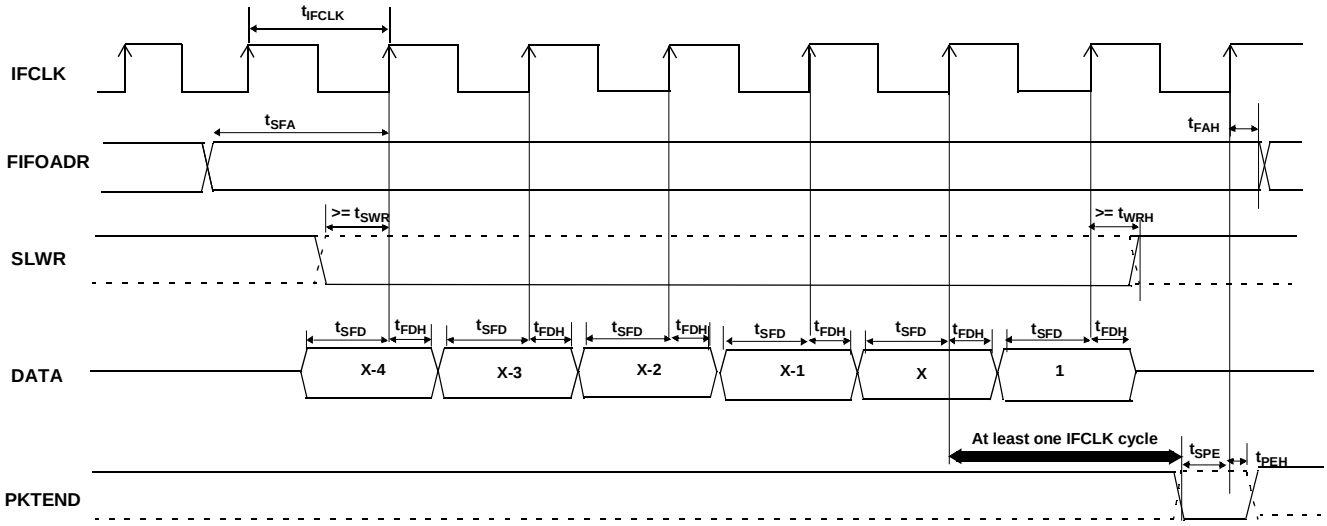



**Figure 10-12. Slave FIFO Synchronous Write Sequence and Timing Diagram**

## 10.12 Slave FIFO Asynchronous Packet End Strobe

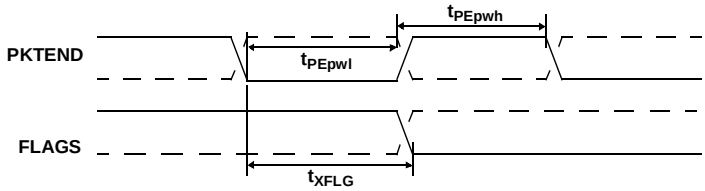



**Figure 10-13. Slave FIFO Asynchronous Packet End Strobe Timing Diagram**[17]

**Table 10-14. Slave FIFO Asynchronous Packet End Strobe Parameters**[20]

| Parameter | Description | Min. | Max. | Unit |
|---|---|---|---|---|
| $t_{PEpwl}$ | PKTEND Pulse Width LOW | 50 | | ns |
| $t_{PWpwh}$ | PKTEND Pulse Width HIGH | 50 | | ns |
| $t_{XFLG}$ | PKTEND to FLAGS Output Propagation Delay | | 115 | ns |

