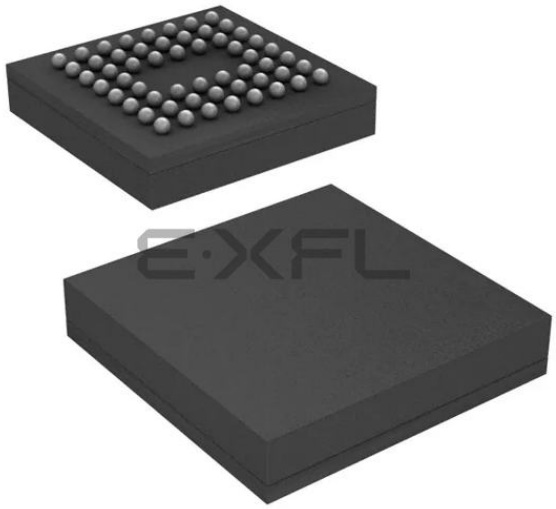


Welcome to E-XFL.COM

## 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 | Active |
| 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-VFBGA |
| Supplier Device Package | 56-VFBGA (5x5) |
| Purchase URL | https://www.e-xfl.com/product-detail/infineon-technologies/cy7c68013a-56baxc |

## More Information

Cypress provides a wealth of data at www.cypress.com to help you to select the right device for your design, and to help you to quickly and effectively integrate the device into your design. For a comprehensive list of resources, see the application note AN65209 - Getting Started with FX2LP.

- Overview: USB Portfolio, USB Roadmap
- USB 2.0 Product Selectors: FX2LP, AT2LP, NX2LP-Flex, SX2
- Application notes: Cypress offers a large number of USB application notes covering a broad range of topics, from basic to advanced level. Recommended application notes for getting started with FX2LP are:
  - AN65209 - Getting Started with FX2LP
  - AN15456 - Guide to Successful EZ-USB® FX2LP™ and EZ-USB FX1™ Hardware Design and Debug
  - AN50963 - EZ-USB® FX1™/FX2LP™ Boot Options
  - AN66806 - EZ-USB® FX2LP™ GPIF Design Guide
  - AN61345 - Implementing an FX2LP™- FPGA Interface
  - AN57322 - Interfacing SRAM with FX2LP over GPIF
  - AN4053 - Streaming Data through Isochronous/Bulk Endpoints on EZ-USB® FX2 and EZUSB FX2LP
  - AN63787 - EZ-USB® FX2LP™ GPIF and Slave FIFO Configuration Examples using 8-bit Asynchronous Interface

For complete list of Application notes, click here.

- Code Examples:
  - USB Hi-Speed
- Technical Reference Manual (TRM):
  - EZ-USB FX2LP Technical Reference Manual
- Reference Designs:
  - CY4661 - External USB Hard Disk Drives (HDD) with Fingerprint Authentication Security
  - FX2LP DMB-T/H TV Dongle reference design
- Models: IBIS

### EZ-USB FX2LP Development Kit

The CY3684 EZ-USB FX2LP Development Kit is a complete development resource for FX2LP. It provides a platform to develop and test custom projects using FX2LP. The development kit contains collateral materials for the firmware, hardware, and software aspects of a design using FX2LP.

### GPIF™ Designer

FX2LP™ General Programmable Interface (GPIF) provides an independent hardware unit, which creates the data and control signals required by an external interface. FX2LP GPIF Designer allows users to create and modify GPIF waveform descriptors for EZ-USB FX2/ FX2LP family of chips using a graphical user interface. Extensive discussion of general GPIF discussion and programming using GPIF Designer is included in *FX2LP Technical Reference Manual* and *GPIF Designer User Guide*, distributed with GPIF Designer. *AN66806 - Getting Started with EZ-USB® FX2LP™ GPIF* can be a good starting point.

**Table 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 [3] |
| 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 INTSET-UP register), the FX 2LP 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 directs the jump to the correct address out of the 14 addresses within the page. When the ISR occurs, the FX2LP pushes the program counter to its stack then jumps to address 0x0053, where it expects to find a “jump” instruction to the interrupt service routine (ISR).

**Note**
3. Errata: In Slave FIFO Asynchronous Word Wide mode, if a single word data is transferred from the USB host to EP2, configured as OUT Endpoint (EP) in the first transaction, then the Empty flag behaves incorrectly. This does not happen if the data size is more than one word in the first transaction. For more information, see the “Errata” on page 65.

## Reset and Wakeup

*Reset Pin*

The input pin, RESET#, resets the FX2LP when asserted. This pin has hysteresis and is active LOW. When a crystal is used with the CY7C680xxA, the reset period must enable stabilization of the crystal and the PLL. This reset period must be approximately 5 ms after VCC reaches 3.0 V. If the crystal input pin is driven by a clock signal, the internal PLL stabilizes in 200 µs after VCC has reached 3.0 V[4].

Figure 2 on page 9 shows a power-on reset condition and a reset applied during operation. A power-on reset is defined as the time reset that is asserted while power is being applied to the circuit. A powered reset is when the FX2LP is powered on and operating and the RESET# pin is asserted.

Cypress provides an application note which describes and recommends power-on reset implementation. For more information about reset implementation for the FX2 family of products, visit http://www.cypress.com.

**Figure 2. Reset Timing Plots**

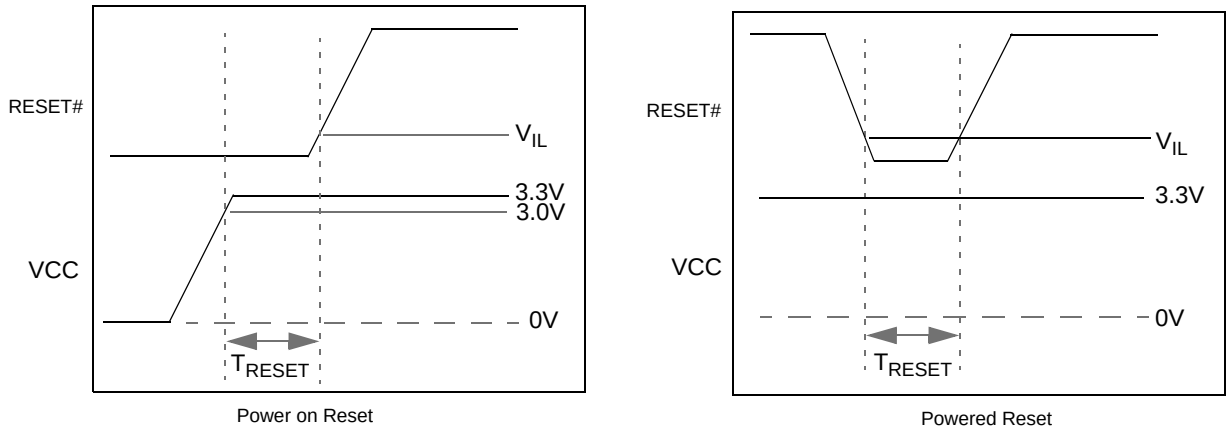



**Table 5. Reset Timing Values**

| Condition | $T_{RESET}$ |
|---|---|
| Power-on reset with crystal | 5 ms |
| Power-on reset with external clock | 200 µs + clock stability time |
| Powered reset | 200 µs |

