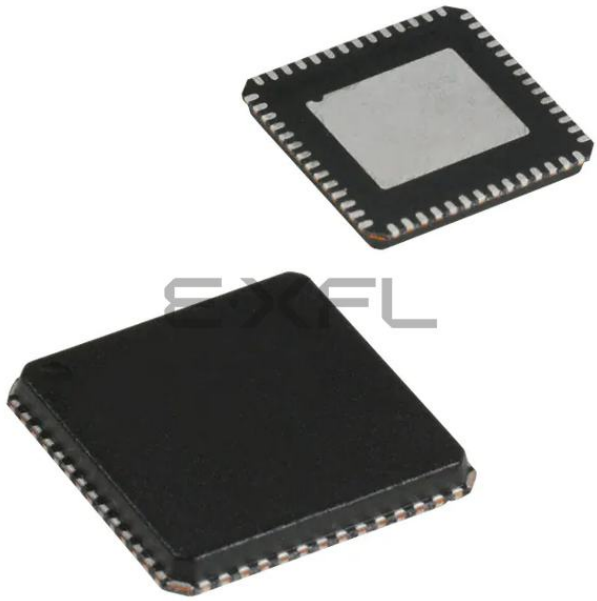


**Welcome to E-XFL.COM**

## Embedded - Microcontrollers - Application Specific: Tailored Solutions for Precision and Performance

**Embedded - Microcontrollers - Application Specific** represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering enhanced performance, efficiency, and functionality to meet the demands of specialized applications.

## What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

| Details | |
|---|---|
| Product Status | Obsolete |
| Applications | USB Microcontroller |
| Core Processor | 8051 |
| Program Memory Type | ROMless |
| Controller Series | CY7C680xx |
| RAM Size | 16K x 8 |
| Interface | I²C, USB, USART |
| Number of I/O | 24 |
| Voltage - Supply | 3V ~ 3.6V |
| Operating Temperature | 0°C ~ 70°C |
| Mounting Type | Surface Mount |
| Package / Case | 56-VFQFN Exposed Pad |
| Supplier Device Package | 56-QFN (8x8) |
| Purchase URL | https://www.e-xfl.com/product-detail/infineon-technologies/cy7c68014a-56lfxc |

## Contents

The FX2LP jump instruction is encoded as follows:.

**Table 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 (or microframe) |
| 3 | 08 | SUTOK | Setup token received |
| 4 | 0C | SUSPEND | USB suspend request |
| 5 | 10 | USB RESET | Bus reset |
| 6 | 14 | HISPEED | Entered high speed operation |
| 7 | 18 | EP0ACK | FX2LP 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 |

If Autovectoring is enabled (AV2EN = 1 in the INTSET-UP register), the FX2LP substitutes its INT2VEC byte. Therefore, if the high byte ("page") of a jump table address is preloaded at the location 0x0044, the automatically inserted INT2VEC byte at 0x0045 directs the jump to the correct address out of the 27 addresses within the page.

*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, similar to the USB Interrupt, can employ autovectoring.

Table 4 on page 8 shows the priority and INT4VEC values for the 14 FIFO/GPIF interrupt sources.

In the Slave (S) mode, FX2LP 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) that enables data of the selected width. External logic must ensure 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#.

*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 inverts the IFCLK signal whether internally or externally sourced.

## GPIF

The GPIF is a flexible 8-bit or 16-bit parallel interface driven by a user-programmable finite state machine. It enables the CY7C68013A/15A 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 is executed to perform the desired data move between the FX2LP and the external device.

*Six Control OUT Signals*

The 100-pin 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).

*Six Ready IN Signals*

The 100-pin 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.

*Nine GPIF Address OUT Signals*

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

*Long Transfer Mode*

In the 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.

## ECC Generation[8]

The EZ-USB 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.

*ECC Implementation*

The two ECC configurations are selected by the ECCM bit:

**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 is calculated and stored in ECC1. The ECC for the next 256 bytes is stored in ECC2. After the second ECC is calculated, the values in the ECCx registers do not change until ECCRESET is written again, even if more data is subsequently passed across the interface.

**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 is calculated and stored in ECC1; ECC2 is unused. After the ECC is calculated, the values in ECC1 do not change even if more data is subsequently passed across the interface, till ECCRESET is written again.

## USB Uploads and Downloads

The core has the ability to directly edit the data contents of the internal 16-KB RAM and of the internal 512-byte scratch pad RAM via a vendor-specific command. This capability is normally used when soft downloading the user code and is available only to and from the internal RAM, only when the 8051 is held in reset. The available RAM spaces are 16 KB from 0x0000–0x3FFF (code/data) and 512 bytes from 0xE000–0xE1FF (scratch pad data RAM)[9].

**Notes**
8. To use the ECC logic, the GPIF or Slave FIFO interface must be configured for byte-wide operation.
9. After the data is downloaded from the host, a “loader” can execute from internal RAM to transfer downloaded data to external memory.

## Pin Assignments

Figure 6 on page 17 identifies all signals for the five 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-pin, 100-pin, and 56-pin packages.

The signals on the left edge of the 56-pin package in Figure 6 are common to all versions in the FX2LP family with the noted differences between the CY7C68013A/14A and the CY7C68015A/16A.

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:

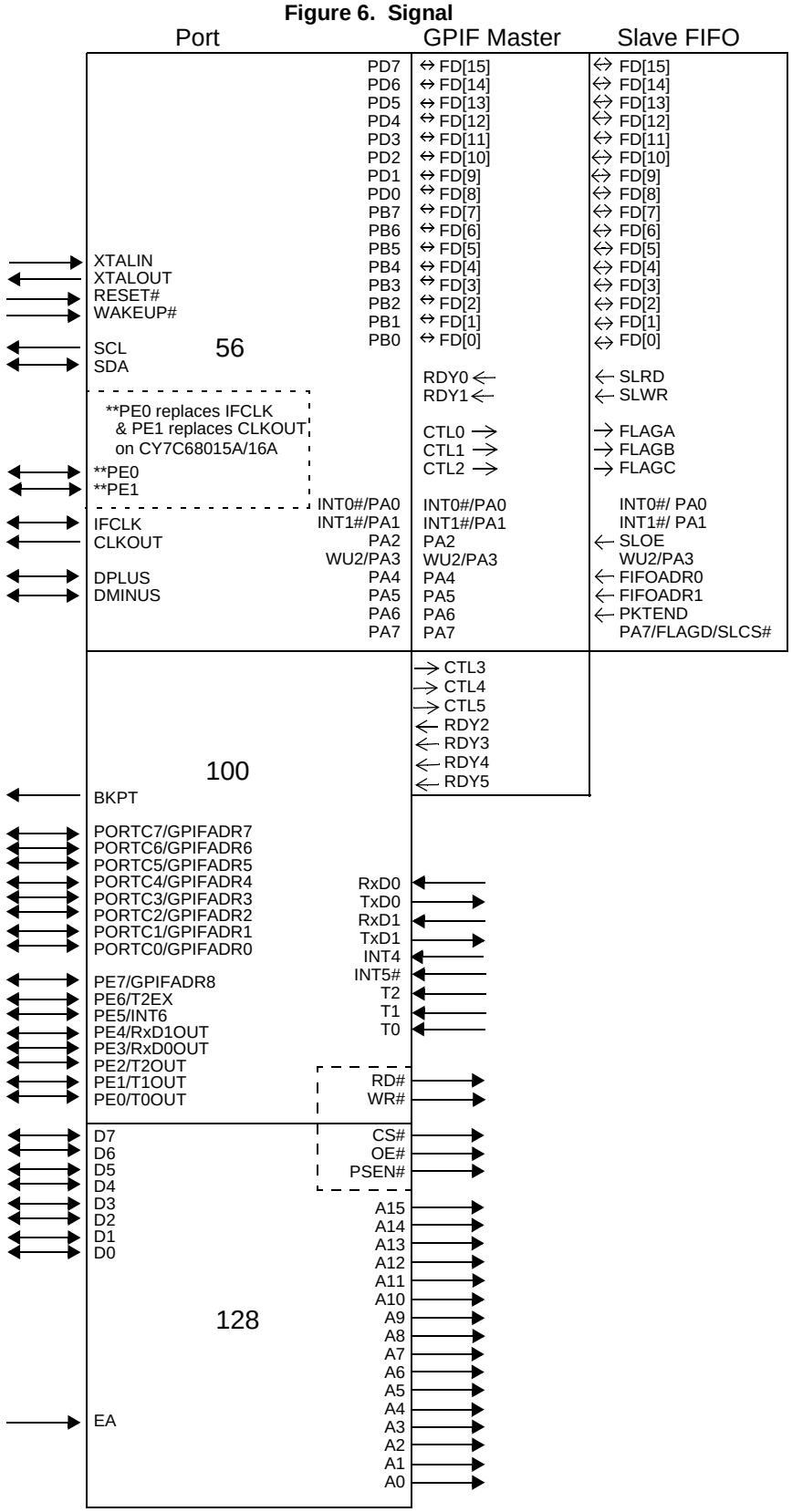
- PORTC or alternate GPIFADR[7:0] address signals
- PORTE or alternate GPIFADR[8] address signal and seven additional 8051 signals
- Three GPIF Control signals
- Four GPIF Ready signals
- Nine 8051 signals (two USARTs, three timer inputs, INT4, and INT5#)
- BKPT, RD#, WR#.

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. This feature is enabled by setting the PORTCSTB bit in the CPUCS register.

PORTC Strobe Feature Timings displays the timing diagram of the read and write strobing function on accessing PORTC.

**Figure 6. Signal**



## CY7C68013A/15A Pin Descriptions

**Table 11. FX2LP Pin Descriptions**[11]

| 128 TQFP | 100 TQFP | 56 SSOP | 56 QFN | 56 VFBGA | Name | Type | Default | Reset[12] | Description |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 9 | 10 | 3 | 2D | AVCC | Power | N/A | N/A | **Analog VCC**. Connect this pin to the 3.3 V power source. This signal provides power to the analog section of the chip. |
| 17 | 16 | 14 | 7 | 1D | AVCC | Power | N/A | N/A | **Analog VCC**. Connect this pin to the 3.3 V power source. This signal provides power to the analog section of the chip. |
| 13 | 12 | 13 | 6 | 2F | AGND | Ground | N/A | N/A | **Analog Ground**. Connect to ground with as short a path as possible. |
| 20 | 19 | 17 | 10 | 1F | AGND | Ground | N/A | N/A | **Analog Ground**. Connect to ground with as short a path as possible. |
| 19 | 18 | 16 | 9 | 1E | DMINUS | I/O/Z | Z | N/A | **USB D– Signal**. Connect to the USB D– signal. |
| 18 | 17 | 15 | 8 | 2E | DPLUS | I/O/Z | Z | N/A | **USB D+ Signal**. Connect to the USB D+ signal. |
| 94 | – | – | – | – | A0 | Output | L | 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 | L | |
| 96 | – | – | – | – | A2 | Output | L | L | |
| 97 | – | – | – | – | A3 | Output | L | L | |
| 117 | – | – | – | – | A4 | Output | L | L | |
| 118 | – | – | – | – | A5 | Output | L | L | |
| 119 | – | – | – | – | A6 | Output | L | L | |
| 120 | – | – | – | – | A7 | Output | L | L | |
| 126 | – | – | – | – | A8 | Output | L | L | |
| 127 | – | – | – | – | A9 | Output | L | L | |
| 128 | – | – | – | – | A10 | Output | L | L | |
| 21 | – | – | – | – | A11 | Output | L | L | |
| 22 | – | – | – | – | A12 | Output | L | L | |
| 23 | – | – | – | – | A13 | Output | L | L | |
| 24 | – | – | – | – | A14 | Output | L | L | |
| 25 | – | – | – | – | A15 | Output | L | L | |
| 59 | – | – | – | – | D0 | I/O/Z | 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 | Z | |
| 61 | – | – | – | – | D2 | I/O/Z | Z | Z | |
| 62 | – | – | – | – | D3 | I/O/Z | Z | Z | |
| 63 | – | – | – | – | D4 | I/O/Z | Z | Z | |
| 86 | – | – | – | – | D5 | I/O/Z | Z | Z | |
| 87 | – | – | – | – | D6 | I/O/Z | Z | Z | |
| 88 | – | – | – | – | D7 | I/O/Z | Z | Z | |
| 39 | – | – | – | – | PSEN# | Output | H | 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. |

**Notes**
11. Unused inputs must not be left floating. Tie either HIGH or LOW as appropriate. Outputs should only be pulled up or down to ensure signals at power up and in standby. Note also that no pins should be driven while the device is powered down.
12. The Reset column indicates the state of signals during reset (RESET# asserted) or during Power on Reset (POR).

**Table 11. FX2LP Pin Descriptions**[11] (continued)

| 128 TQFP | 100 TQFP | 56 SSOP | 56 QFN | 56 VFBGA | Name | Type | Default | Reset[12] | Description |
|---|---|---|---|---|---|---|---|---|---|
| 110 | 88 | – | – | – | PE2 or T2OUT | I/O/Z | I (PE2) | Z (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. |
| 111 | 89 | – | – | – | PE3 or RXD0OUT | I/O/Z | I (PE3) | Z (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) | Z (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) | Z (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) | Z (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) | Z (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 | 8 | 1 | 1A | RDY0 or SLRD | Input | N/A | N/A | Multiplexed pin whose function is selected by the following bits:<br>IFCONFIG[1..0].<br>**RDY0** is a GPIF input signal.<br>**SLRD** is the input-only read strobe with programmable polarity (FIFOPINPOLAR.3) for the slave FIFOs connected to FD[7..0] or FD[15..0]. |
| 5 | 4 | 9 | 2 | 1B | RDY1 or SLWR | Input | N/A | N/A | Multiplexed pin whose function is selected by the following bits:<br>IFCONFIG[1..0].<br>**RDY1** is a GPIF input signal.<br>**SLWR** is the input-only write strobe with programmable polarity (FIFOPINPOLAR.2) for the slave FIFOs connected to FD[7..0] or FD[15..0]. |
| 6 | 5 | – | – | – | RDY2 | Input | N/A | N/A | **RDY2** is a GPIF input signal. |