### 10.13 Slave FIFO Output Enable

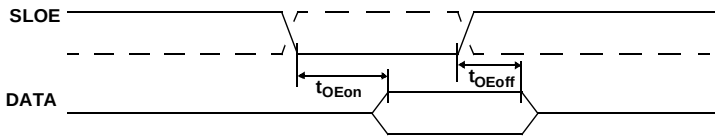



**Figure 10-14. Slave FIFO Output Enable Timing Diagram**[17]

**Table 10-15. Slave FIFO Output Enable Parameters**

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

### 10.14 Slave FIFO Address to Flags/Data

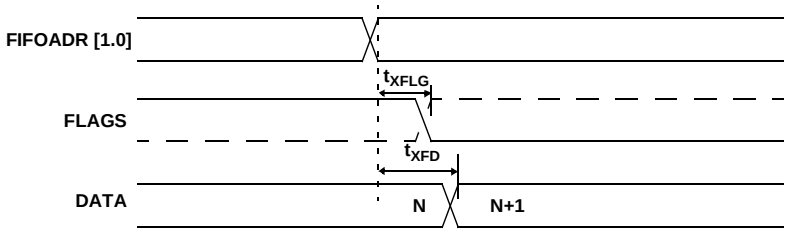



**Figure 10-15. Slave FIFO Address to Flags/Data Timing Diagram**[17]

**Table 10-16. 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 |

### 10.15 Slave FIFO Synchronous Address

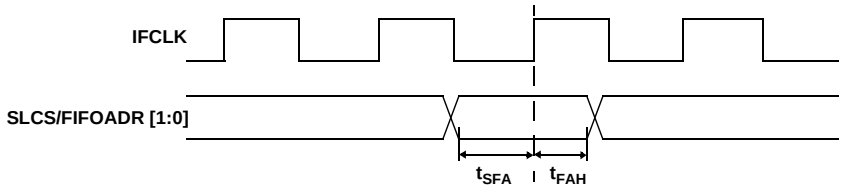



**Figure 10-16. Slave FIFO Synchronous Address Timing Diagram**

**Table 10-17. Slave FIFO Synchronous Address Parameters** [19]

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

### 10.16 Slave FIFO Asynchronous Address

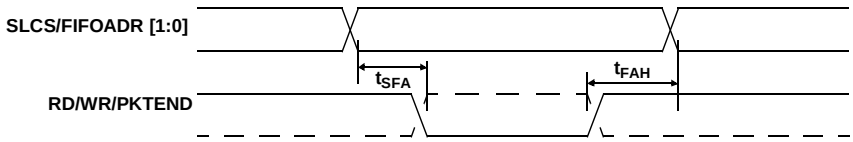



**Figure 10-17. Slave FIFO Asynchronous Address Timing Diagram**[17]

**Table 10-18. Slave FIFO Asynchronous Address Parameters**[20]

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

## 10.17 Sequence Diagram

### *10.17.1 Single and Burst Synchronous Read Example*

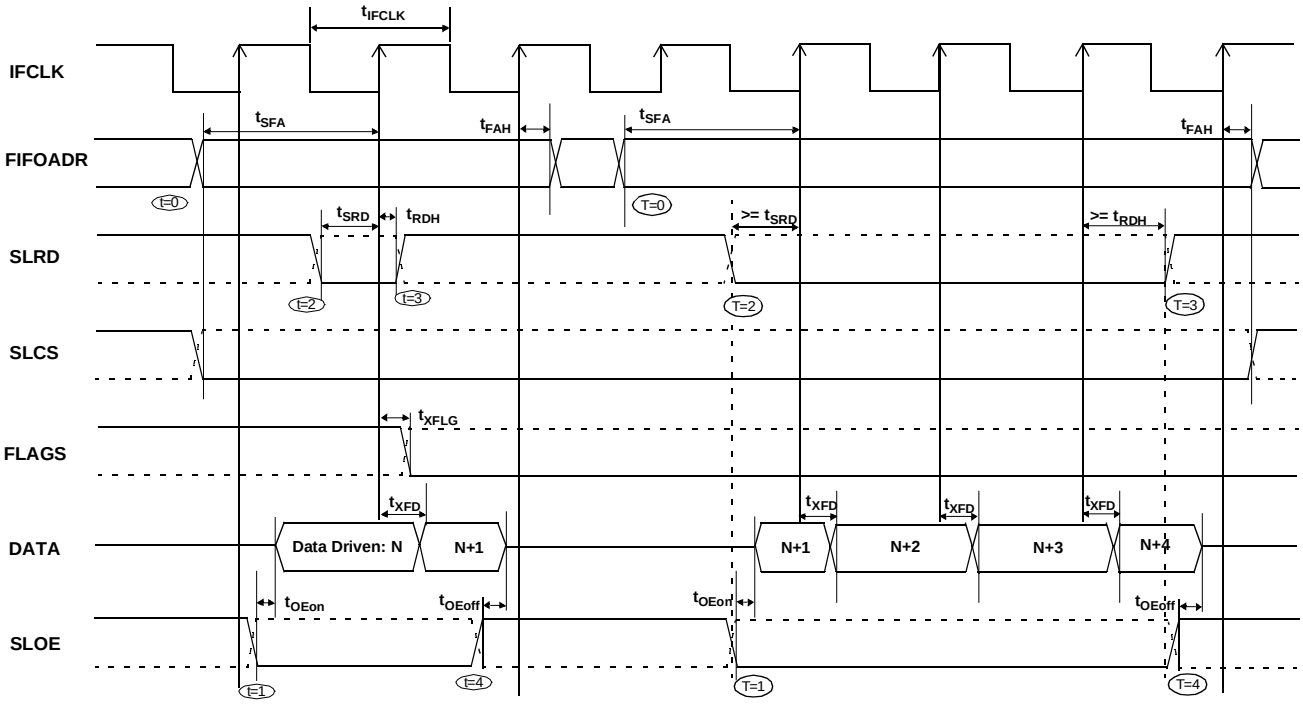



**Figure 10-18. Slave FIFO Synchronous Read Sequence and Timing Diagram**

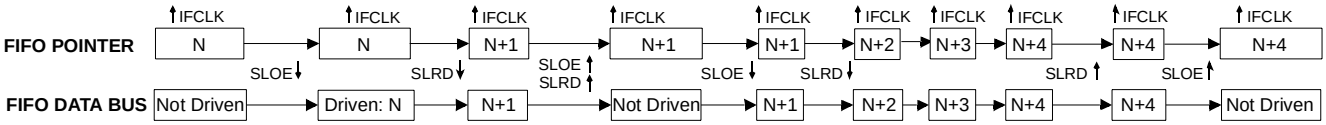



**Figure 10-19. Slave FIFO Synchronous Sequence of Events Diagram**

*Figure 10-18* 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 when IFCLK is running at 48 MHz, the FIFO address setup 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 setup 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 (i.e. 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 are shown for a burst read and are 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.