*Wakeup Pins*

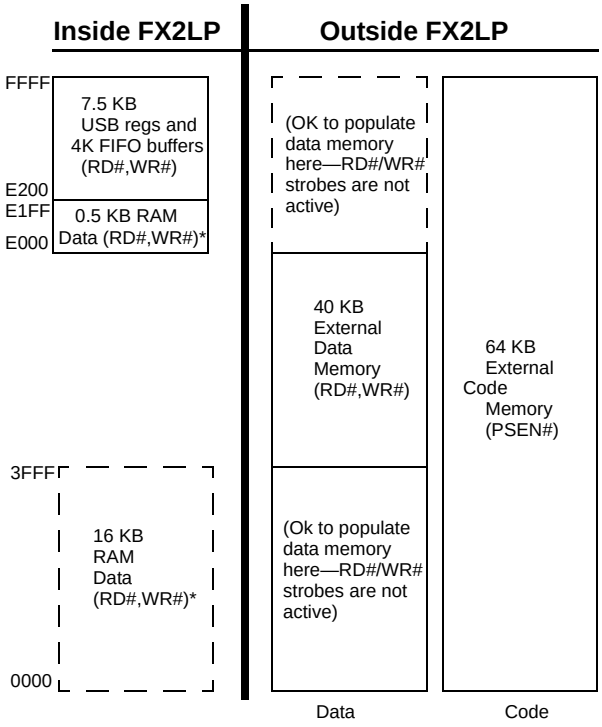
The 8051 puts itself and the rest of the chip into a power-down mode by setting PCON.0 = 1. This stops the oscillator and PLL. When WAKEUP is asserted by external logic, the oscillator restarts after the PLL stabilizes, and the 8051 receives a wakeup interrupt. This applies irrespective of whether FX2LP is connected to the USB.

The FX2LP exits the power-down (USB suspend) state by 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 FX2LP 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 enables a simple external R-C network to be used as a periodic wakeup source. WAKEUP is by default active LOW.

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




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

## Register Addresses

| | |
|---|---|
| FFFF<br>F000 | 4 KB EP2-EP8 buffers (8 x 512) |
| EFFF<br>E800 | 2 KB RESERVED |
| E7FF<br>E7C0 | 64 BEP1IN |
| 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 8051 xdata RAM |

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.

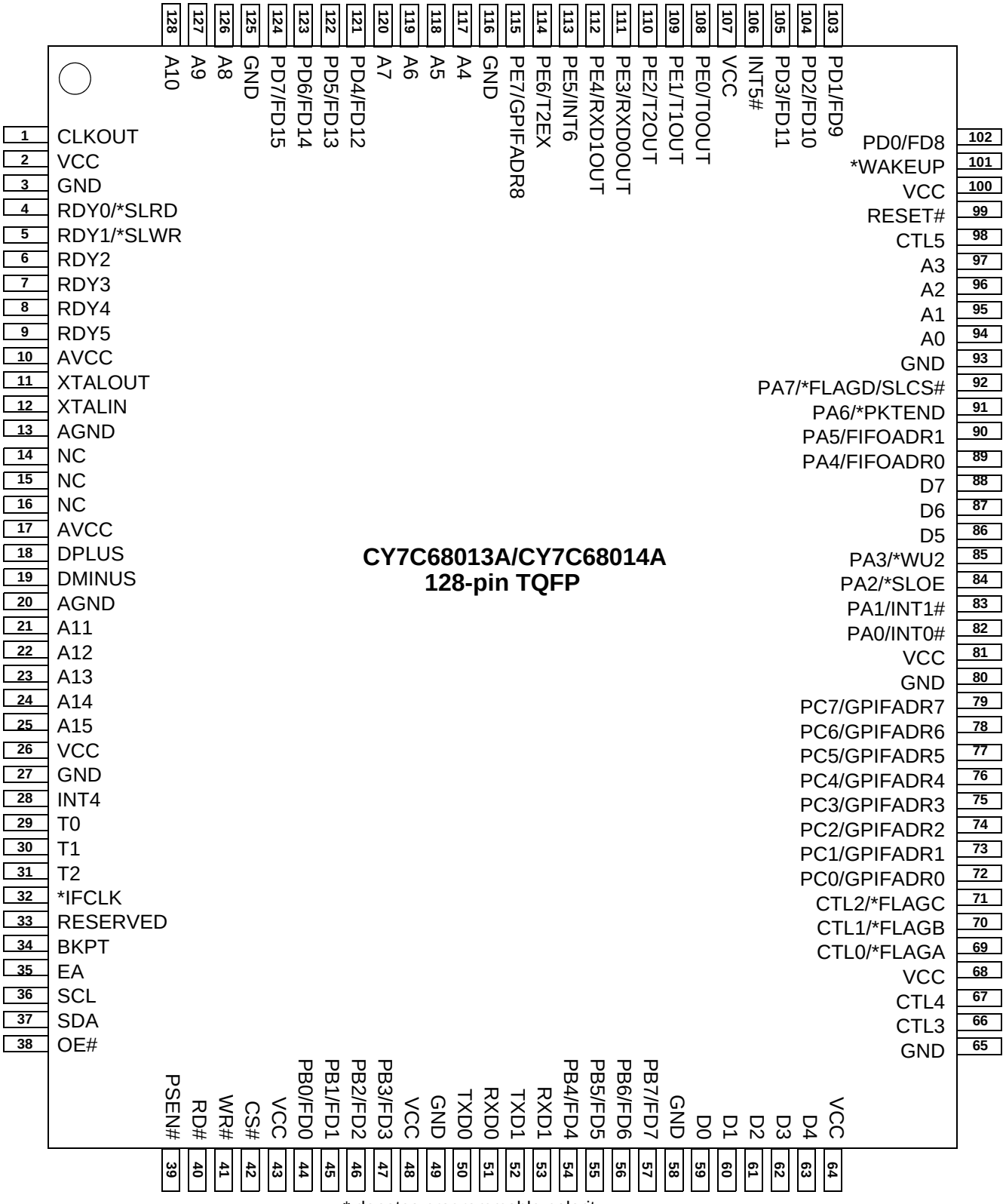
Write any value to ECCRESET then pass data across the GPIF or Slave FIFO interface. The ECC for the first 512 bytes of data is calculated and stored in ECC1; ECC2 is unused. After the ECC is calculated, the values in ECC1 do not change even if more data is subsequently passed across the interface, till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

