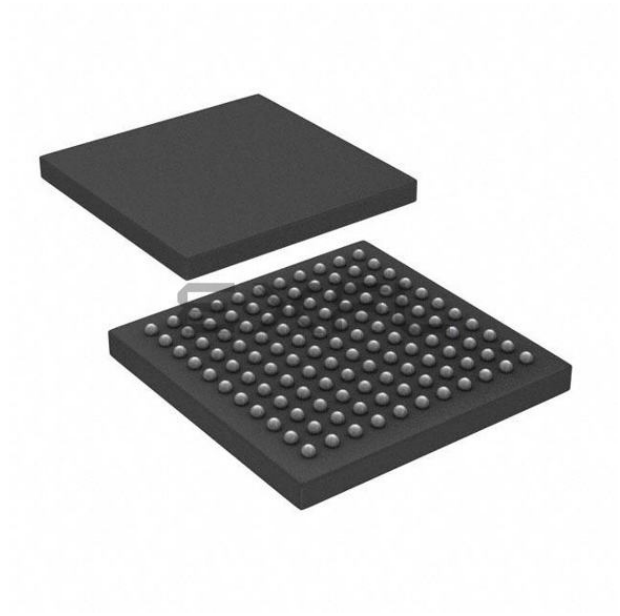


Welcome to E-XFL.COM

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	SuperSpeed USB Peripheral Controller
Core Processor	ARM9®
Program Memory Type	External Program Memory
Controller Series	CYUSB
RAM Size	512K x 8
Interface	GPIF, I²C, I²S, SPI, UART, USB
Number of I/O	60
Voltage - Supply	1.15V ~ 1.25V
Operating Temperature	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	121-TFBGA
Supplier Device Package	121-FBGA (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cyusb3014-bzxct

More Information

Cypress provides a wealth of data at www.cypress.com to help you to select the right <product> 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 knowledge base article KBA87889, How to design with FX3/FX3S.

- Overview: USB Portfolio, USB Roadmap
- USB 3.0 Product Selectors: FX3, FX3S, CX3, HX3, West Bridge Benicia
- 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 FX3 are:
  - AN75705 - Getting Started with EZ-USB FX3
  - AN76405 - EZ-USB FX3 Boot Options
  - AN70707 - EZ-USB FX3/FX3S Hardware Design Guidelines and Schematic Checklist
  - AN65974 - Designing with the EZ-USB FX3 Slave FIFO Interface
  - AN75779 - How to Implement an Image Sensor Interface with EZ-USB FX3 in a USB Video Class (UVC) Framework
  - AN86947 - Optimizing USB 3.0 Throughput with EZ-USB FX3
  - AN84868 - Configuring an FPGA over USB Using Cypress EZ-USB FX3
  - AN68829 - Slave FIFO Interface for EZ-USB FX3: 5-Bit Address Mode
  - AN73609 - EZ-USB FX2LP/ FX3 Developing Bulk-Loop Example on Linux
  - AN77960 - Introduction to EZ-USB FX3 High-Speed USB Host Controller
  - AN76348 - Differences in Implementation of EZ-USB FX2LP and EZ-USB FX3 Applications
  - AN89661 - USB RAID 1 Disk Design Using EZ-USB FX3S
- Code Examples: < Modify as required >
  - USB Hi-Speed
  - USB Full-Speed
  - USB SuperSpeed
- Technical Reference Manual (TRM):
  - EZ-USB FX3 Technical Reference Manual
- Development Kits:
  - CYUSB3KIT-003, EZ-USB FX3 SuperSpeed Explorer Kit
  - CYUSB3KIT-001, EZ-USB FX3 Development Kit
- Models: IBIS

EZ-USB FX3 Software Development Kit

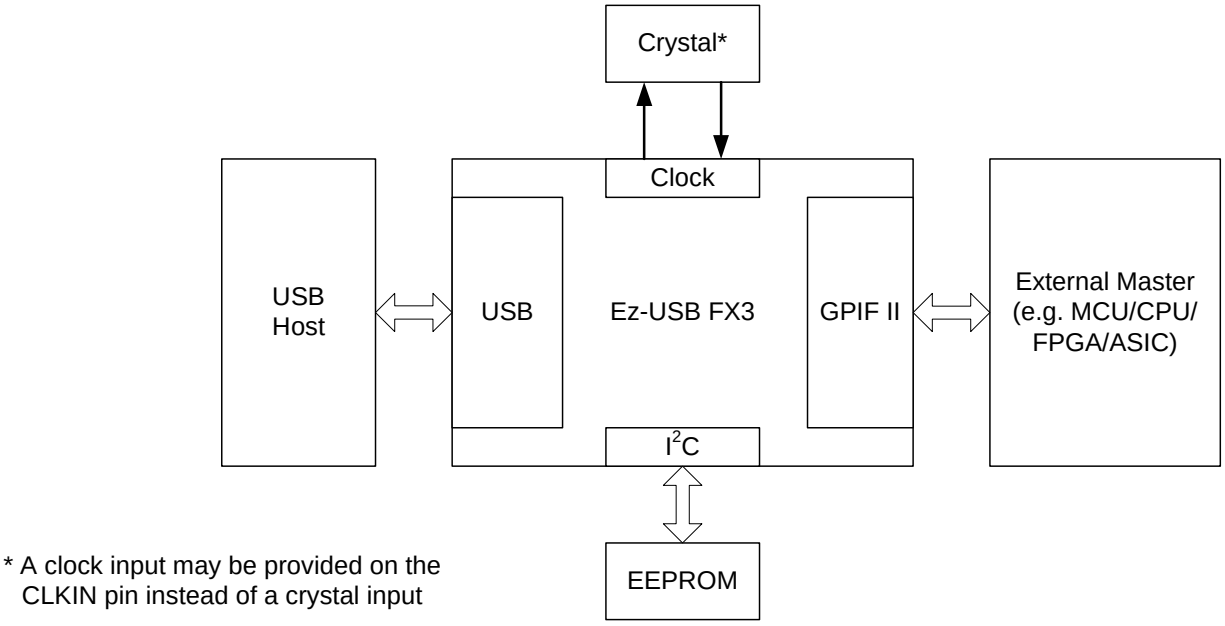
Cypress delivers the complete software and firmware stack for FX3, in order to easily integrate SuperSpeed USB into any embedded application. The Software Development Kit (SDK) comes with tools, drivers and application examples, which help accelerate application development.