**Table 12. FX2LP Register Summary** (continued)

| Hex | Size | Name | Description | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | Default | Access |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E65D | 1 | USBIRQ[14] | USB Interrupt Requests | 0 | EP0ACK | HSGRANT | 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[14] | Endpoint Interrupt Requests | EP8 | EP6 | EP4 | EP2 | EP1OUT | EP1IN | EP0OUT | EP0IN | 0 | RW |
| E660 | 1 | GPIFIE[13] | GPIF Interrupt Enable | 0 | 0 | 0 | 0 | 0 | 0 | GPIFWF | GPIFDONE | 00000000 | RW |
| E661 | 1 | GPIFIRQ[13] | 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[14] | 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 | INTSET-UP | 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 | reserved | | | | | | | | | | | |
| E677 | 1 | reserved | | | | | | | | | | | |
| E678 | 1 | $I^2CS$ | 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 | $I^2CTL$ | 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[13] | UDMA CRC MSB | CRC15 | CRC14 | CRC13 | CRC12 | CRC11 | CRC10 | CRC9 | CRC8 | 01001010 | RW |
| E67E | 1 | UDMACRCL[13] | 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 | HSM | 0 | 0 | 0 | DISCON | NOSYNSOF | RENUM | SIGRSUME | x0000000 | rrrrbbbb |
| E681 | 1 | SUSPEND | Put chip into suspend | x | x | x | x | x | x | x | x | 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 | I/O | 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 | MICROFRAME | Microframe count, 0-7 | 0 | 0 | 0 | 0 | 0 | MF2 | MF1 | MF0 | 00000xxx | R |
| E687 | 1 | FNADDR | USB Function address | 0 | FA6 | FA5 | FA4 | FA3 | FA2 | FA1 | FA0 | 0xxxxxxx | R |
| E688 | 2 | reserved | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | ENDPOINTS | | | | | | | | | | | |
| E68A | 1 | EP0BCH[13] | Endpoint 0 Byte Count H | (BC15) | (BC14) | (BC13) | (BC12) | (BC11) | (BC10) | (BC9) | (BC8) | xxxxxxxx | RW |
| E68B | 1 | EP0BCL[13] | 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 | 0xxxxxxx | RW |
| E68E | 1 | reserved | | | | | | | | | | | |
| E68F | 1 | EP1INBC | Endpoint 1 IN Byte Count | 0 | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 | 0xxxxxxx | RW |
| E690 | 1 | EP2BCH[13] | Endpoint 2 Byte Count H | 0 | 0 | 0 | 0 | 0 | BC10 | BC9 | BC8 | 00000xxx | RW |
| E691 | 1 | EP2BCL[13] | Endpoint 2 Byte Count L | BC7/SKIP | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 | xxxxxxxx | RW |
| E692 | 2 | reserved | | | | | | | | | | | |
| E694 | 1 | EP4BCH[13] | Endpoint 4 Byte Count H | 0 | 0 | 0 | 0 | 0 | 0 | BC9 | BC8 | 000000xx | RW |
| E695 | 1 | EP4BCL[13] | Endpoint 4 Byte Count L | BC7/SKIP | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 | xxxxxxxx | RW |
| E696 | 2 | reserved | | | | | | | | | | | |
| E698 | 1 | EP6BCH[13] | Endpoint 6 Byte Count H | 0 | 0 | 0 | 0 | 0 | BC10 | BC9 | BC8 | 00000xxx | RW |
| E699 | 1 | EP6BCL[13] | Endpoint 6 Byte Count L | BC7/SKIP | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 | xxxxxxxx | RW |
| E69A | 2 | reserved | | | | | | | | | | | |
| E69C | 1 | EP8BCH[13] | Endpoint 8 Byte Count H | 0 | 0 | 0 | 0 | 0 | 0 | BC9 | BC8 | 000000xx | RW |
| E69D | 1 | EP8BCL[13] | Endpoint 8 Byte Count L | BC7/SKIP | BC6 | BC5 | BC4 | BC3 | BC2 | BC1 | BC0 | xxxxxxxx | RW |
| E69E | 2 | reserved | | | | | | | | | | | |

**Table 12. FX2LP Register Summary** (continued)

| Hex | Size | Name | Description | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | Default | Access |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| xxxx | | I²C Configuration Byte | | 0 | DISCON | 0 | 0 | 0 | 0 | 0 | 400KHZ | xxxxxxxx [16] | n/a |
| | | Special Function Registers (SFRs) | | | | | | | | | | | |
| 80 | 1 | IOA[15] | Port A (bit addressable) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| 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[15] | Data Pointer 1 L | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | 00000000 | RW |
| 85 | 1 | DPH1[15] | Data Pointer 1 H | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | 00000000 | RW |
| 86 | 1 | DPS[15] | 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[15] | Clock Control | x | x | T2M | T1M | T0M | MD2 | MD1 | MD0 | 00000001 | RW |
| 8F | 1 | reserved | | | | | | | | | | | |
| 90 | 1 | IOB[15] | Port B (bit addressable) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| 91 | 1 | EXIF[15] | External Interrupt Flag(s) | IE5 | IE4 | I²CINT | USBNT | 1 | 0 | 0 | 0 | 00001000 | RW |
| 92 | 1 | MPAGE[15] | 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[15] | Autopointer 1 Address H | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | 00000000 | RW |
| 9B | 1 | AUTOPTRL1[15] | Autopointer 1 Address L | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | 00000000 | RW |
| 9C | 1 | reserved | | | | | | | | | | | |
| 9D | 1 | AUTOPTRH2[15] | Autopointer 2 Address H | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | 00000000 | RW |
| 9E | 1 | AUTOPTRL2[15] | Autopointer 2 Address L | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | 00000000 | RW |
| 9F | 1 | reserved | | | | | | | | | | | |
| A0 | 1 | IOC[15] | Port C (bit addressable) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| A1 | 1 | INT2CLR[15] | Interrupt 2 clear | x | x | x | x | x | x | x | x | xxxxxxxx | W |
| A2 | 1 | INT4CLR[15] | 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[15] | Endpoint 2,4,6,8 status flags | EP8F | EP8E | EP6F | EP6E | EP4F | EP4E | EP2F | EP2E | 01011010 | R |
| AB | 1 | EP24FIFOFLGS[15] | Endpoint 2,4 slave FIFO status flags | 0 | EP4PF | EP4EF | EP4FF | 0 | EP2PF | EP2EF | EP2FF | 00100010 | R |
| AC | 1 | EP68FIFOFLGS[15] | Endpoint 6,8 slave FIFO status flags | 0 | EP8PF | EP8EF | EP8FF | 0 | EP6PF | EP6EF | EP6FF | 01100110 | R |
| AD | 2 | reserved | | | | | | | | | | | |
| AF | 1 | AUTOPTRSETUP[15] | Autopointer 1&2 setup | 0 | 0 | 0 | 0 | 0 | APTR2INC | APTR1INC | APTREN | 00000110 | RW |
| B0 | 1 | IOD[15] | Port D (bit addressable) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| B1 | 1 | IOE[15] | Port E (NOT bit addressable) | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| B2 | 1 | OEA[15] | Port A Output Enable | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00000000 | RW |
| B3 | 1 | OEB[15] | Port B Output Enable | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00000000 | RW |
| B4 | 1 | OEC[15] | Port C Output Enable | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00000000 | RW |
| B5 | 1 | OED[15] | Port D Output Enable | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00000000 | RW |
| B6 | 1 | OEE[15] | 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[15] | Endpoint 0&1 Status | 0 | 0 | 0 | 0 | 0 | EP1INBSY | EP1OUTBSY | EP0BSY | 00000000 | R |
| BB | 1 | GPIFTRIG[15, 13] | 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[15] | GPIF Data H (16-bit mode only) | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | xxxxxxxx | RW |