**Figure 7. CY7C68013A/CY7C68014A 128-Pin TQFP Pin Assignment**




* denotes programmable polarity

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

| 128 TQFP | 100 TQFP | 56 SSOP | 56 QFN | 56 VFBGA | Name | Type | Default | Reset[12] | Description |
|---|---|---|---|---|---|---|---|---|---|
| 84 | 69 | 42 | 35 | 8F | PA2 or SLOE | I/O/Z | I (PA2) | Z (PA2) | Multiplexed pin whose function is selected by two bits: IFCONFIG[1:0]. **PA2** is a bidirectional I/O port pin. **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 | 43 | 36 | 7F | PA3 or WU2 | I/O/Z | I (PA3) | Z (PA3) | Multiplexed pin whose function is selected by: WAKEUP.7 and OEA.3 **PA3** is a bidirectional I/O port pin. **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 enable it to exit the suspend mode. Asserting this pin inhibits the chip from suspending if WU2EN = 1. |
| 89 | 71 | 44 | 37 | 6F | PA4 or FIFOADR0 | I/O/Z | I (PA4) | Z (PA4) | Multiplexed pin whose function is selected by: IFCONFIG[1..0]. **PA4** is a bidirectional I/O port pin. **FIFOADR0** is an input-only address select for the slave FIFOs connected to FD[7..0] or FD[15..0]. |
| 90 | 72 | 45 | 38 | 8C | PA5 or FIFOADR1 | I/O/Z | I (PA5) | Z (PA5) | Multiplexed pin whose function is selected by: IFCONFIG[1..0]. **PA5** is a bidirectional I/O port pin. **FIFOADR1** is an input-only address select for the slave FIFOs connected to FD[7..0] or FD[15..0]. |
| 91 | 73 | 46 | 39 | 7C | PA6 or PKTEND | I/O/Z | I (PA6) | Z (PA6) | Multiplexed pin whose function is selected by the IFCONFIG[1:0] bits. **PA6** is a bidirectional I/O port pin. **PKTEND** is an input used to commit the FIFO packet data to the endpoint and whose polarity is programmable via FIFOPINPOLAR.5. |
| 92 | 74 | 47 | 40 | 6C | PA7 or FLAGD or SLCS# | I/O/Z | I (PA7) | Z (PA7) | Multiplexed pin whose function is selected by the IFCONFIG[1:0] and PORTACFG.7 bits. **PA7** is a bidirectional I/O port pin. **FLAGD** is a programmable slave-FIFO output status flag signal. **SLCS#** gates all other slave FIFO enable/strobes |
| **Port B** | | | | | | | | | |
| 44 | 34 | 25 | 18 | 3H | PB0 or FD[0] | I/O/Z | I (PB0) | Z (PB0) | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0]. **PB0** is a bidirectional I/O port pin. **FD[0]** is the bidirectional FIFO/GPIF data bus. |
| 45 | 35 | 26 | 19 | 4F | PB1 or FD[1] | I/O/Z | I (PB1) | Z (PB1) | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0]. **PB1** is a bidirectional I/O port pin. **FD[1]** is the bidirectional FIFO/GPIF data bus. |
| 46 | 36 | 27 | 20 | 4H | PB2 or FD[2] | I/O/Z | I (PB2) | Z (PB2) | Multiplexed pin whose function is selected by the following bits: IFCONFIG[1..0]. **PB2** is a bidirectional I/O port pin. **FD[2]** is the bidirectional FIFO/GPIF data bus. |

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

| 128 TQFP | 100 TQFP | 56 SSOP | 56 QFN | 56 VFBGA | Name | Type | Default | Reset[12] | Description |
|---|---|---|---|---|---|---|---|---|---|
| 47 | 37 | 28 | 21 | 4G | PB3 or FD[3] | I/O/Z | I (PB3) | Z (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 | 29 | 22 | 5H | PB4 or FD[4] | I/O/Z | I (PB4) | Z (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 | 30 | 23 | 5G | PB5 or FD[5] | I/O/Z | I (PB5) | Z (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 | 31 | 24 | 5F | PB6 or FD[6] | I/O/Z | I (PB6) | Z (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 | 32 | 25 | 6H | PB7 or FD[7] | I/O/Z | I (PB7) | Z (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) | Z (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) | Z (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) | Z (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) | Z (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) | Z (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. |
| 77 | 62 | – | – | – | PC5 or GPIFADR5 | I/O/Z | I (PC5) | Z (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) | Z (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) | Z (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. |

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

| 128 TQFP | 100 TQFP | 56 SSOP | 56 QFN | 56 VFBGA | Name | Type | Default | Reset[12] | Description |
|---|---|---|---|---|---|---|---|---|---|
| 100 | 78 | 50 | 43 | 5B | VCC | Power | N/A | N/A | **VCC**. Connect to the 3.3-V power source. |
| 107 | 85 | – | – | – | VCC | Power | N/A | N/A | **VCC**. Connect to the 3.3-V power source. |
| | | | | | | | | | |
| 3 | 2 | 7 | 56 | 4B | GND | Ground | N/A | N/A | **Ground** |
| 27 | 21 | 19 | 12 | 1H | GND | Ground | N/A | N/A | **Ground** |
| 49 | 39 | – | – | – | GND | Ground | N/A | N/A | **Ground** |
| 58 | 48 | 33 | 26 | 7D | GND | Ground | N/A | N/A | **Ground** |
| 65 | 50 | 35 | 28 | 8D | GND | Ground | N/A | N/A | **Ground** |
| 80 | 65 | – | – | – | GND | Ground | N/A | N/A | **Ground** |
| 93 | 75 | 48 | 41 | 4C | GND | Ground | N/A | N/A | **Ground** |
| 116 | 94 | – | – | – | GND | Ground | N/A | N/A | **Ground** |
| 125 | 99 | 4 | 53 | 4A | GND | Ground | N/A | N/A | **Ground** |
| | | | | | | | | | |
| 14 | 13 | – | – | – | NC | N/A | N/A | N/A | **No Connect**. This pin must be left open. |
| 15 | 14 | – | – | – | NC | N/A | N/A | N/A | **No Connect**. This pin must be left open. |
| 16 | 15 | – | – | – | NC | N/A | N/A | N/A | **No Connect**. This pin must be left open. |