*10.17.2 Single and Burst Synchronous Write*

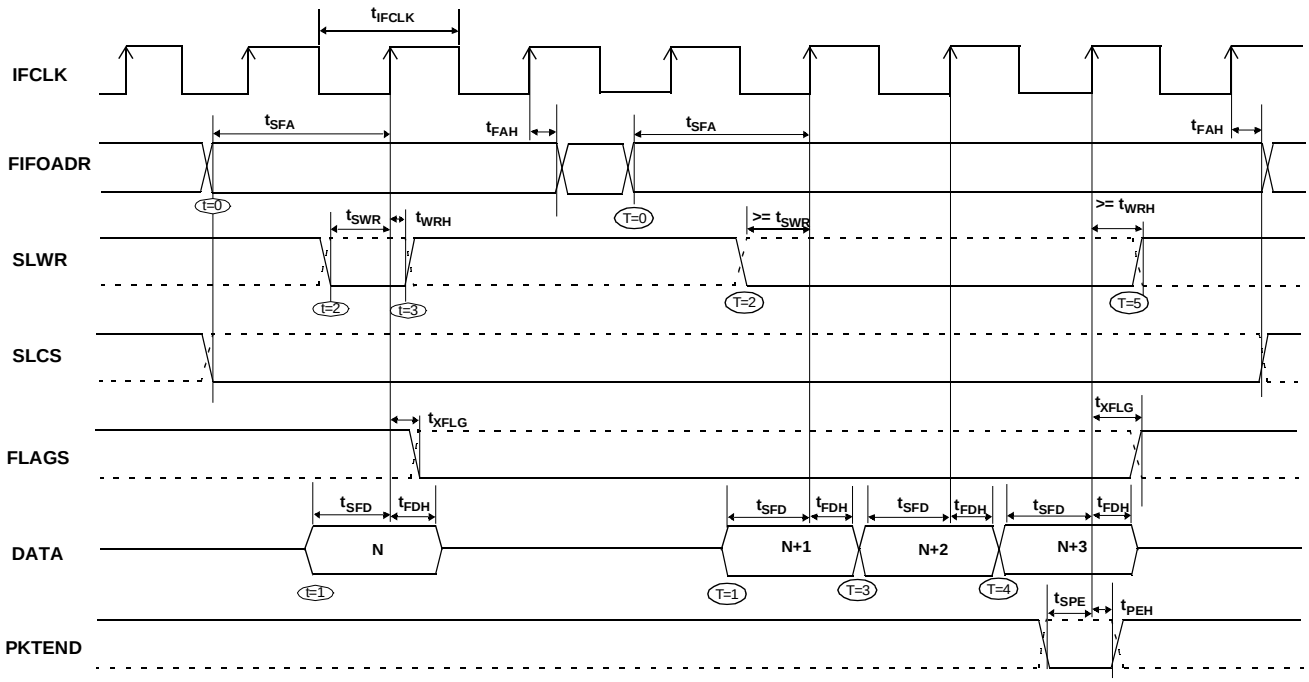



**Figure 10-20. Slave FIFO Synchronous Write Sequence and Timing Diagram**[17]

The *Figure 10-20* 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 when IFCLK is running at 48 MHz, the FIFO address setup time is more than one IFCLK cycle.
- At t = 1, the external master/peripheral must outputs 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 setup 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 SLCS signal is used, it must be asserted with SLWR or before SLWR is asserted. (i.e., 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 will also be updated after a delay of $t_{XFLG}$ from the rising edge of the clock.

The same sequence of events are also shown for a burst write and are 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 10-20*, once the four bytes are written to the FIFO, SLWR is de-asserted. 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 PKTEND signal with regards to asserting the SLWR signal. PKTEND can be asserted with the last data value or thereafter. The only consideration is the setup time $t_{SPE}$ and the hold time $t_{PEH}$ must be met. In the scenario of *Figure 10-20*, 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 be asserted in subsequent clock cycles. The FIFOADDR lines should be held constant during the PKTEND assertion.