**Notes**
15. SFRs not part of the standard 8051 architecture.
16. If no EEPROM is detected by the SIE then the default is 00000000.

## DC Characteristics

**Table 14. DC Characteristics**

| Parameter | Description | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| VCC | Supply voltage | – | 3.00 | 3.3 | 3.60 | V |
| VCC Ramp Up | 0 to 3.3 V | – | 200 | – | – | µs |
| $V_{IH}$ | 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 | – | –0.5 | – | 0.8 | V |
| $I_I$ | Input leakage current | $0< V_{IN} < V_{CC}$ | – | – | ±10 | µA |
| $V_{OH}$ | Output voltage HIGH | $I_{OUT}$ = 4 mA | 2.4 | – | – | V |
| $V_{OL}$ | Output LOW voltage | $I_{OUT}$ = –4 mA | – | – | 0.4 | V |
| $I_{OH}$ | Output current HIGH | – | – | – | 4 | mA |
| $I_{OL}$ | Output current LOW | – | – | – | 4 | mA |
| $C_{IN}$ | Input pin capacitance | Except D+/D– | – | – | 10 | pF |
| | | D+/D– | – | – | 15 | pF |
| $I_{SUSP}$ | Suspend current CY7C68014/CY7C68016 | Connected | – | 300 | 380[18] | µA |
| | | Disconnected | – | 100 | 150[18] | µA |
| | Suspend current CY7C68013/CY7C68015 | Connected | – | 0.5 | 1.2[18] | mA |
| | | Disconnected | – | 0.3 | 1.0[18] | mA |
| $I_{CC}$ | Supply current | 8051 running, connected to USB HS | – | 50 | 85 | mA |
| | | 8051 running, connected to USB FS | – | 35 | 65 | mA |
| $T_{RESET}$ | Reset time after valid power | $V_{CC}$ min = 3.0 V | 5.0 | – | – | ms |
| | Pin reset after powered on | | 200 | – | – | µs |

### USB Transceiver

USB 2.0 compliant in Full Speed and Hi-Speed modes.

**Note**
18. Measured at Max $V_{CC}$, 25 °C.

## 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 or 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 is asserted two clock cycles after PORTC is updated and is active for two clock cycles after that, as shown in Figure 16.

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 two clock cycles after three clock cycles from the point when the 8051 has performed a read function on PORTC.

The RD# signal prompts 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 easily meets the setup time to the next read.

The purpose of this pulsing of RD# is to allow the external peripheral to know that the 8051 is done reading PORTC and the data was latched into PORTC three CLKOUT cycles before asserting the RD# signal. After the RD# is pulsed, the external logic can update the data on PORTC.

Following is the timing diagram of the read and write strobing function on accessing PORTC. Refer to Data Memory Read[21] and Data Memory Write[23] for details on propagation delay of RD# and WR# signals.

**Figure 16. WR# Strobe Function when PORTC is Accessed by 8051**

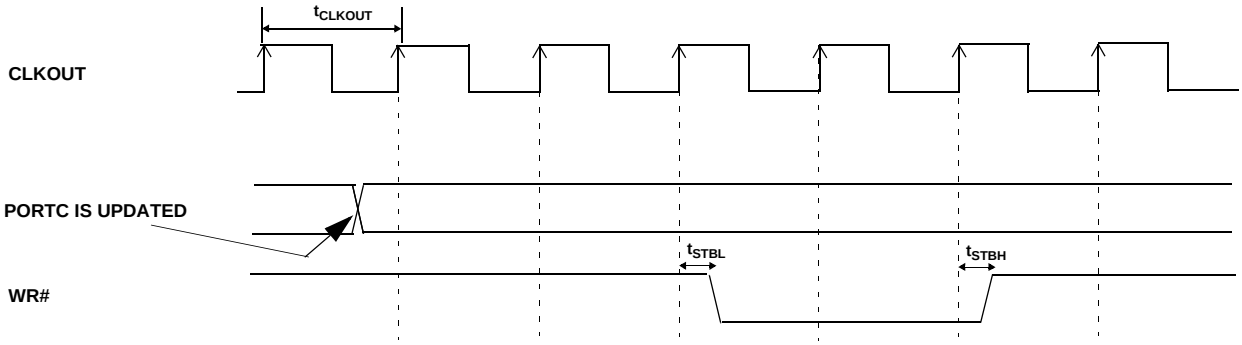
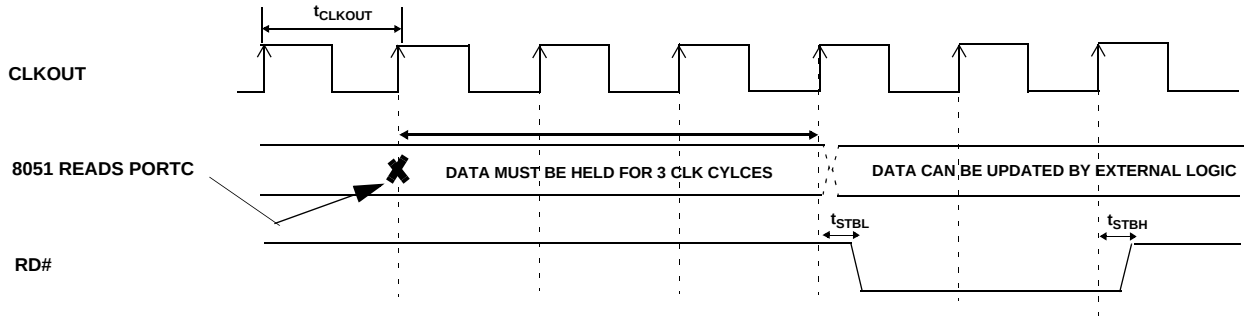