GPIF™ II Designer

The GPIF II Designer is a graphical software that allows designers to configure the GPIF II interface of the EZ-USB FX3 USB 3.0 Device Controller.

The tool allows users the ability to select from one of five Cypress supplied interfaces, or choose to create their own GPIF II interface from scratch. Cypress has supplied industry standard interfaces such as Asynchronous and Synchronous Slave FIFO, Asynchronous and Synchronous SRAM, and Asynchronous SRAM. Designers who already have one of these pre-defined interfaces in their system can simply select the interface of choice, choose from a set of standard parameters such as bus width (x8, 16, x32) endianess, clock settings, and compile the interface. The tool has a streamlined three step GPIF interface development process for users who need a customized interface. Users are able to first select their pin configuration and standard parameters. Secondly, they can design a virtual state machine using configurable actions. Finally, users can view output timing to verify that it matches the expected timing. Once the three step process is complete, the interface can be compiled and integrated with FX3.

Figure 2. EZ-USB FX3 as a Coprocessor



USB Interface

FX3 complies with the following specifications and supports the following features:

- Supports USB peripheral functionality compliant with USB 3.1 Specification Revision 1.0 and is also backward compatible with the USB 2.0 Specification.
- FX3 Hi-Speed parts (CYUSB201X) only support USB 2.0.
- Complies with OTG Supplement Revision 2.0. It supports High-Speed, Full-Speed, and Low-Speed OTG dual-role device capability. As a peripheral, FX3 is capable of SuperSpeed, High-Speed, and Full-Speed. As a host, it is capable of High-Speed, Full-Speed, and Low-Speed.
- Supports Carkit Pass-Through UART functionality on USB D+/D– lines based on the CEA-936A specification.
- Supports 16 IN and 16 OUT endpoints.
- Supports the USB 3.0 Streams feature. It also supports USB Attached SCSI (UAS) device-class to optimize mass-storage access performance.
- As a USB peripheral, application examples show that the FX3 supports UAS, USB Video Class (UVC), and Mass Storage Class (MSC) USB peripheral classes. All other device classes can be supported by customer firmware; a template example is provided as a starting point.
- As an OTG host, application examples show that FX3 supports MSC and HID device classes.

Note When the USB port is not in use, disable the PHY and transceiver to save power.

OTG

FX3 is compliant with the OTG Specification Revision 2.0. In OTG mode, FX3 supports both A and B device modes and supports Control, Interrupt, Bulk, and Isochronous data transfers.

FX3 requires an external charge pump (either standalone or integrated into a PMIC) to power VBUS in the OTG A-device mode.

The Target Peripheral List for OTG host implementation consists of MSC- and HID-class devices.

FX3 does not support Attach Detection Protocol (ADP).

OTG Connectivity

In OTG mode, FX3 can be configured to be an A, B, or dual-role device. It can connect to the following:

- ACA device
- Targeted USB peripheral
- SRP-capable USB peripheral
- HNP-capable USB peripheral
- OTG host
- HNP-capable host
- OTG device

ReNumeration

Because of FX3's soft configuration, one chip can take on the identities of multiple distinct USB devices.

When first plugged into USB, FX3 enumerates automatically with the Cypress Vendor ID (0x04B4) and downloads firmware and USB descriptors over the USB interface. The downloaded firmware executes an electrical disconnect and connect. FX3 enumerates again, this time as a device defined by the downloaded information. This patented two-step process, called ReNumeration, happens instantly when the device is plugged in.

EZ-Dtect

FX3 supports USB Charger and accessory detection (EZ-Dtect). The charger detection mechanism complies with the Battery Charging Specification Revision 1.1. In addition to supporting this version of the specification, FX3 also provides hardware support to detect the resistance values on the ID pin.

FX3 can detect the following resistance ranges:

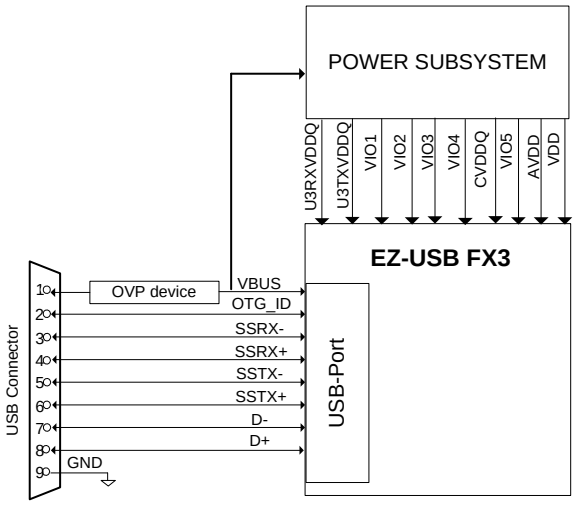
- Less than 10 Ω
- Less than 1 kΩ
- 65 kΩ to 72 kΩ
- 35 kΩ to 39 kΩ
- 99.96 kΩ to 104.4 kΩ (102 kΩ ± 2%)
- 119 kΩ to 132 kΩ
- Higher than 220 kΩ
- 431.2 kΩ to 448.8 kΩ (440 kΩ ± 2%)

FX3's charger detects a dedicated wall charger, Host/Hub charger, and Host/Hub.

VBUS Overvoltage Protection

The maximum input voltage on FX3's VBUS pin is 6 V. A charger can supply up to 9 V on VBUS. In this case, an external overvoltage protection (OVP) device is required to protect FX3 from damage on VBUS. Figure 3 shows the system application diagram with an OVP device connected on VBUS. Refer to Table 8 for the operating range of VBUS and VBATT.