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

| Hex | Size | Name | Description | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | Default | Access |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | SET-UPDAT | 8 bytes of setup data | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | R |
| | | | SET-UPDAT[0] = bmRequestType | | | | | | | | | | |
| | | | SET-UPDAT[1] = bmRequest | | | | | | | | | | |
| | | | SET-UPDAT[2:3] = wValue | | | | | | | | | | |
| | | | SET-UPDAT[4:5] = wIndex | | | | | | | | | | |
| | | | SET-UPDAT[6:7] = wLength | | | | | | | | | | |
| | | GPIF | | | | | | | | | | | |
| E6C0 | 1 | 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[13] | GPIF Address H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | GPIFA8 | 00000000 | RW |
| E6C5 | 1 | GPIFADRL[13] | 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 | HOPERIOD0 | HOSTATE | HOCTL2 | HOCTL1 | HOCTL0 | 00010010 | RW |
| E6CB | 1 | FLOWSTB | Flowstate Strobe Configuration | SLAVE | RDYASYNC | CTLTOGL | SUSTAIN | 0 | MSTB2 | MSTB1 | MSTB0 | 00100000 | RW |
| E6CC | 1 | FLOWSTBEDGE | Flowstate Rising/Falling Edge Configuration | 0 | 0 | 0 | 0 | 0 | 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[13] | GPIF Transaction Count Byte 3 | TC31 | TC30 | TC29 | TC28 | TC27 | TC26 | TC25 | TC24 | 00000000 | RW |

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

| Hex | Size | Name | Description | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 | Default | Access |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BE | 1 | GPIFSGLDATLX[15] | GPIF Data L w/ Trigger | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | RW |
| BF | 1 | GPIFSGLDATL-NOX[15] | GPIF Data L w/ No Trigger | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | xxxxxxxx | R |
| C0 | 1 | SCON1[15] | 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[15] | Serial Port 1 Data Buffer | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | 00000000 | RW |
| C2 | 6 | reserved | | | | | | | | | | | |
| C8 | 1 | T2CON | Timer/Counter 2 Control (bit addressable) | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | CT2 | CPRL2 | 00000000 | RW |
| 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[15] | 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[15] | 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[15] | 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

## Data Memory Read[21]

**Figure 13. Data Memory Read Timing Diagram**

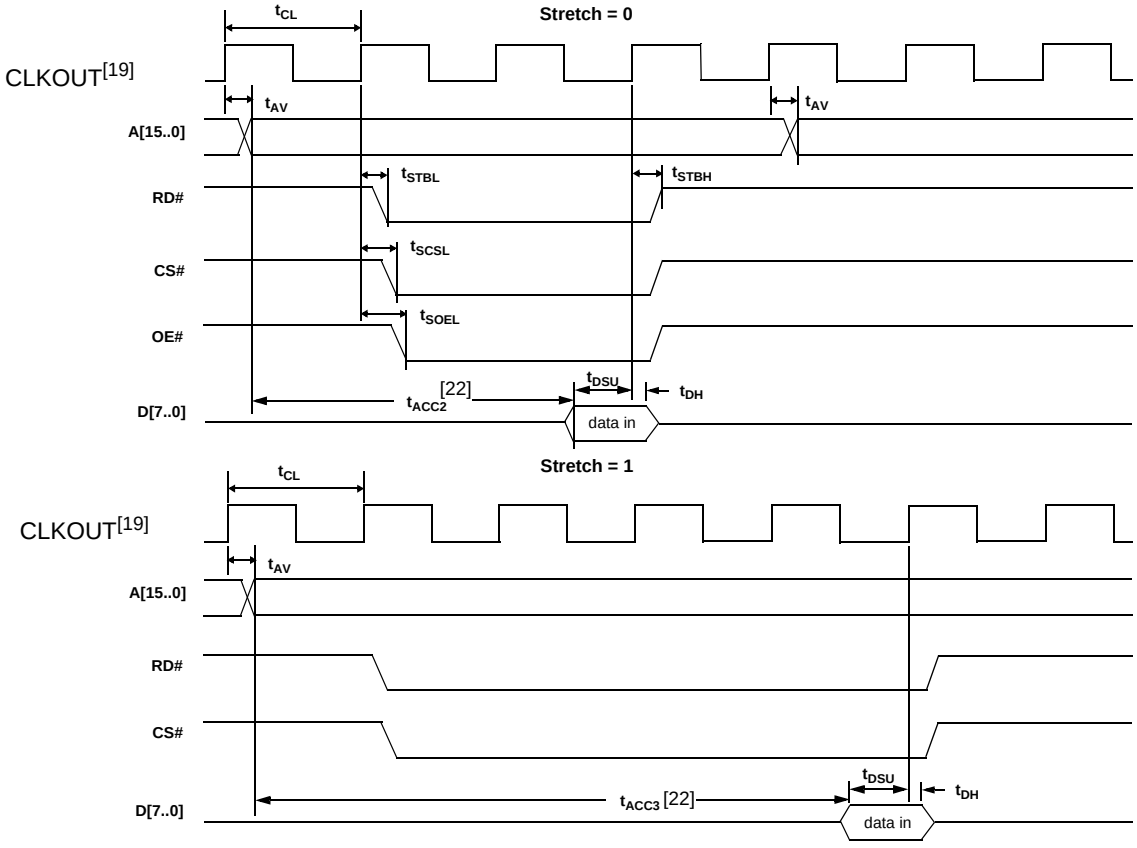



**Table 16. Data Memory Read Parameters**

| 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 is only active while either RD# or WR# is active. The address of AUTOPTR2 is active throughout the cycle and meets the address valid time for which is based on the stretch value.