**Figure 17. RD# Strobe Function when PORTC is Accessed by 8051**



## Slave FIFO Synchronous Read

**Figure 19. Slave FIFO Synchronous Read Timing Diagram**[24]

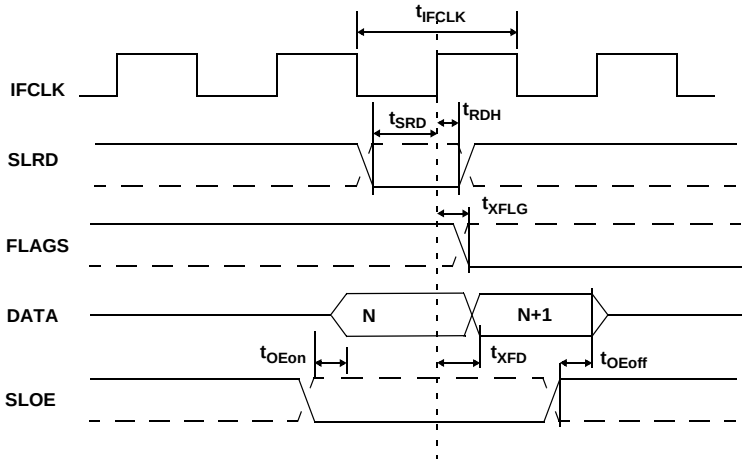



**Table 20. Slave FIFO Synchronous Read Parameters with Internally Sourced IFCLK**[25]

| Parameter | Description | Min | Max | Typ | | Unit |
|---|---|---|---|---|---|---|
| | | | | Min | Max | |
| $t_{IFCLK}$ | 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 |
| $t_{IFCLKR}$ | IFCLK rise time | – | – | – | 900 | ps |
| $t_{IFCLKF}$ | IFCLK fall time | – | – | – | 900 | ps |
| $t_{IFCLKOD}$ | IFCLK output duty cycle | – | – | 49 | 51 | % |
| $t_{IFCLKJ}$ | IFCLK jitter peak to peak | – | – | – | 300 | ps |

### Slave FIFO Synchronous Write

**Figure 21. Slave FIFO Synchronous Write Timing Diagram[24]**

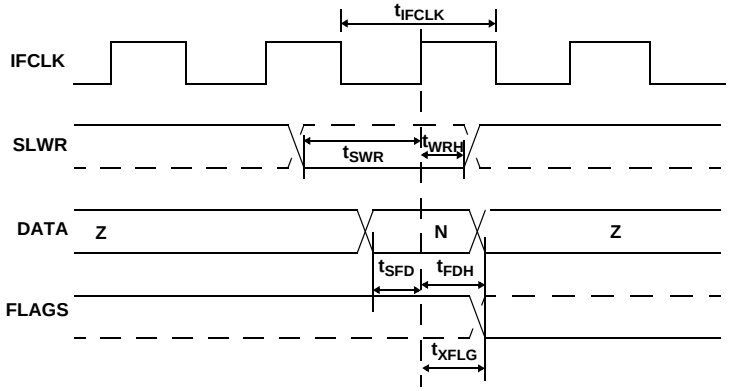



**Table 23. Slave FIFO Synchronous Write Parameters with Internally Sourced IFCLK[25]**

| Parameter | Description | Min | Max | Unit |
|---|---|---|---|---|
| $t_{IFCLK}$ | IFCLK period | 20.83 | – | ns |
| $t_{SWR}$ | SLWR to clock setup time | 10.4 | – | ns |
| $t_{WRH}$ | Clock to SLWR hold time | 0 | – | ns |
| $t_{SFD}$ | FIFO data to clock setup time | 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 24. Slave FIFO Synchronous Write Parameters with Externally Sourced IFCLK[25]**

| 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 |

### Slave FIFO Asynchronous Write

**Figure 22. Slave FIFO Asynchronous Write Timing Diagram[24]**

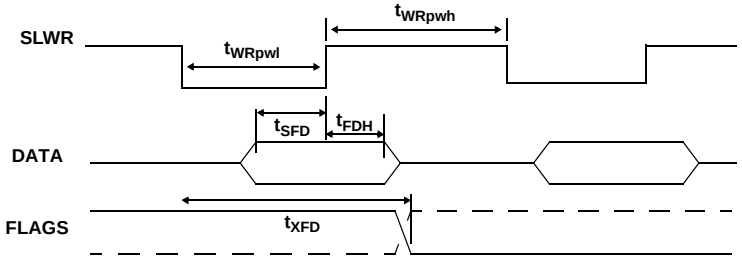


**Table 25. Slave FIFO Asynchronous Write Parameters with Internally Sourced IFCLK[27]**

| 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 |

## Slave FIFO Synchronous Packet End Strobe

**Figure 23. Slave FIFO Synchronous Packet End Strobe Timing Diagram[24]**

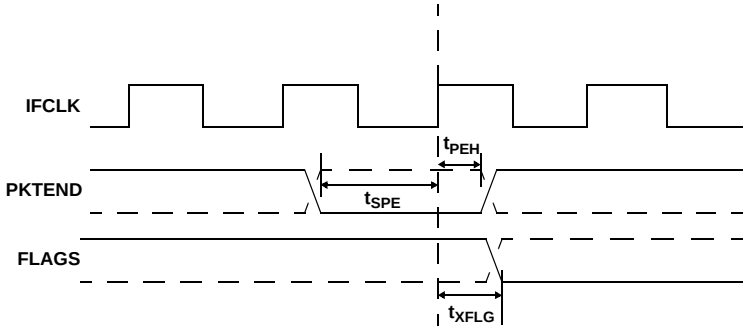



**Table 26. Slave FIFO Synchronous Packet End Strobe Parameters with Internally Sourced IFCLK[25]**

| 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 27. Slave FIFO Synchronous Packet End Strobe Parameters with Externally Sourced IFCLK[25]**

| 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 |

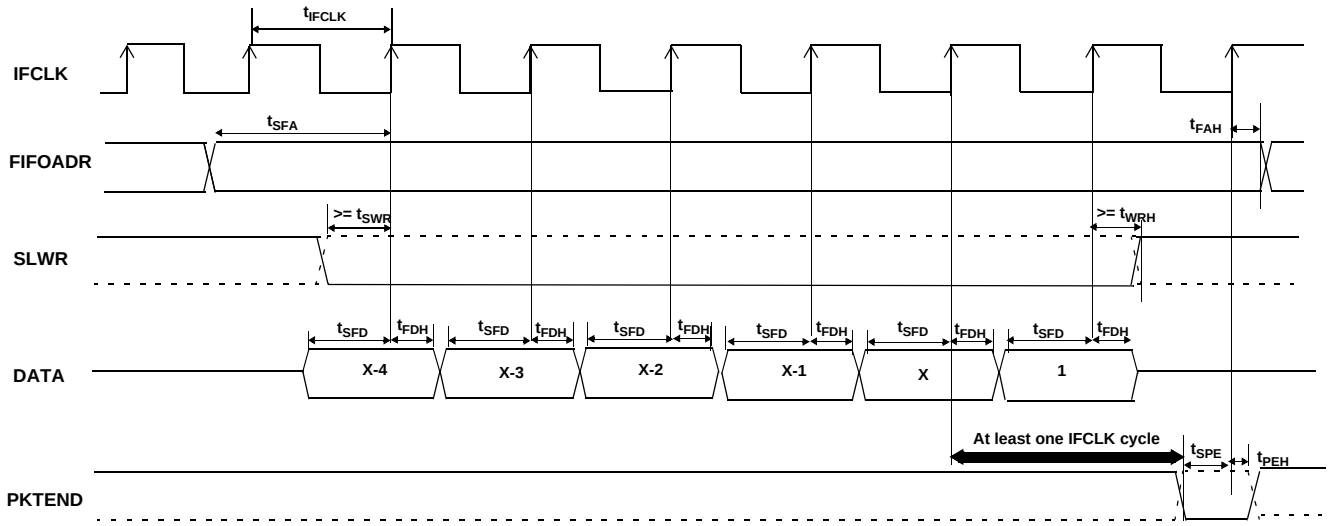
There is no specific timing requirement that should be met for asserting the PKTEND pin to asserting SLWR. PKTEND can be asserted with the last data value clocked into the FIFOs or thereafter. The setup time $t_{SPE}$ and the hold time $t_{PEH}$ must be met.