Although there are no specific timing requirement for asserting PKTEND, there is a specific corner case condition that needs attention while using the PKTEND to commit a one byte/word packet. Additional timing requirements exists when the FIFO is configured to operate in auto mode and it is desired 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 section 10-10 for further details on this timing.

*10.17.3 Sequence Diagram of a Single and Burst Asynchronous Read*

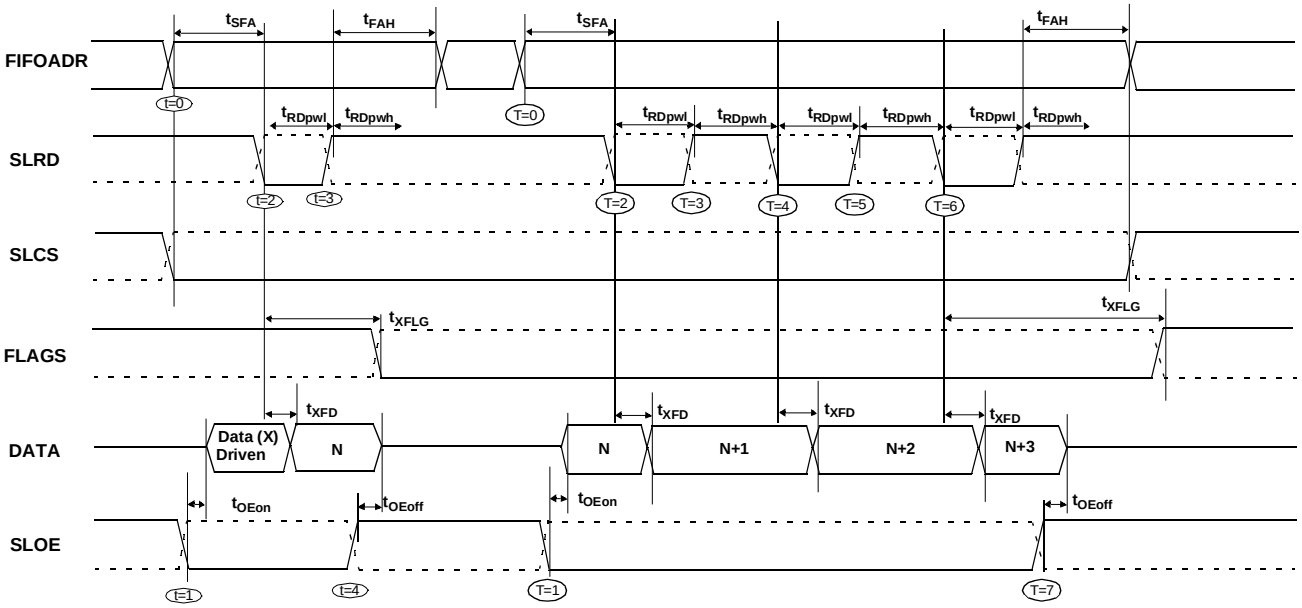



**Figure 10-21. Slave FIFO Asynchronous Read Sequence and Timing Diagram**

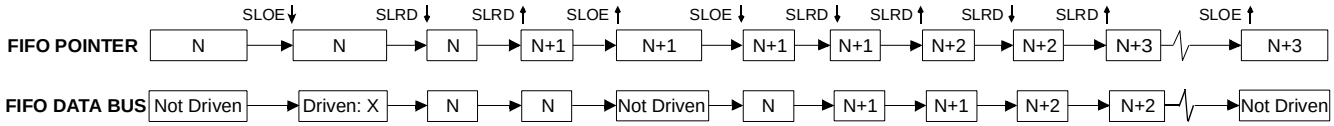



**Figure 10-22. Slave FIFO Asynchronous Read Sequence of Events Diagram**

*Figure 10-21* diagrams 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 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 in asserted with SLRD or before SLRD is asserted. (i.e., the SLCS and SLRD signals must both be asserted to start a valid read condition.)
- The data that will be 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 10-21*, data N is the first valid data read from the FIFO. For data to appear on the data bus during the read cycle (i.e. 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 is 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.