Figure 3. System Diagram with OVP Device For VBUS



Carkit UART Mode

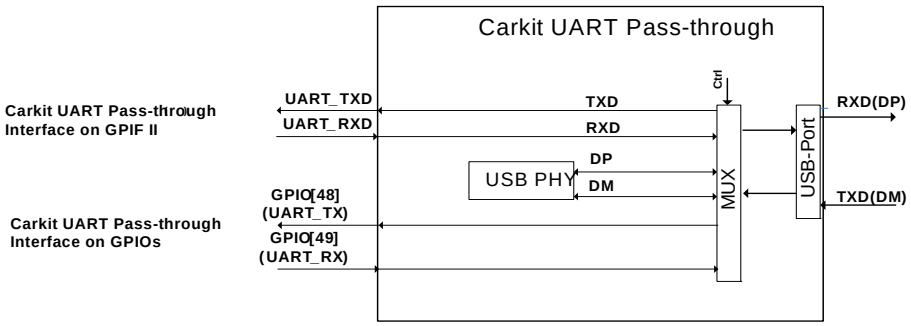
The USB interface supports the Carkit UART mode (UART over D+/D–) for non-USB serial data transfer. This mode is based on the CEA-936A specification.

In the Carkit UART mode, the output signaling voltage is 3.3 V. When configured for the Carkit UART mode, TXD of UART (output) is mapped to the D– line, and RXD of UART (input) is mapped to the D+ line.

In the Carkit UART mode, FX3 disables the USB transceiver and D+ and D– pins serve as pass-through pins to connect to the UART of the host processor. The Carkit UART signals may be routed to the GPIF II interface or to GPIO[48] and GPIO[49], as shown in Figure on page 8.

In this mode, FX3 supports a rate of up to 9600 bps.

Figure 4. Carkit UART Pass-through Block Diagram



GPIF II

The high-performance GPIF II interface enables functionality similar to, but more advanced than, FX2LP's GPIF and Slave FIFO interfaces.

The GPIF II is a programmable state machine that enables a flexible interface that may function either as a master or slave in industry-standard or proprietary interfaces. Both parallel and serial interfaces may be implemented with GPIF II.

Here is a list of GPIF II features:

- Functions as master or slave
- Provides 256 firmware programmable states
- Supports 8-bit, 16-bit, 24-bit, and 32-bit parallel data bus
- Enables interface frequencies up to 100 MHz
- Supports 14 configurable control pins when a 32- bit data bus is used. All control pins can be either input/output or bidirectional.
- Supports 16 configurable control pins when a 16/8 data bus is used. All control pins can be either input/output or bi-directional.

GPIF II state transitions are based on control input signals. The control output signals are driven as a result of the GPIF II state transitions. The INT# output signal can be controlled by GPIF II. Refer to the GPIFII Designer tool. The GPIF II state machine's behavior is defined by a GPIF II descriptor. The GPIF II descriptor is designed such that the required interface specifications are met. 8 KB of memory (separate from the 256/512 KB of embedded SRAM) is dedicated to the GPIF II waveform where the GPIF II descriptor is stored in a specific format.

Cypress's GPIFII Designer Tool enables fast development of GPIF II descriptors and includes examples for common interfaces.

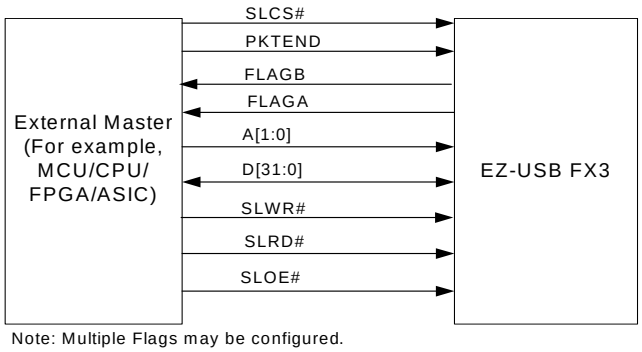
Example implementations of GPIF II are the asynchronous slave FIFO and synchronous slave FIFO interfaces.

Slave FIFO interface

The Slave FIFO interface signals are shown in Figure 5. This interface allows an external processor to directly access up to four buffers internal to FX3. Further details of the Slave FIFO interface are described on page 24.

Note Access to all 32 buffers is also supported over the slave FIFO interface. For details, contact Cypress Applications Support.

Figure 5. Slave FIFO Interface



Note: Multiple Flags may be configured.

CPU

FX3 has an on-chip 32-bit, 200-MHz ARM926EJ-S core CPU. The core has direct access to 16 KB of Instruction Tightly Coupled Memory (TCM) and 8 KB of Data TCM. The ARM926EJ-S core provides a JTAG interface for firmware debugging.

FX3 offers the following advantages:

- Integrates 256/512 KB of embedded SRAM for code and data and 8 KB of Instruction cache and Data cache.
- Implements efficient and flexible DMA connectivity between the various peripherals (such as, USB, GPIF II, I^2S, SPI, UART, I^2C), requiring firmware only to configure data accesses between peripherals, which are then managed by the DMA fabric.
- Allows easy application development using industry-standard development tools for ARM926EJ-S.

Examples of the FX3 firmware are available with the Cypress EZ-USB FX3 Development Kit.

JTAG Interface

FX3's JTAG interface has a standard five-pin interface to connect to a JTAG debugger in order to debug firmware through the CPU-core's on-chip-debug circuitry.

Industry-standard debugging tools for the ARM926EJ-S core can be used for the FX3 application development.

Other Interfaces

FX3 supports the following serial peripherals:

- SPI
- UART
- I^2C
- I^2S

The SPI, UART, and I^2S interfaces are multiplexed on the serial peripheral port.

The CYUSB3012 and CYUSB3014 Pin List on page 15 shows details of how these interfaces are multiplexed. Note that when GPIF II is configured for a 32-bit data bus width (CYUSB3012 and CYUSB3014), then the SPI interface is not available.