**Notes**

21. The stretch memory cycle feature enables EZ-USB firmware to adjust the speed of data memory accesses not the program memory accesses. Details including typical strobe width timings can be found in the section 12.1.2 of the Technical Reference Manual. The address cycle width can be interpreted from these.
22. $t_{ACC2}$ and $t_{ACC3}$ are computed from these 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

## Data Memory Write[23]

**Figure 14. Data Memory Write Timing Diagram**

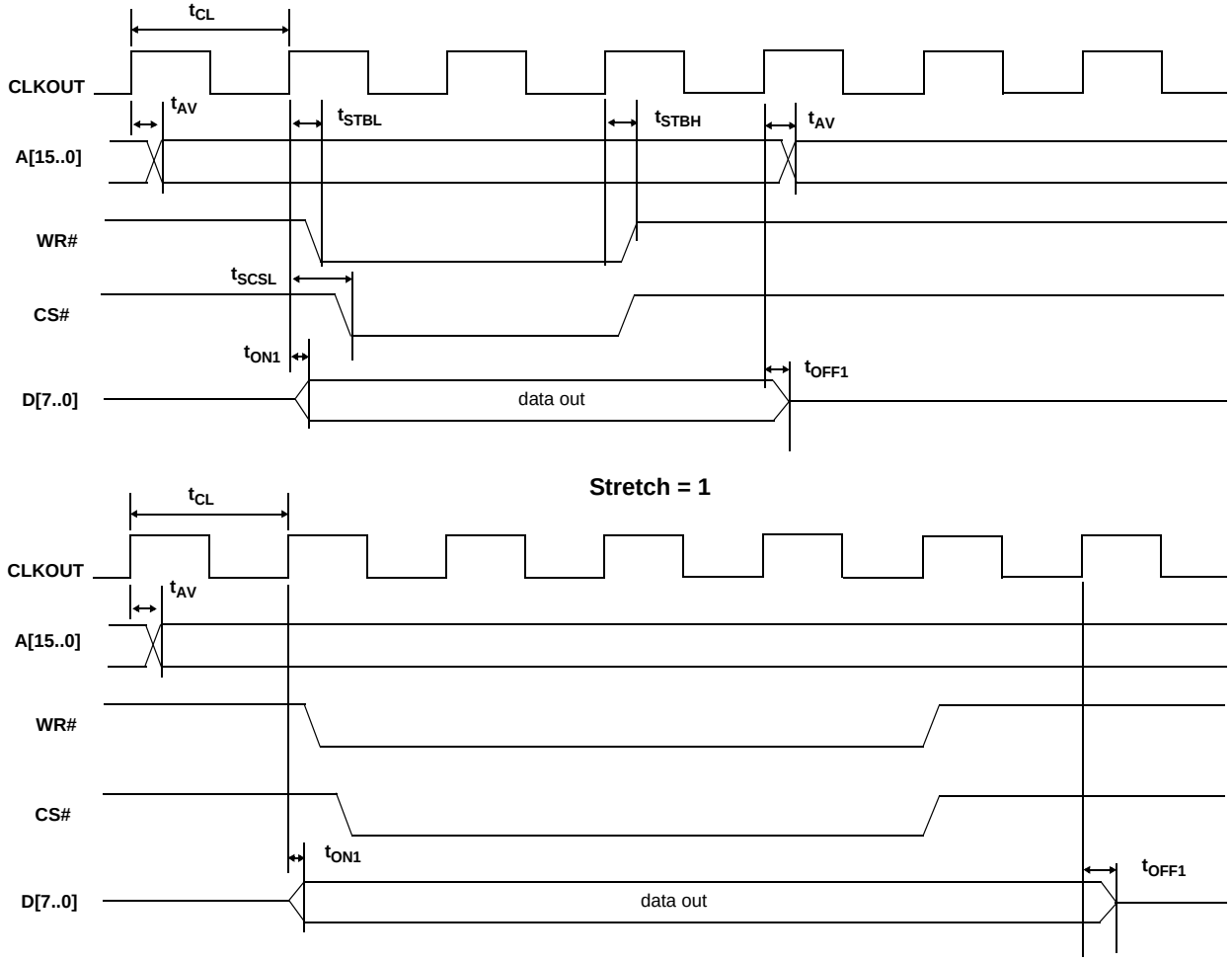



**Table 17. Data Memory Write Parameters**

| 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 is only active while either RD# or WR# are active. The address of AUTOPTR2 is active throughout the cycle and meets the address valid time for which is based on the stretch value.

**Note**
23. The stretch memory cycle feature enables EZ-USB firmware to adjust the speed of data memory accesses not the program memory accesses. Details including typical strobe width timings can be found in the section 12.1.2 of the Technical Reference Manual. The address cycle width can be interpreted from these.

## 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**

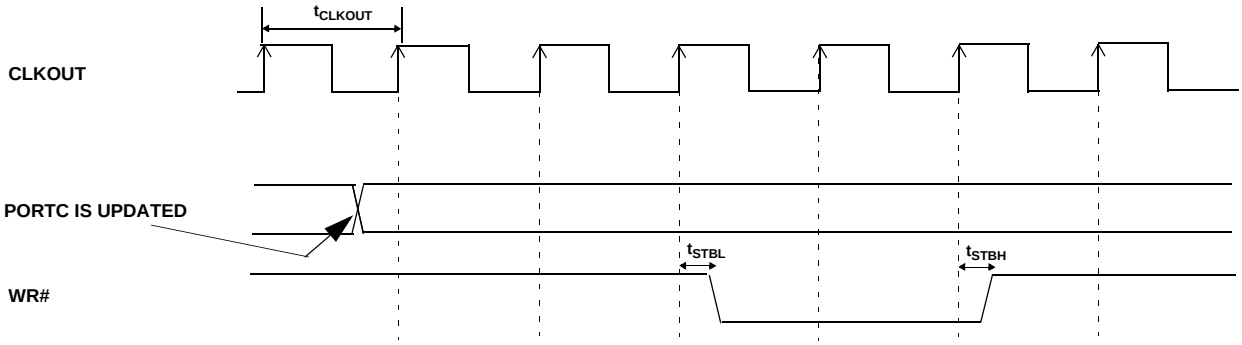
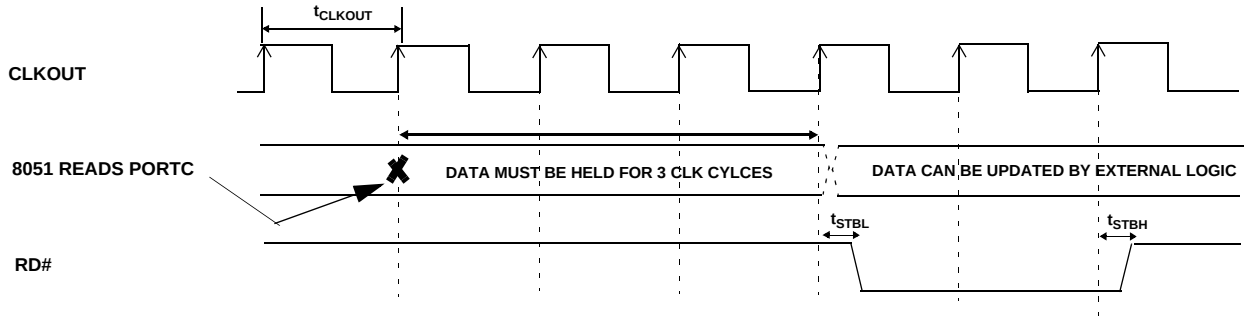