*10.17.4 Sequence Diagram of a Single and Burst Asynchronous Write*

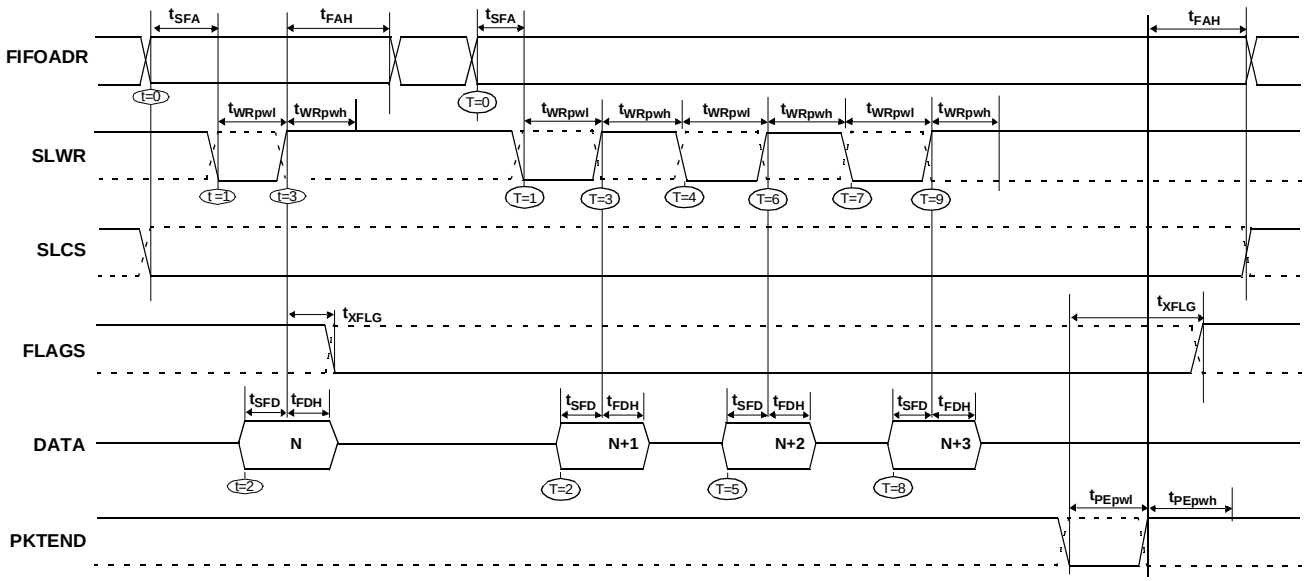



**Figure 10-23. Slave FIFO Asynchronous Write Sequence and Timing Diagram**[17]

*Figure 10-23* diagrams 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, insuring that it meets the setup 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 in 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 will cause the data to be written from the data bus to the FIFO and then increments the FIFO pointer. The FIFO flag is also updated after $t_{XFLG}$ from the de-asserting edge of SLWR.

The same sequence of events are 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 10-23*, 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 should be designed to not assert SLWR and the PKTEND signal at the same time. It should be designed to assert the PKTEND after SLWR is deasserted and met the minimum deasserted pulse width. The FIFOADDR lines are to be held constant during the PKTEND assertion.

## 11.0 Ordering Information

**Table 11-1. Ordering Information**

| Ordering Code | Package Type | RAM Size | # Prog I/Os | 8051 Address /Data Busses |
|---|---|---|---|---|
| **Ideal for battery powered applications** | | | | |
| CY7C64714-128AXC | 128 TQFP – Lead-Free | 16K | 40 | 16/8 bit |
| CY7C64714-100AXC | 100 TQFP – Lead-Free | 16K | 40 | – |
| CY7C64714-56LFXC | 56 QFN – Lead-Free | 16K | 24 | – |
| **Ideal for non-battery powered applications** | | | | |
| CY7C64713-128AXC | 128 TQFP - Lead-Free | 16K | 40 | 16/8 bit |
| CY7C64713-100AXC | 100 TQFP - Lead-Free | 16K | 40 | - |
| CY7C64713-56LFXC | 56 QFN - Lead-Free | 16K | 24 | - |
| CY3674 | EZ-USB FX1 Development Kit | | | |

## 12.0 Package Diagrams

The FX1 is available in three packages:

- 56-pin QFN
- 100-pin TQFP
- 128-pin TQFP

## Package Diagrams

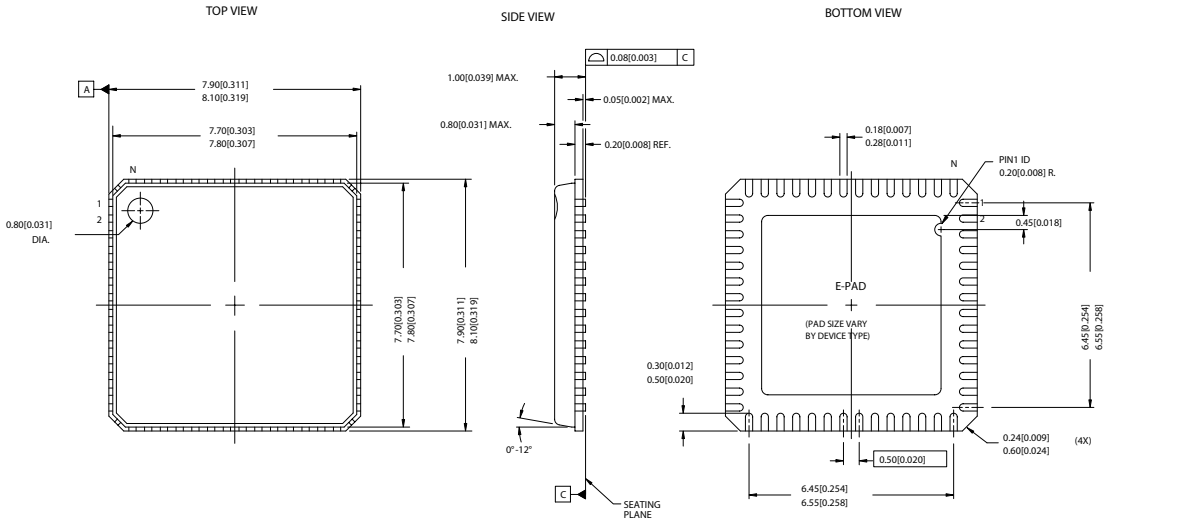



**Figure 12-1. 56-Lead QFN 8 x 8 mm LF56A**

**Package Diagrams** (continued)

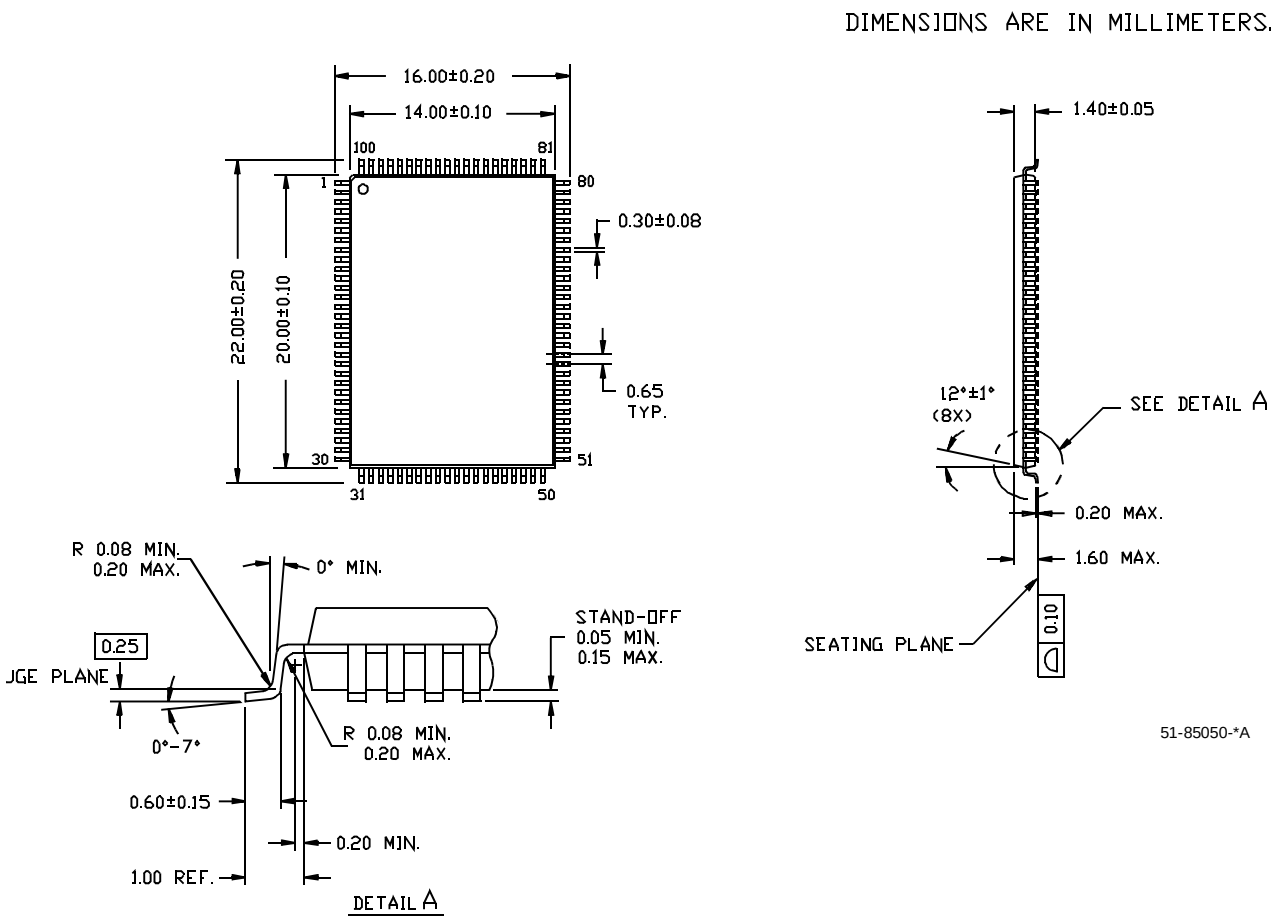
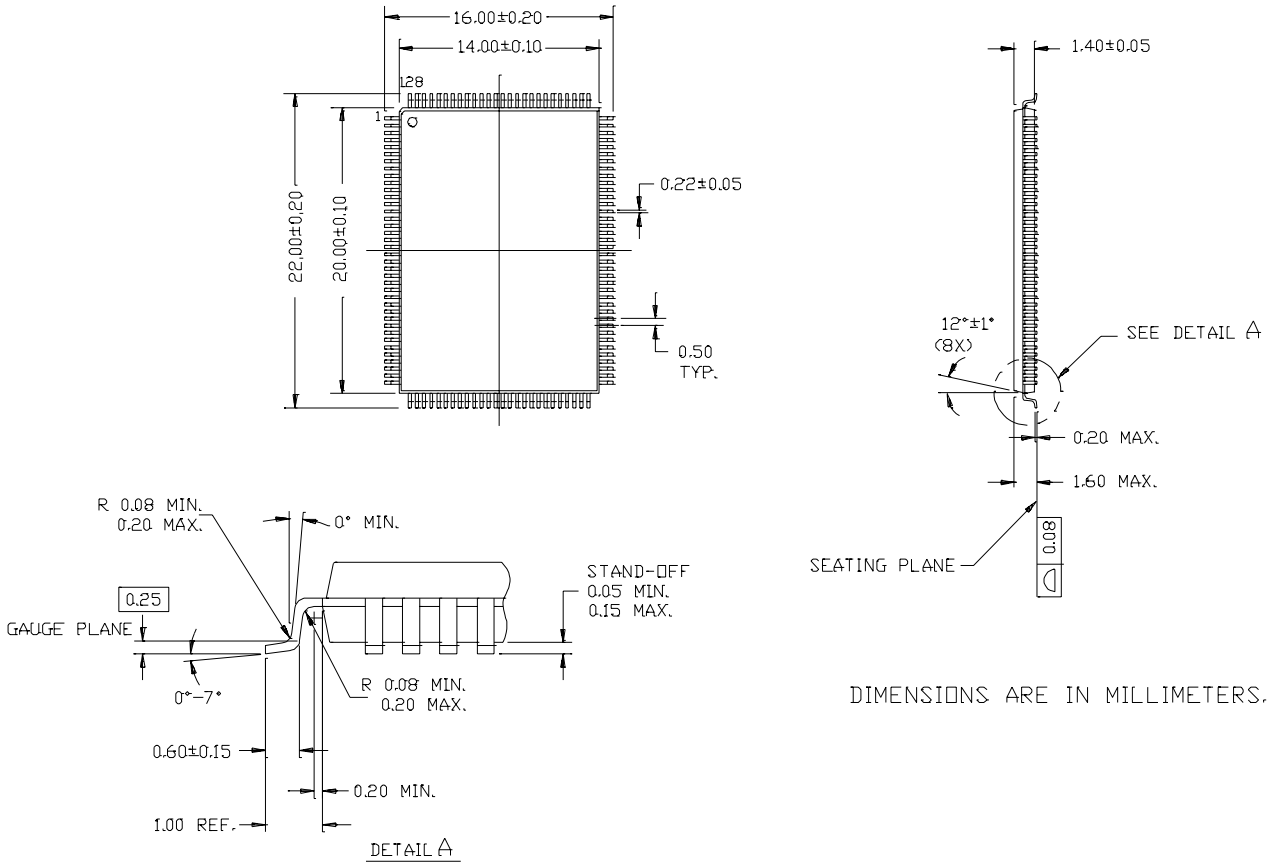