Although there are no specific timing requirements for PKTEND assertion, there is a specific corner-case condition that needs attention while using the PKTEND pin to commit a one byte or word packet. There is an additional timing requirement that needs to be met when the FIFO is configured to operate in auto mode and it is required 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 or word packet committed manually using the PKTEND pin. In this scenario, the user must ensure to assert PKTEND, at least one clock cycle after the rising edge that caused the last byte or word to be clocked into the previous auto committed packet. Figure 24 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 24 shows a scenario where two packets are 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 the assertion of PKTEND and clocking of the last byte of the previous packet (causing the packet to be committed automatically). Failing to adhere to this timing results in the FX2 failing to send the one byte or word short packet.

**Figure 24. Slave FIFO Synchronous Write Sequence and Timing Diagram[24]**




## Slave FIFO Asynchronous Packet End Strobe

**Figure 25. Slave FIFO Asynchronous Packet End Strobe Timing Diagram[24]**

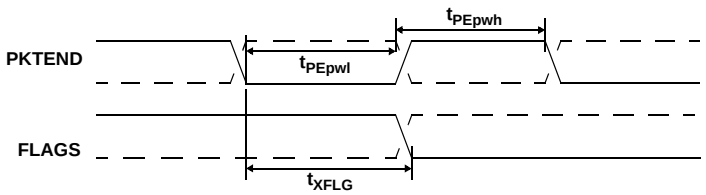



**Table 28. Slave FIFO Asynchronous Packet End Strobe Parameters[27]**

| 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 |

## Slave FIFO Output Enable

**Figure 26. Slave FIFO Output Enable Timing Diagram[24]**

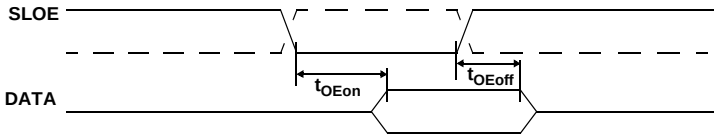



**Table 29. Slave FIFO Output Enable Parameters**

| 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 |

*Single and Burst Synchronous Write*

**Figure 32. Slave FIFO Synchronous Write Sequence and Timing Diagram**[24]

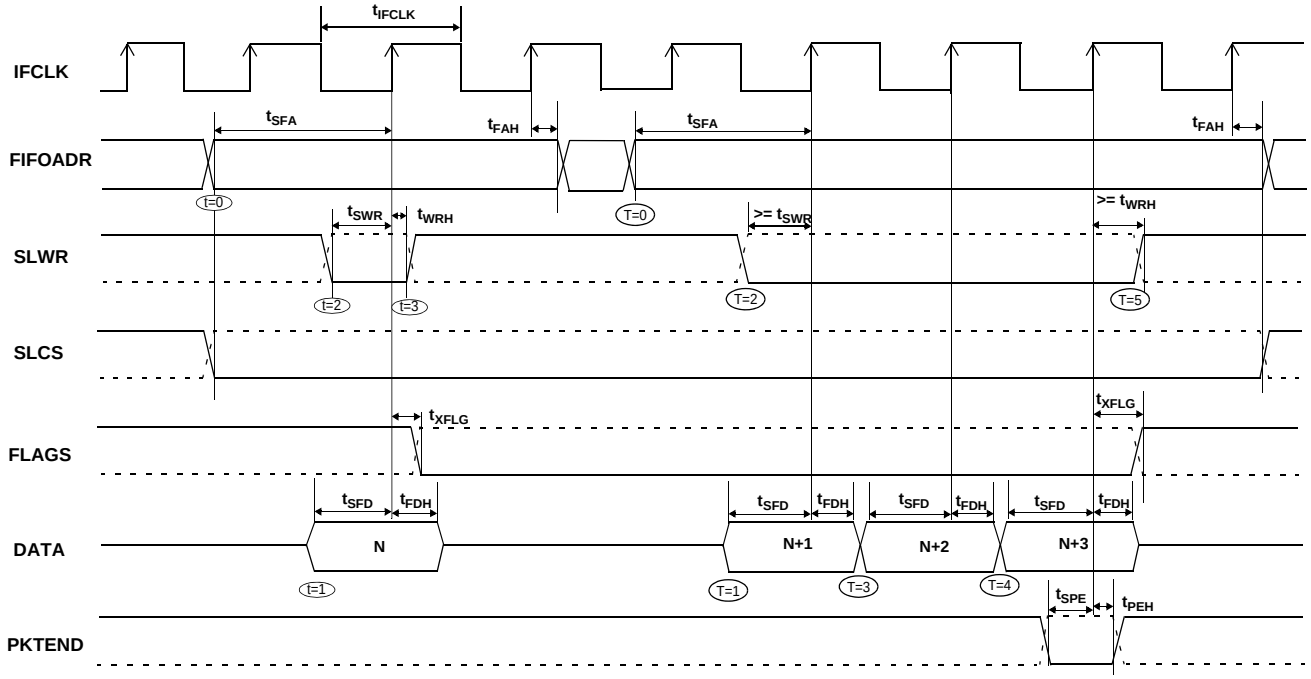



Figure 32 shows the timing relationship of the SLAVE FIFO signals during a synchronous write using IFCLK as the synchronizing clock. The diagram illustrates a single write followed by burst write of three bytes and committing all four bytes as a short packet using the PKTEND pin.

- At t = 0 the FIFO address is stable and the signal SLCS is asserted. (SLCS may be tied LOW in some applications) Note that $t_{SFA}$ 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 the SLCS signal is used, it must be asserted with SLWR or before SLWR is asserted (The SLCS and SLWR signals must both be asserted to start a valid write condition).
- While the SLWR is asserted, data is written to the FIFO and on the rising edge of the IFCLK, the FIFO pointer is incremented. The FIFO flag is also updated after a delay of $t_{XFLG}$ from the rising edge of the clock.