**Figure 17. RD# Strobe Function when PORTC is Accessed by 8051**



### Slave FIFO Synchronous Write

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

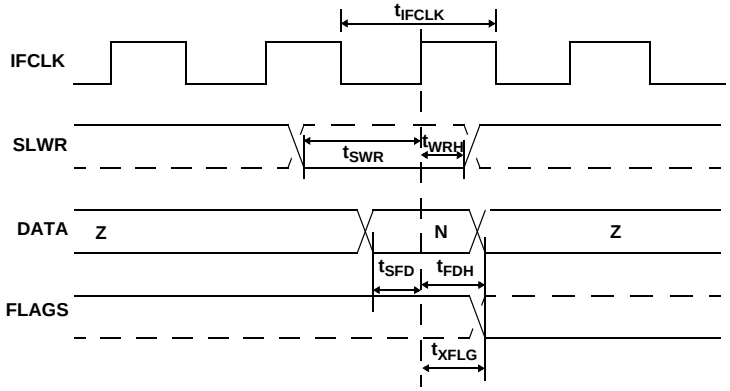



**Table 23. Slave FIFO Synchronous Write Parameters with Internally Sourced IFCLK[25]**

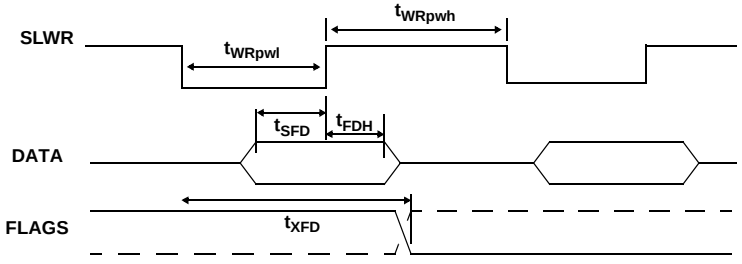
| Parameter | Description | Min | Max | Unit |
|---|---|---|---|---|
| $t_{IFCLK}$ | IFCLK period | 20.83 | – | ns |
| $t_{SWR}$ | SLWR to clock setup time | 10.4 | – | ns |
| $t_{WRH}$ | Clock to SLWR hold time | 0 | – | ns |
| $t_{SFD}$ | FIFO data to clock setup time | 9.2 | – | ns |
| $t_{FDH}$ | 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]**



### Slave FIFO Address to Flags/Data

**Figure 27. Slave FIFO Address to Flags/Data Timing Diagram[24]**

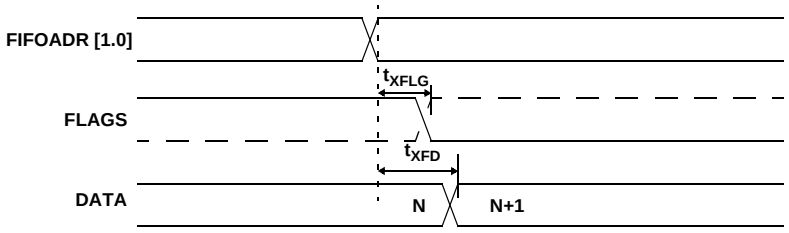



**Table 30. Slave FIFO Address to Flags/Data Parameters**

| 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 |

### Slave FIFO Synchronous Address

**Figure 28. Slave FIFO Synchronous Address Timing Diagram[24]**

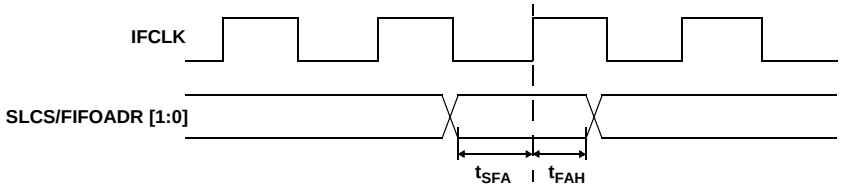



**Table 31. Slave FIFO Synchronous Address Parameters[25]**

| 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 |

### Slave FIFO Asynchronous Address

**Figure 29. Slave FIFO Asynchronous Address Timing Diagram[24]**

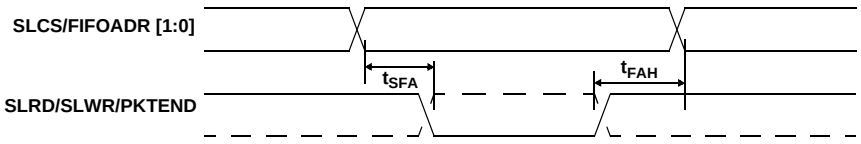



**Table 32. Slave FIFO Asynchronous Address Parameters[27]**

| Parameter | Description | Min | Max | Unit |
|---|---|---|---|---|
| $t_{SFA}$ | FIFOADR[1:0] to SLRD/SLWR/PKTEND setup time | 10 | – | ns |
| $t_{FAH}$ | RD/WR/PKTEND to FIFOADR[1:0] hold time | 10 | – | ns |