**Figure 12-2. 100-Pin Thin Plastic Quad Flatpack (14 x 20 x 1.4 mm) A101**

**Package Diagrams** (continued)




51-85101-*B

**Figure 12-3. 128-Lead Thin Plastic Quad Flatpack (14 x 20 x 1.4 mm) A128**

## 13.0 Quad Flat Package No Leads (QFN) Package Design Notes

Electrical contact of the part to the Printed Circuit Board (PCB) is made by soldering the leads on the bottom surface of the package to the PCB. Hence, special attention is required to the heat transfer area below the package to provide a good thermal bond to the circuit board. A Copper (Cu) fill is to be designed into the PCB as a thermal pad under the package. Heat is transferred from the FX1 through the device's metal paddle on the bottom side of the package. Heat from here, is conducted to the PCB at the thermal pad. It is then conducted from the thermal pad to the PCB inner ground plane by a 5 x 5 array of via. A via is a plated through hole in the PCB with a finished diameter of 13 mil. The QFN's metal die paddle must be soldered to the PCB's thermal pad. Solder mask is placed on the board top side over each via to resist solder flow into the via. The mask on the top side also minimizes outgassing during the solder reflow process.

For further information on this package design please refer to the application note *Surface Mount Assembly of AMKOR's MicroLeadFrame (MLF) Technology*. This application note can be downloaded from AMKOR's website from the following URL:

http://www.amkor.com/products/notes_papers/MLF_AppNote_0902.pdf.