SPI Interface

FX3 supports an SPI Master interface on the Serial Peripherals port. The maximum operation frequency is 33 MHz.

The SPI controller supports four modes of SPI communication (see SPI Timing Specification on page 40 for details on the modes) with the Start-Stop clock. This controller is a single-master controller with a single automated SSN control. It supports transaction sizes ranging from four bits to 32 bits.

Table 6. Entry and Exit Methods for Low-Power Modes (continued)

Low-Power Mode	Characteristics	Methods of Entry	Methods of Exit
Suspend Mode with USB 3.0 PHY Disabled (L2)	■ The power consumption in this mode does not exceed ISB_2 ■ USB 3.0 PHY is disabled and the USB interface is in suspend mode ■ The clocks are shut off. The PLLs are disabled ■ All I/Os maintain their previous state ■ USB interface maintains the previous state ■ Power supply for the wakeup source and core power must be retained. All other power domains can be turned on/off individually ■ The states of the configuration registers, buffer memory and all internal RAM are maintained ■ All transactions must be completed before FX3 enters Suspend mode (state of outstanding transactions are not preserved) ■ The firmware resumes operation from where it was suspended (except when woken up by RESET# assertion) because the program counter does not reset	■ Firmware executing on ARM926EJ-S core can put FX3 into suspend mode. For example, on USB suspend condition, firmware may decide to put FX3 into suspend mode ■ External Processor, through the use of mailbox registers can put FX3 into suspend mode	■ D+ transitioning to low or high ■ D- transitioning to low or high ■ Impedance change on OTG_ID pin ■ Resume condition on SSRX± ■ Detection of VBUS ■ Level detect on UART_CTS (programmable polarity) ■ GPIF II interface assertion of CTL[0] ■ Assertion of RESET#
Standby Mode (L3)	■ The power consumption in this mode does not exceed ISB3 ■ All configuration register settings and program/data RAM contents are preserved. However, data in the buffers or other parts of the data path, if any, is not guaranteed. Therefore, the external processor should take care that the data needed is read before putting FX3 into this Standby Mode ■ The program counter is reset after waking up from Standby ■ GPIO pins maintain their configuration ■ Crystal oscillator is turned off ■ Internal PLL is turned off ■ USB transceiver is turned off ■ ARM926EJ-S core is powered down. Upon wakeup, the core re-starts and runs the program stored in the program/data RAM ■ Power supply for the wakeup source and core power must be retained. All other power domains can be turned on/off individually	■ Firmware executing on ARM926EJ-S core or external processor configures the appropriate register	■ Detection of VBUS ■ Level detect on UART_CTS (Programmable Polarity) ■ GPIF II interface assertion of CTL[0] ■ Assertion of RESET#

Table 6. Entry and Exit Methods for Low-Power Modes (continued)

Low-Power Mode	Characteristics	Methods of Entry	Methods of Exit
Core Power Down Mode (L4)	■ The power consumption in this mode does not exceed ISB_4 ■ Core power is turned off ■ All buffer memory, configuration registers, and the program RAM do not maintain state. After exiting this mode, reload the firmware ■ In this mode, all other power domains can be turned on/off individually	■ Turn off V_{DD}	■ Reapply VDD ■ Assertion of RESET#

Digital I/Os

FX3 has internal firmware-controlled pull-up or pull-down resistors on all digital I/O pins. An internal 50-kΩ resistor pulls the pins high, while an internal 10-kΩ resistor pulls the pins low to prevent them from floating. The I/O pins may have the following states:

- Tristated (High-Z)
- Weak pull-up (via internal 50 kΩ)
- Pull-down (via internal 10 kΩ)
- Hold (I/O hold its value) when in low-power modes
- The JTAG TDI, TMS, and TRST# signals have fixed 50-kΩ internal pull-ups, and the TCK signal has a fixed 10-kΩ pull-down resistor.

All unused I/Os should be pulled high by using the internal pull-up resistors. All unused outputs should be left floating. All I/Os can be driven at full-strength, three-quarter strength, half-strength, or quarter-strength. These drive strengths are configured separately for each interface.

GPIOs

EZ-USB enables a flexible pin configuration both on the GPIF II and the serial peripheral interfaces. Any unused control pins (except CTL[15]) on the GPIF II interface can be used as GPIOs. Similarly, any unused pins on the serial peripheral interfaces may be configured as GPIOs. See Pin Configurations for pin configuration options.

All GPIF II and GPIO pins support an external load of up to 16 pF for every pin.

EMI

FX3 meets EMI requirements outlined by FCC 15B (USA) and EN55022 (Europe) for consumer electronics. FX3 can tolerate EMI, conducted by the aggressor, outlined by these specifications and continue to function as expected.

System-level ESD

FX3 has built-in ESD protection on the D+, D–, and GND pins on the USB interface. The ESD protection levels provided on these ports are:

- ±2.2-kV human body model (HBM) based on JESD22-A114 Specification
- ±6-kV contact discharge and ±8-kV air gap discharge based on IEC61000-4-2 level 3A
- ± 8-kV Contact Discharge and ±15-kV Air Gap Discharge based on IEC61000-4-2 level 4C.

This protection ensures the device continues to function after ESD events up to the levels stated in this section.

The SSRX+, SSRX–, SSTX+, and SSTX– pins only have up to ±2.2-kV HBM internal ESD protection.