*Sequence Diagram of a Single and Burst Asynchronous Read*

**Figure 33. Slave FIFO Asynchronous Read Sequence and Timing Diagram**[24]

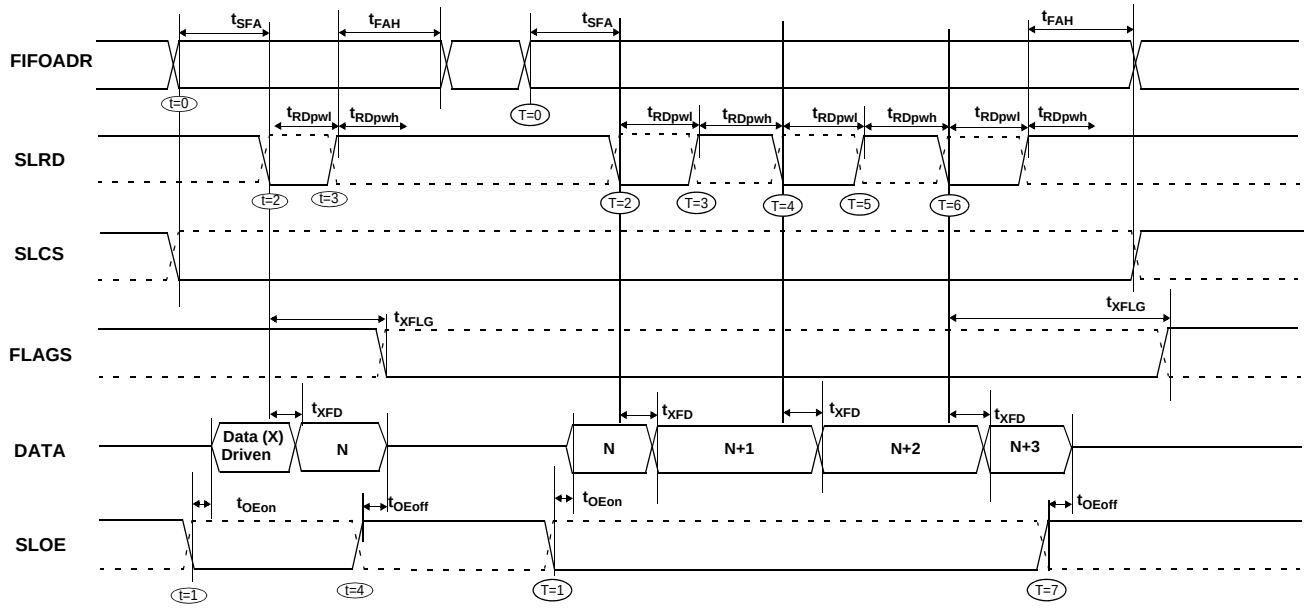



**Figure 34. Slave FIFO Asynchronous Read Sequence of Events Diagram**

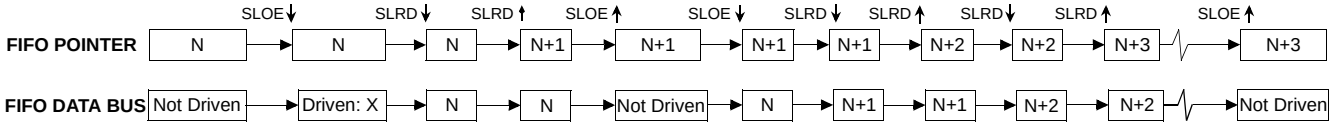


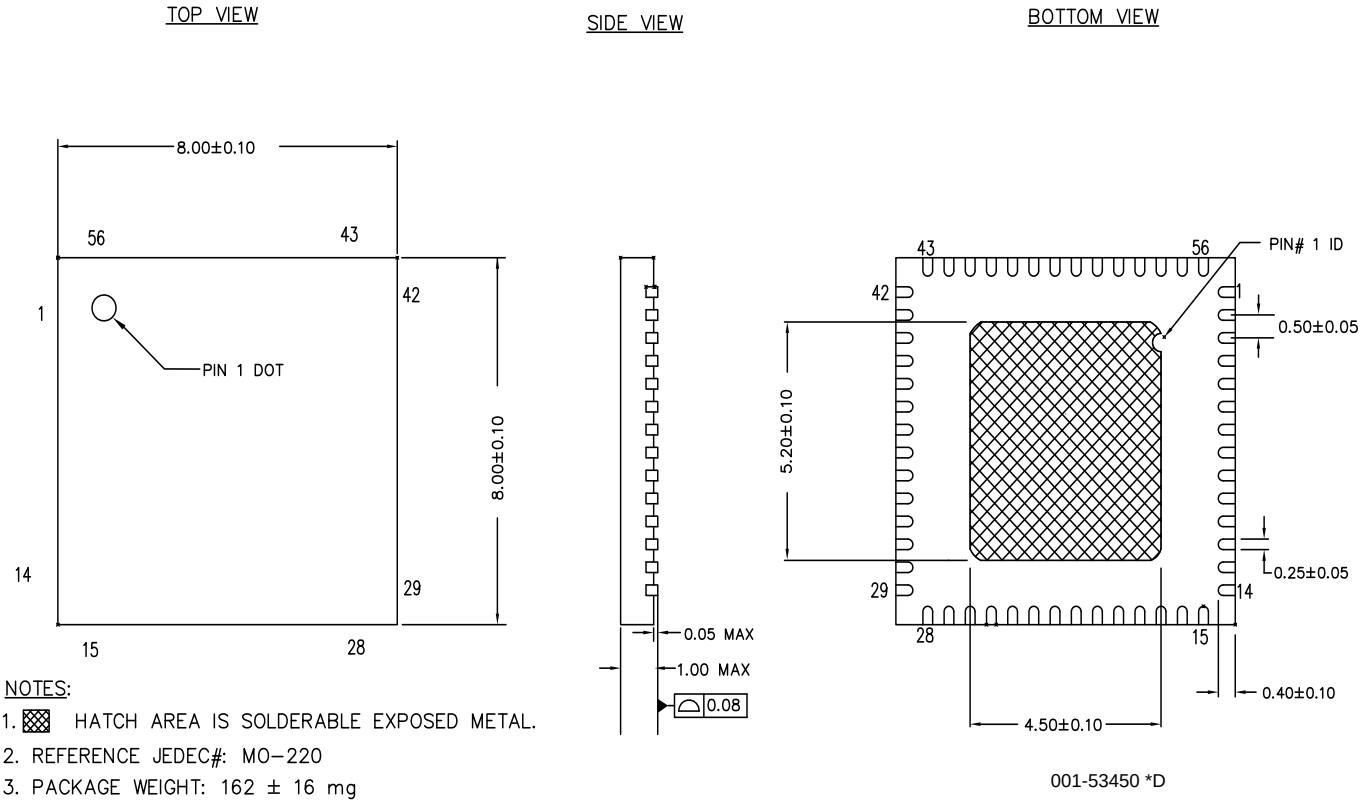

Figure 33 shows the timing relationship of the SLAVE FIFO signals during an asynchronous FIFO read. It shows a single read followed by a burst read.