The same sequence of events 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, after 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 32, after the four bytes are written to the FIFO, SLWR is deasserted. The short 4 byte packet can be committed to the host by asserting the PKTEND signal.

There is no specific timing requirement that should 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 requirement is that the setup time $t_{SPE}$ and the hold time $t_{PEH}$ must be met. In the scenario of Figure 32, the number of data values committed includes the last value written to the FIFO. In this example, both the data value and the PKTEND signal are clocked on the same rising edge of IFCLK. PKTEND can also be asserted in subsequent clock cycles. The FIFOADDR lines should be held constant during the PKTEND assertion.

Although there are no specific timing requirement for the PKTEND assertion, there is a specific corner-case condition that needs attention while using the PKTEND to commit a one byte/word packet. Additional timing requirements exist when the FIFO is configured to operate in auto mode and 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 the AUTOINLEN register) committed automatically followed by a short one byte or word packet committed manually using the PKTEND pin.

In this case, the external master must ensure to assert the PKTEND pin at least one clock cycle after the rising edge that caused the last byte or word that needs to be clocked into the previous auto committed packet (the packet with the number of bytes equal to what is set in the AUTOINLEN register). Refer to Figure 24 on page 50 for further details on this timing.

*Sequence Diagram of a Single and Burst Asynchronous Write*

**Figure 35. Slave FIFO Asynchronous Write Sequence and Timing Diagram**[24]

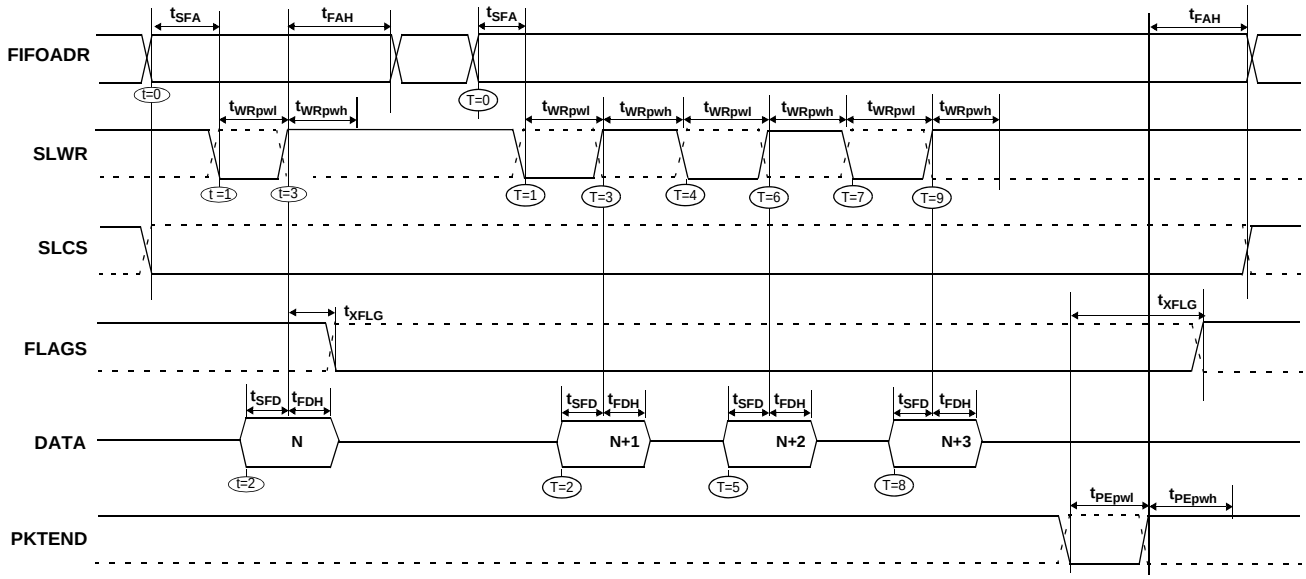



Figure 35 shows the timing relationship of the SLAVE FIFO write in an asynchronous mode. The diagram shows a single write followed by a burst write of 3 bytes and committing the 4byte short packet using PKTEND.

- At t = 0 the FIFO address is applied, ensuring 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 asserted with SLWR or before SLWR is asserted.
- At t = 2, data must be present on the bus $t_{SFD}$ before the deasserting edge of SLWR.
- At t = 3, deasserting SLWR causes the data to be written from the data bus to the FIFO and then increments the FIFO pointer. The FIFO flag is also updated after $t_{XFLG}$ from the deasserting edge of SLWR.

The same sequence of events is shown for a burst write and is indicated by the timing marks of T = 0 through 5.

**Note** In the burst write mode, after 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 35, after the four bytes are written to the FIFO and SLWR is deasserted, the short 4-byte packet can be committed to the host using 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 have to held constant during the PKTEND assertion.

## Package Diagrams

The FX2LP is available in five packages:

- 56-pin SSOP
- 56-pin QFN
- 100-pin TQFP
- 128-pin TQFP
- 56-ball VFBGA

**Figure 36. 56-Pin Shrunk Small Outline Package O56 (51-85062)**

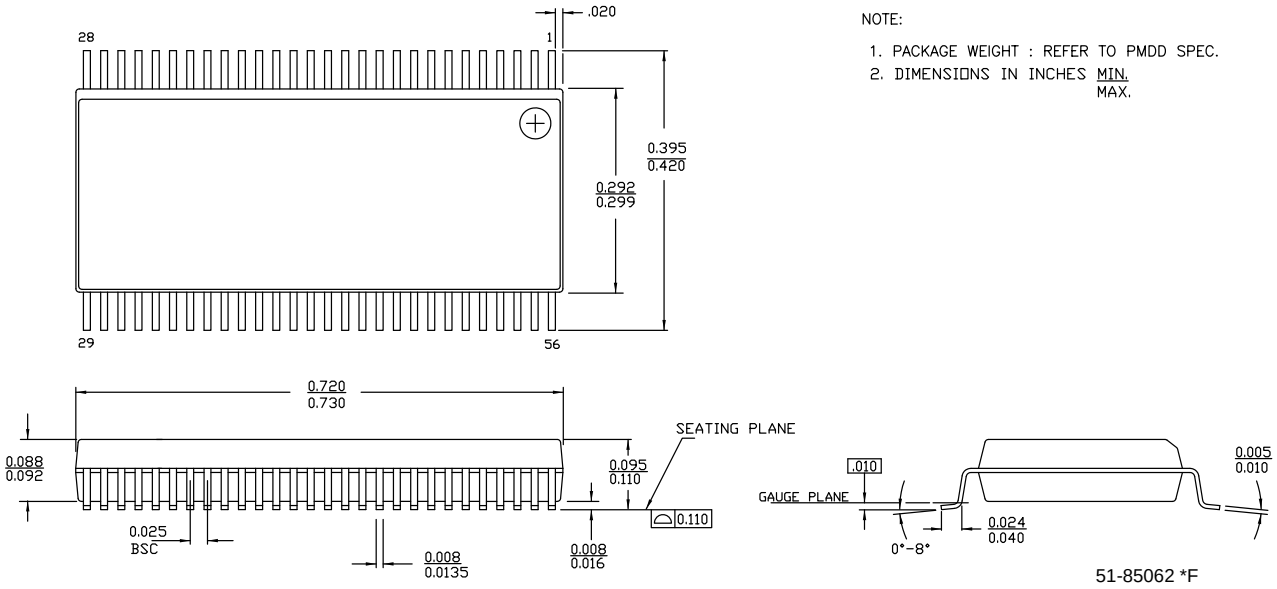
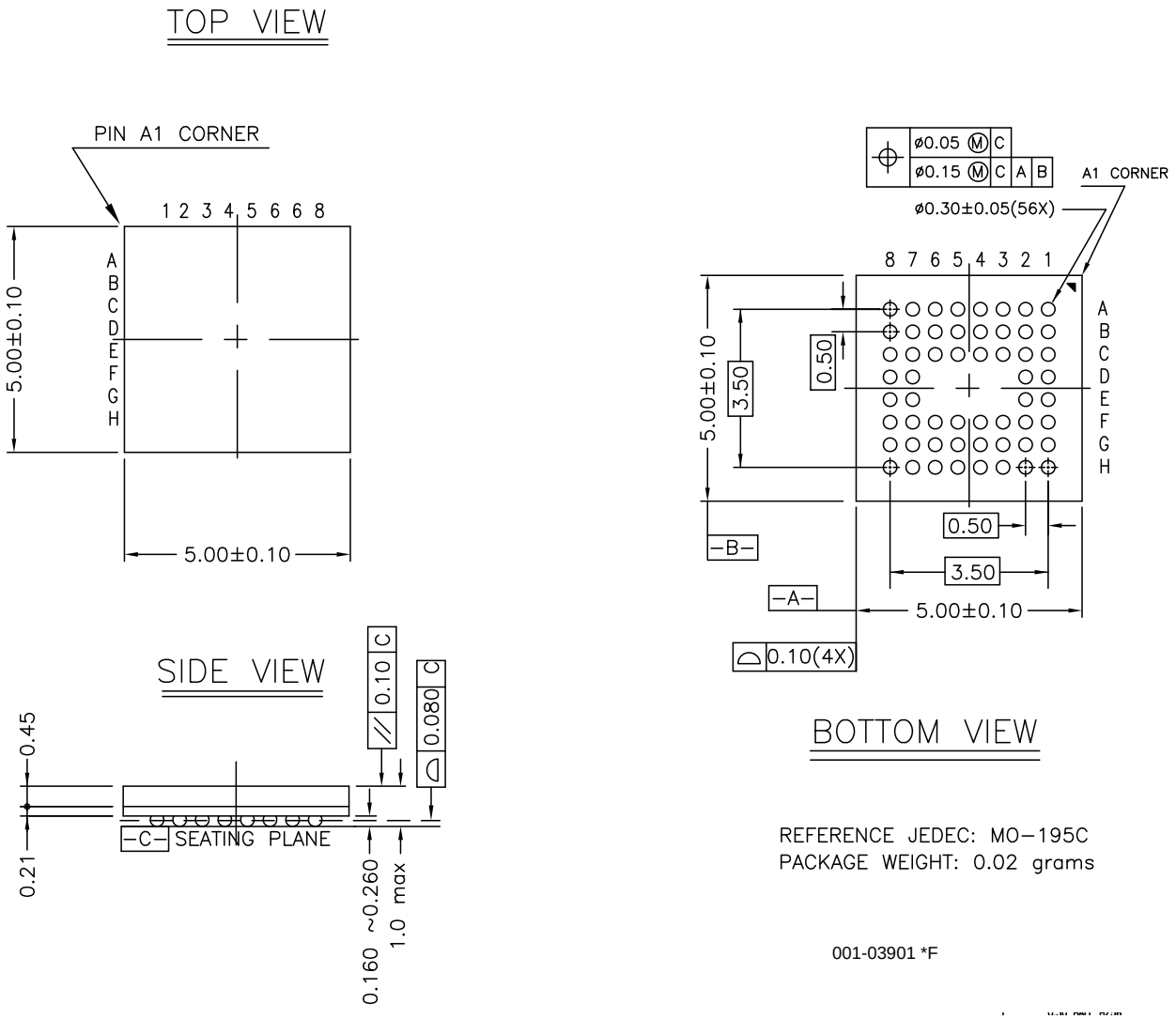


**Figure 40. 56-Pin VFBGA (5 × 5 × 1.0 mm) 0.50 Pitch, 0.30 Ball BZ56 (001-03901)**



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

Electrical contact of the part to the PCB is made by soldering the leads on the bottom surface of the package to the PCB. Therefore, special attention is required to the heat transfer area below the package to provide a good thermal bond to the circuit board. Design a copper (Cu) fill in the PCB as a thermal pad under the package. Heat is transferred from the FX2LP 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 × 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, refer to application notes for Surface Mount Assembly of Amkor's MicroLeadFrame (MLF) Packages. You can find this on Amkor's website http://www.amkor.com.

This application note provides detailed information about board mounting guidelines, soldering flow, rework process, etc.

Figure 41 shows a cross-sectional area underneath the package. The cross section is of only one via. The solder paste template should be designed to allow at least 50% solder coverage. The thickness of the solder paste template should be 5 mil. Use the No Clean type 3 solder paste for mounting the part. Nitrogen purge is recommended during reflow.

Figure 42 is a plot of the solder mask pattern and Figure 43 displays an X-Ray image of the assembly (darker areas indicate solder).

**Figure 41. Cross-section of the Area Underneath the QFN Package**

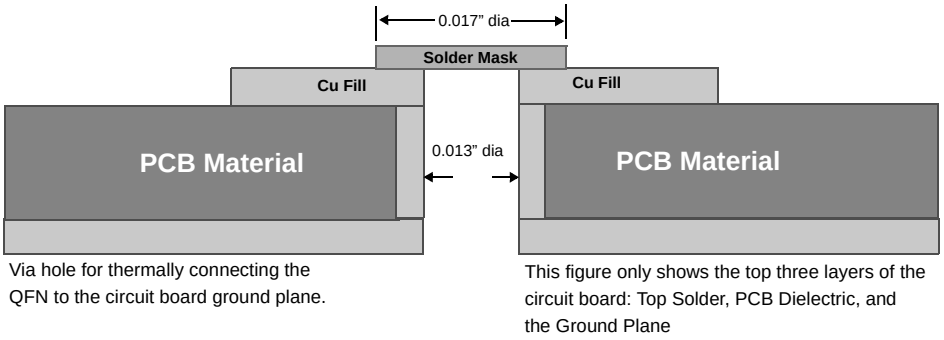



**Figure 42. Plot of the Solder Mask (White Area)**

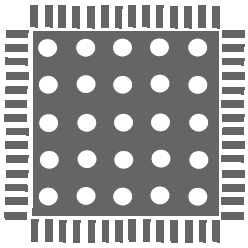


**Figure 43. X-ray Image of the Assembly**