Table 7. CYUSB3012 and CYUSB3014 Pin List (continued)

BGA	WLCSP	Power Domain	I/O	Name	Description
B9	E4	–	PWR	VSS	–
F11	F6	–	PWR	VDD	–
–	E5	–	PWR	VSS	**GND**
–	F7	–	PWR	VDD	–
–	E6	–	PWR	VSS	GND
–	E7	–	PWR	VSS	GND
H1	G6	–	PWR	VDD	–
L7	D7	–	PWR	VDD	–
J11	L10	–	PWR	VDD	–
L5	L12	–	PWR	VDD	–
K4	H7	–	PWR	VSS	–
L3	G7	–	PWR	VSS	–
K3	L11	–	PWR	VSS	–
L2	G8	–	PWR	VSS	–
A8	G5	–	PWR	VSS	–
–	B4	–	–	NC	No Connect
A11	B2	–	–	NC	No Connect

AC Timing Parameters

GPIF II Timing

Figure 9. GPIF II Timing in Synchronous Mode

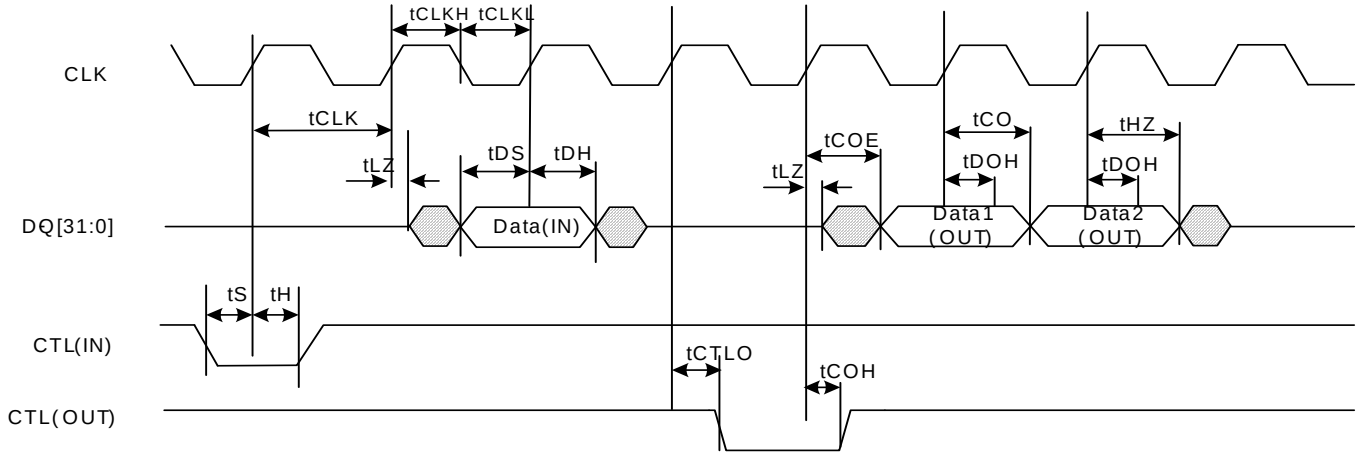


Table 9. GPIF II Timing Parameters in Synchronous Mode [2]

Parameter	Description	Min	Max	Units
Frequency	Interface clock frequency	–	100	MHz
tCLK	Interface clock period	10	–	ns
tCLKH	Clock high time	4	–	ns
tCLKL	Clock low time	4	–	ns
tS	CTL input to clock setup time	2	–	ns
tH	CTL input to clock hold time	0.5	–	ns
tDS	Data in to clock setup time	2	–	ns
tDH	Data in to clock hold time	0.5	–	ns
tCO	Clock to data out propagation delay when DQ bus is already in output direction	–	7	ns
tCOE	Clock to data out propagation delay when DQ lines change to output from tristate and valid data is available on the DQ bus	–	9	ns
tCTLO	Clock to CTL out propagation delay	–	8	ns
tDOH	Clock to data out hold	2	–	ns
tCOH	Clock to CTL out hold	0	–	ns
tHZ	Clock to high-Z	–	8	ns
tLZ	Clock to low-Z	0	–	ns

Note
2. All parameters guaranteed by design and validated through characterization.

Figure 14. Synchronous Slave FIFO ZLP Write Cycle Timing

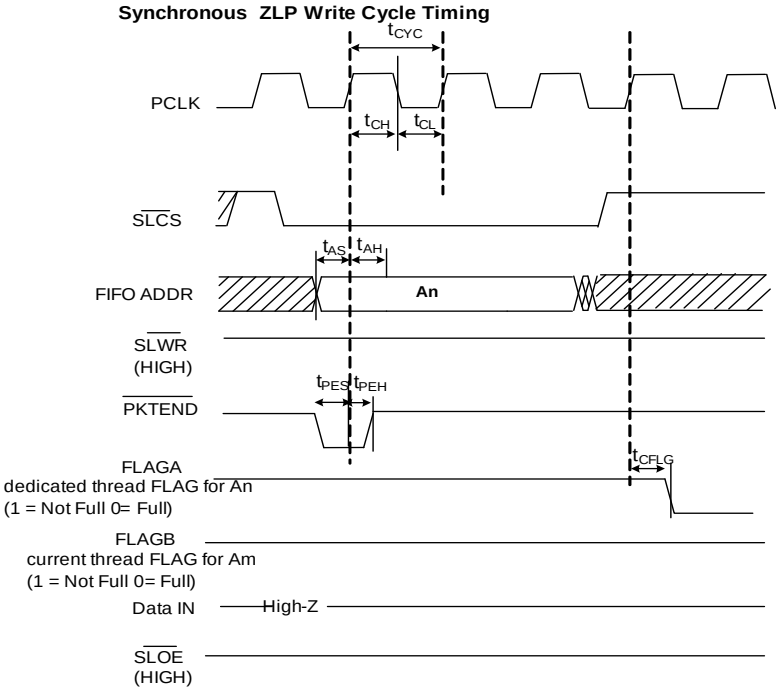


Table 11. Synchronous Slave FIFO Parameters[5]