- 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 the previous data, the data that was in the FIFO from anearlier read cycle.
- At t = 2, SLRD is asserted. The SLRD must meet the minimum active pulse of $t_{RDpwl}$ and minimum de-active pulse width of $t_{RDpwh}$. If SLCS is used, then SLCS must be asserted before SLRD is asserted (The SLCS and SLRD signals must both be asserted to start a valid read condition.)
- The data that is driven, after asserting SLRD, is the updated data from the FIFO. This data is valid after a propagation delay of $t_{XFD}$ from the activating edge of SLRD. In Figure 33, data N is the first valid data read from the FIFO. For data to appear on the data bus during the read cycle (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.

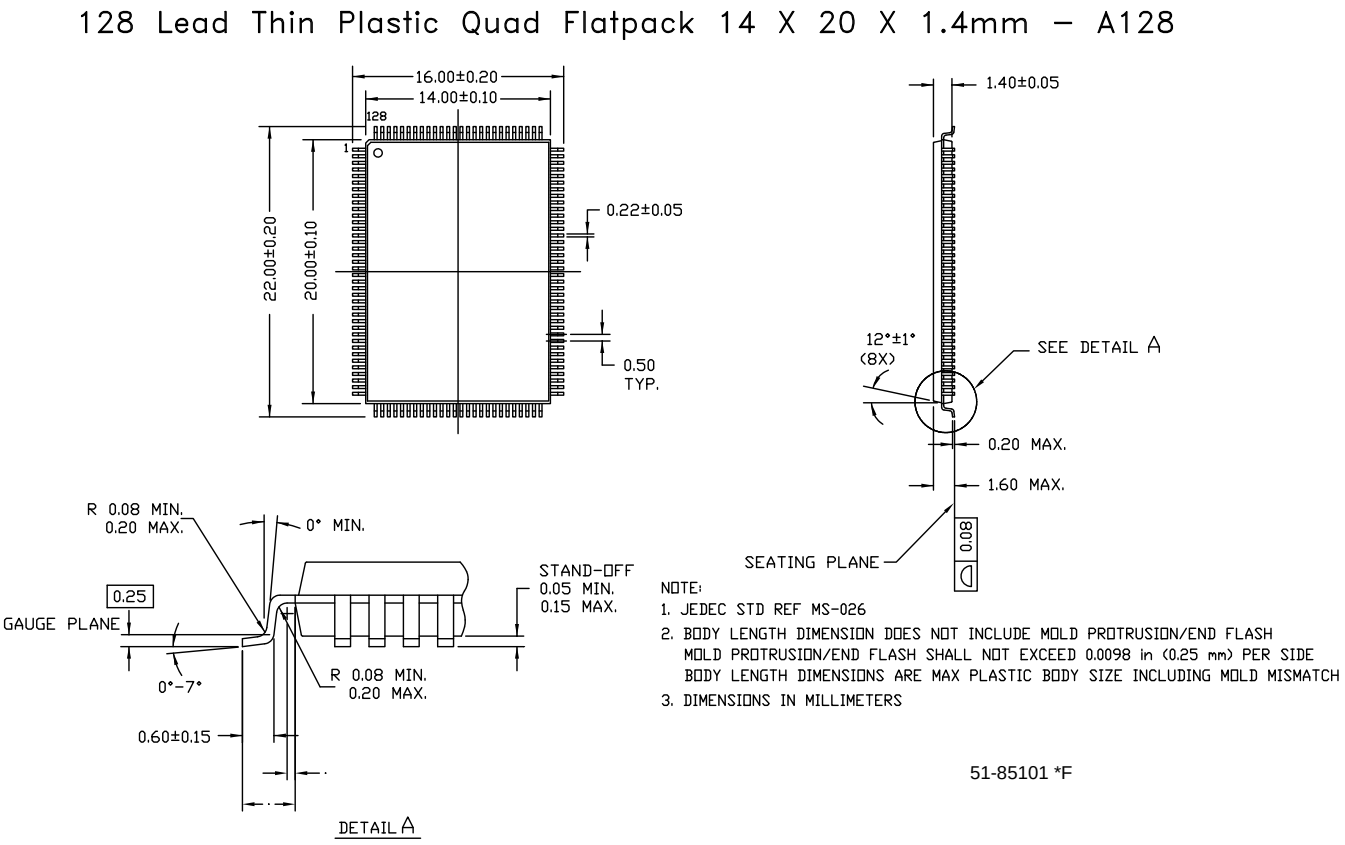
**Note** In the burst read mode, during SLOE is asserted, the data bus is in a driven state and outputs the previous data. After SLRD is asserted, the data from the FIFO is driven on the data bus (SLOE must also be asserted) and then the FIFO pointer is incremented.

**Figure 37. 56-Pin QFN 8 × 8 mm Sawn Version (001-53450)**




NOTES:
1. HATCH AREA IS SOLDERABLE EXPOSED METAL.
2. REFERENCE JEDEC#: MO–220
3. PACKAGE WEIGHT: 162 ± 16 mg
4. ALL DIMENSIONS ARE IN MILLIMETERS

**Figure 39. 128-Pin Thin Plastic Quad Flatpack (14 × 20 × 1.4 mm) A128 (51-85101)**



# Sales, Solutions, and Legal Information

## Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

### Products

| | |
|---|---|
| ARM® Cortex® Microcontrollers | cypress.com/arm |
| Automotive | cypress.com/automotive |
| Clocks & Buffers | cypress.com/clocks |
| Interface | cypress.com/interface |
| Lighting & Power Control | cypress.com/powerpsoc |
| Memory | cypress.com/memory |
| PSoC | cypress.com/psoc |
| Touch Sensing | cypress.com/touch |
| USB Controllers | cypress.com/usb |
| Wireless/RF | cypress.com/wireless |

### PSoC®Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

### Cypress Developer Community

Forums | Projects | Video | Blogs | Training | Components

### Technical Support

cypress.com/support