The application note provides detailed information on board mounting guidelines, soldering flow, rework process, etc.

*Figure 13-1* below displays a cross-sectional area underneath the package. The cross section is of only one via. The solder paste template needs to be designed to allow at least 50% solder coverage. The thickness of the solder paste template should be 5 mil. It is recommended that "No Clean" type 3 solder paste is used for mounting the part. Nitrogen purge is recommended during reflow.

*Figure 13-2* is a plot of the solder mask pattern and *Figure 13-3* displays an X-Ray image of the assembly (darker areas indicate solder).

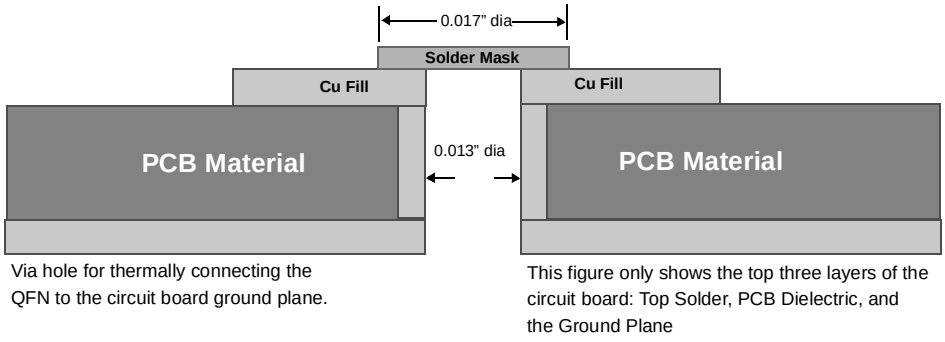



**Figure 13-1. Cross-section of the Area Underneath the QFN Package**

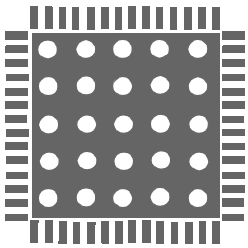


**Figure 13-2. Plot of the Solder Mask (White Area)**

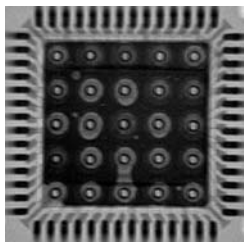


**Figure 13-3. X-ray Image of the Assembly**

Purchase of $I^2C$ components from Cypress, or one of its sublicensed Associated Companies, conveys a license under the Philips $I^2C$ Patent Rights to use these components in an $I^2C$ system, provided that the system conforms to the $I^2C$ Standard Specification as defined by Philips. EZ-USB FX1, EZ-USB FX2LP, EZ-USB FX2, and ReNumeration are trademarks, and EZ-USB is a registered trademark, of Cypress Semiconductor. All product and company names mentioned in this document are the trademarks of their respective holders.



© Cypress Semiconductor Corporation, 2005. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

## Document History Page

| Document Title: CY7C64713/4 EZ-USB FX1™ USB Microcontroller Full-Speed USB Peripheral Controller<br>Document Number: 38-08039 | | | | |
|---|---|---|---|---|
| **REV.** | **ECN NO.** | **Issue Date** | **Orig. of Change** | **Description of Change** |
| ** | 132091 | 02/10/04 | KKU | New Data Sheet |
| *A | 230709 | SEE ECN | KKU | Changed Lead free Marketing part numbers in *Table 11-1* according to spec change in 28-00054. |
| *B | 307474 | SEE ECN | BHA | Changed default PID in Table 4-2.<br>Updated register table.<br>Removed word compatible where associated with I2C.<br>Changed Set-up to Setup.<br>Added Power Dissipation.<br>Changed Vcc from ± 10% to ± 5%<br>Added values for $V_{IH_X}$, $V_{IL_X}$<br>Added values for $I_{CC}$<br>Added values for $I_{SUSP}$<br>Removed $I_{UNCONFIGURED}$ from table 9-1<br>Changed PKTEND to FLAGS output propagation delay (asynchronous interface) in Table 10-14 from a maximum value of 70 ns to 115 ns.<br>Removed 56 SSOP and added 56 QFN package<br>Provided additional timing restrictions and requirement regarding the use of PKTEND pin to commit a short one byte/word packet subsequent to committing a packet automatically (when in auto mode).<br>Added part number CY7C64714 ideal for battery powered applications.<br>Changed Supply Voltage in section 8 to read +3.15V to +3.45V<br>Added Min Vcc Ramp Up time (0 to 3.3v)<br>Removed Preliminary |
| *C | 392702 | SEE ECN | BHA | Corrected signal name for pin 54 in Figure 5-4.<br>Added information on the AUTOPTR1/AUTOPTR2 address timing with regards to data memory read/write timing diagram.<br>Removed TBD in Table 10-6.<br>Added Section 10-5. |



© Cypress Semiconductor Corporation, 2005. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.