Parameter	Description	Min	Max	Units
FREQ	Interface clock frequency	–	100	MHz
tCYC	Clock period	10	–	ns
tCH	Clock high time	4	–	ns
tCL	Clock low time	4	–	ns
tRDS	SLRD# to CLK setup time	2	–	ns
tRDH	SLRD# to CLK hold time	0.5	–	ns
tWRS	SLWR# to CLK setup time	2	–	ns
tWRH	SLWR# to CLK hold time	0.5	–	ns
tCO	Clock to valid data	–	7	ns
tDS	Data input setup time	2	–	ns
tDH	CLK to data input hold	0.5	–	ns
tAS	Address to CLK setup time	2	–	ns
tAH	CLK to address hold time	0.5	–	ns
tOELZ	SLOE# to data low-Z	0	–	ns
tCFLG	CLK to flag output propagation delay	–	8	ns
tOEZ	SLOE# deassert to Data Hi Z	–	8	ns
tPES	PKTEND# to CLK setup	2	–	ns
tPEH	CLK to PKTEND# hold	0.5	–	ns
tCDH	CLK to data output hold	2	–	ns
tSSD	Socket switching delay	2	68	Clock cycles
tACCD	Latency from SLRD# to Data	2	2	Clock cycles
tFAD	Latency from SLWR# to FLAG	3	3	Clock cycles
Note Three-cycle latency from ADDR to DATA/FLAGS.				

Note
5. All parameters guaranteed by design and validated through characterization.

Asynchronous Slave FIFO Write Sequence Description

- FIFO address is driven and SLCS# is asserted
- SLWR# is asserted. SLCS# must be asserted with SLWR# or before SLWR# is asserted
- Data must be present on the tWRS bus before the deasserting edge of SLWR#
- Deassertion of SLWR# causes the data to be written from the data bus to the FIFO, and then the FIFO pointer is incremented
- The FIFO flag is updated after the tWFLG from the deasserting edge of SLWR.

The same sequence of events is shown for a burst write.

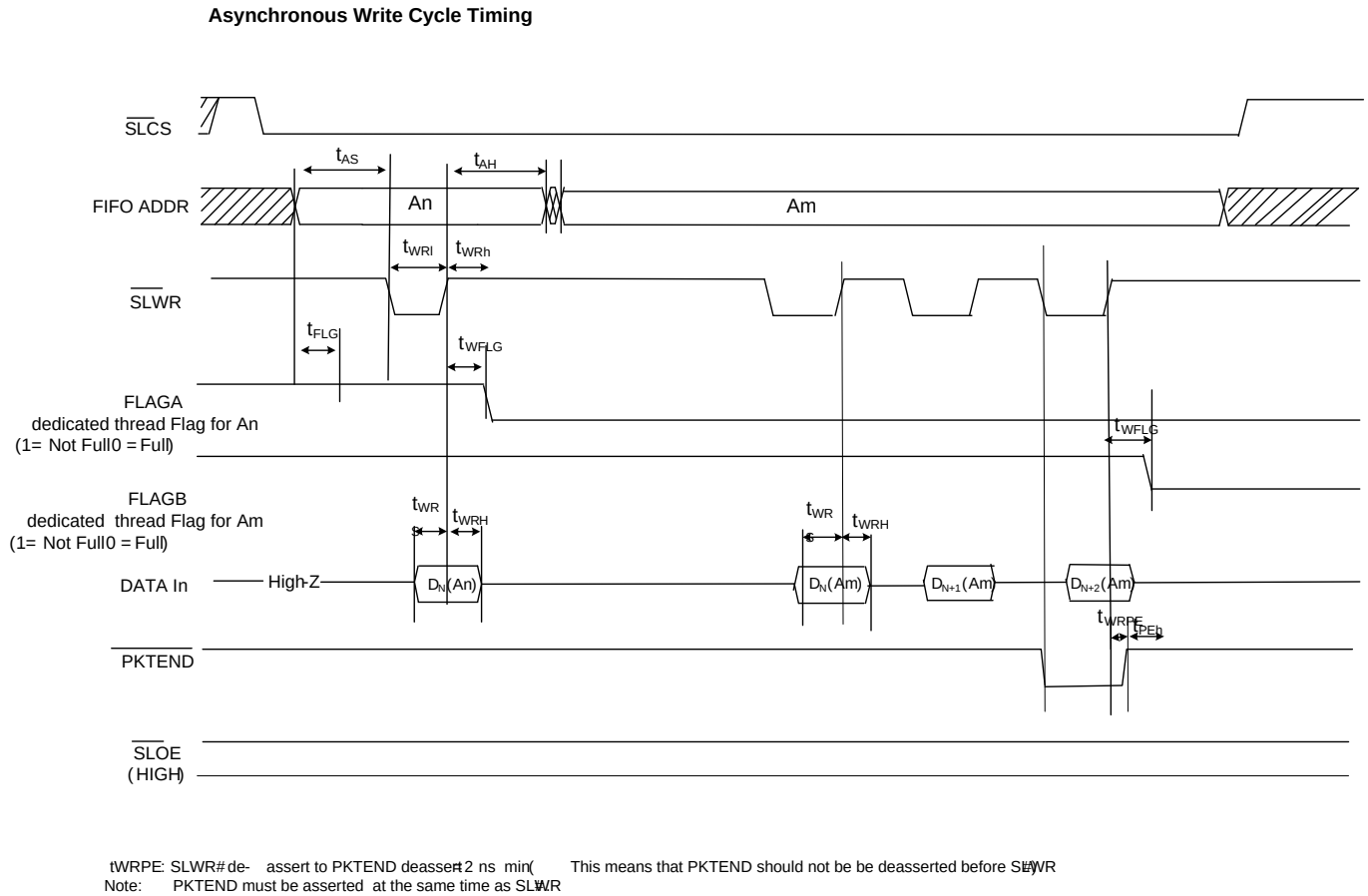
Note that in the burst write mode, after SLWR# deassertion, the data is written to the FIFO, and then the FIFO pointer is incremented.

Short Packet: A short packet can be committed to the USB host by using the PKTEND#. The external device or processor should be designed to assert the PKTEND# along with the last word of data and SLWR# pulse corresponding to the last word. The FIFOADDR lines must be held constant during the PKTEND# assertion.

Zero-Length Packet: The external device or processor can signal a zero-length packet (ZLP) to FX3 simply by asserting PKTEND#, without asserting SLWR#. SLCS# and the address must be driven as shown in Figure 17 on page 29.

FLAG Usage: The FLAG signals are monitored by the external processor for flow control. FLAG signals are FX3 outputs that can be configured to show empty, full, and partial status for a dedicated address or the current address.

Figure 16. Asynchronous Slave FIFO Write Mode



tWRPE: SLWR# de- assert to PKTEND deasser+2 ns min(This means that PKTEND should not be be deasserted before SLWR
Note: PKTEND must be asserted at the same time as SLWR

Host Processor Interface (P-Port) Timing

Asynchronous SRAM Timing

Figure 18. Non-multiplexed Asynchronous SRAM Read Timing

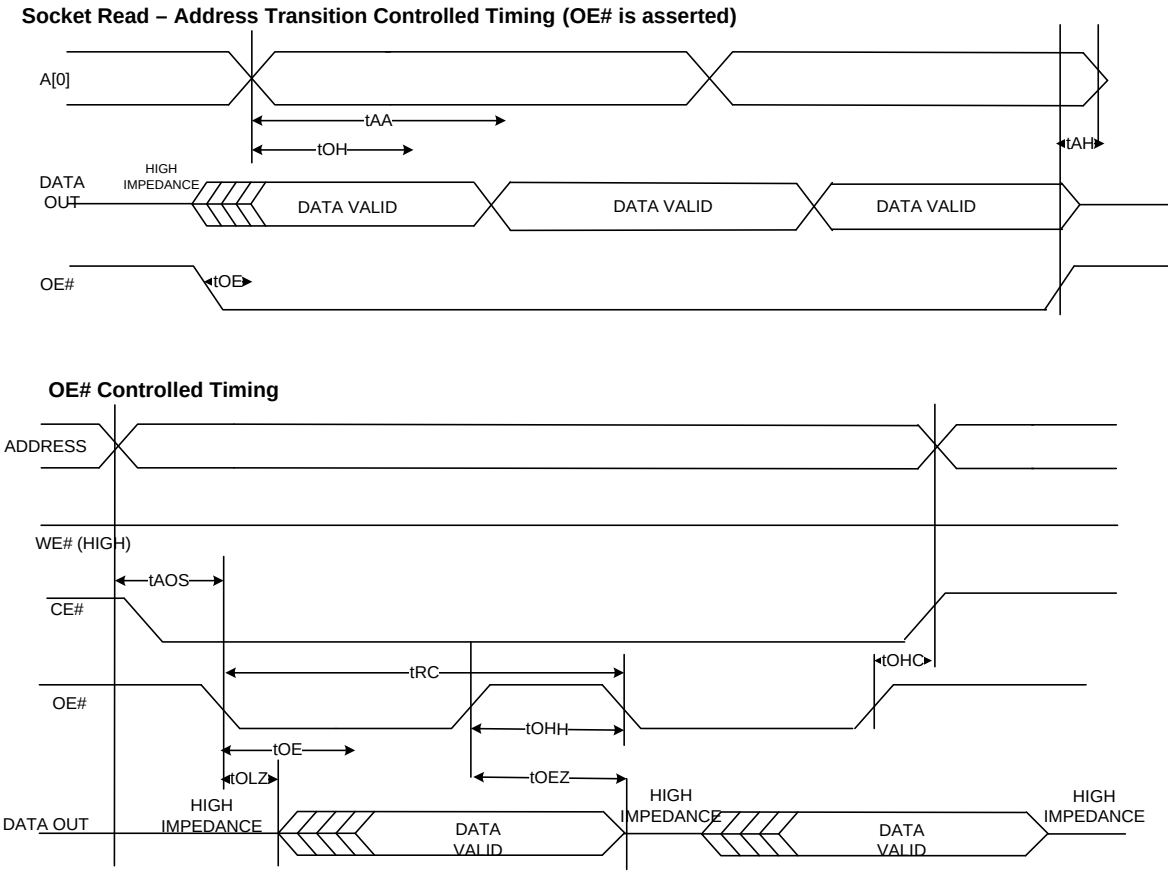
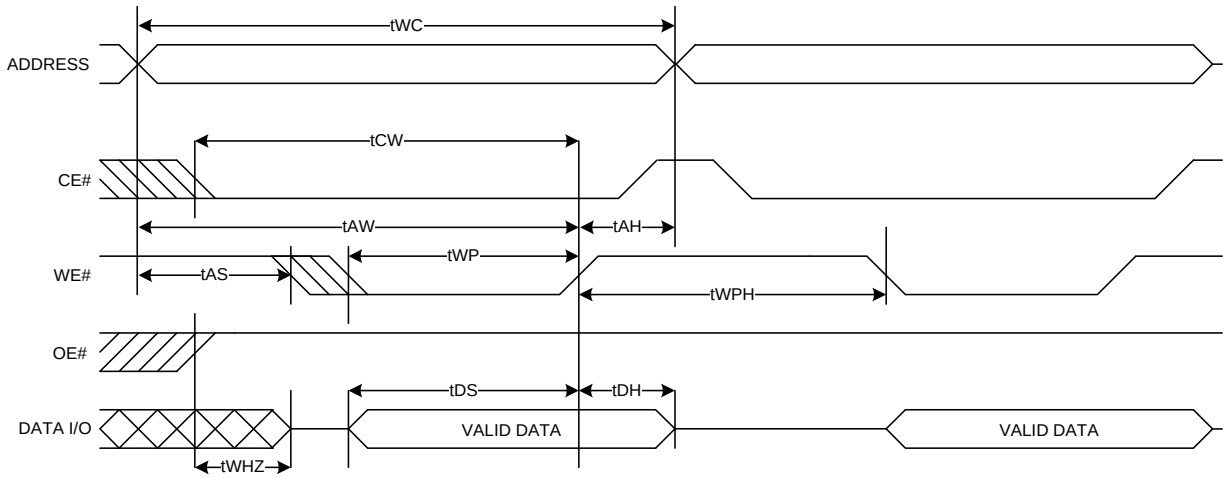
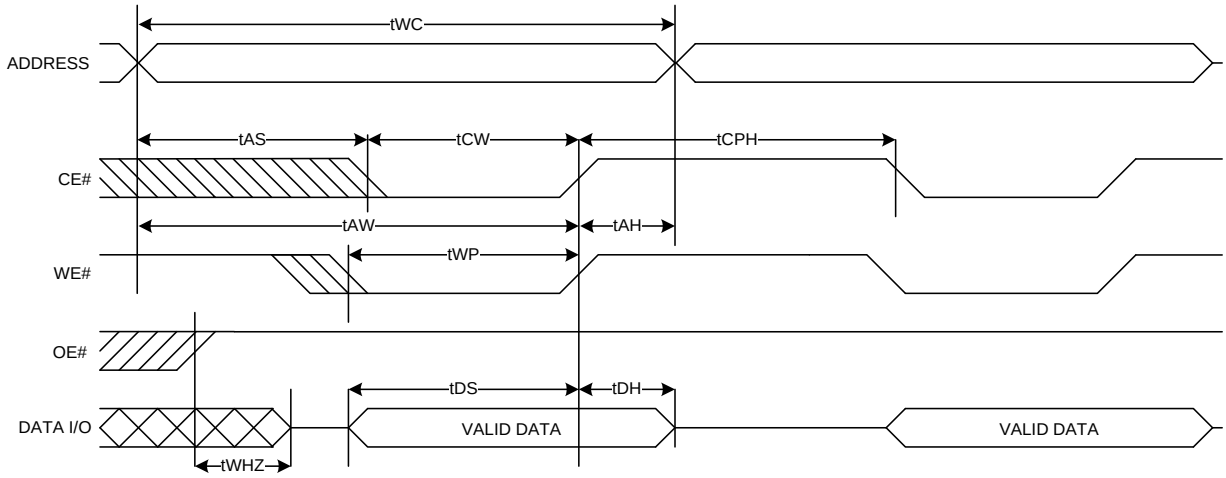


Figure 19. Non-multiplexed Asynchronous SRAM Write Timing (WE# and CE# Controlled)

Write Cycle 1 WE# Controlled, OE# High During Write

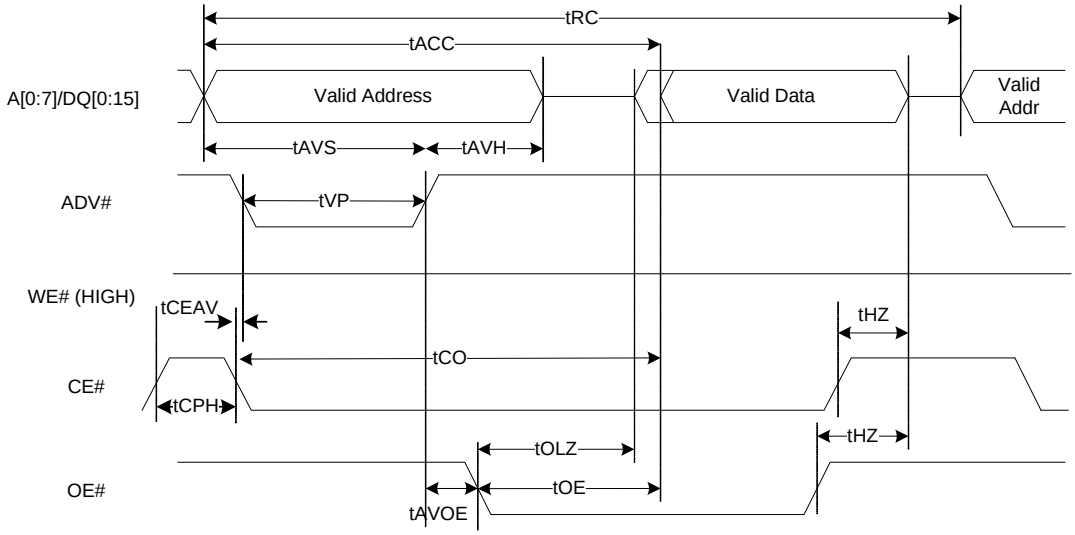


Write Cycle 2 CE# Controlled, OE# High During Write



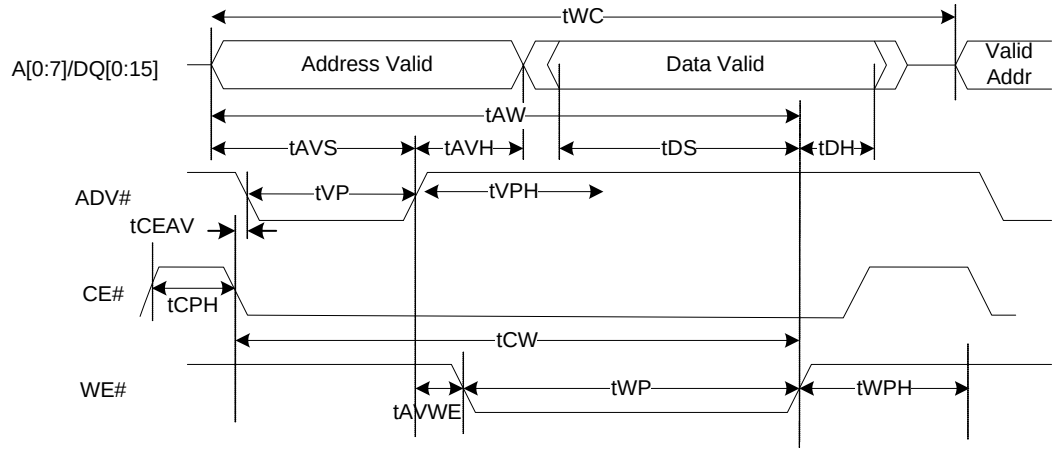
ADMux Timing for Asynchronous Access

Figure 21. ADMux Asynchronous Random Read



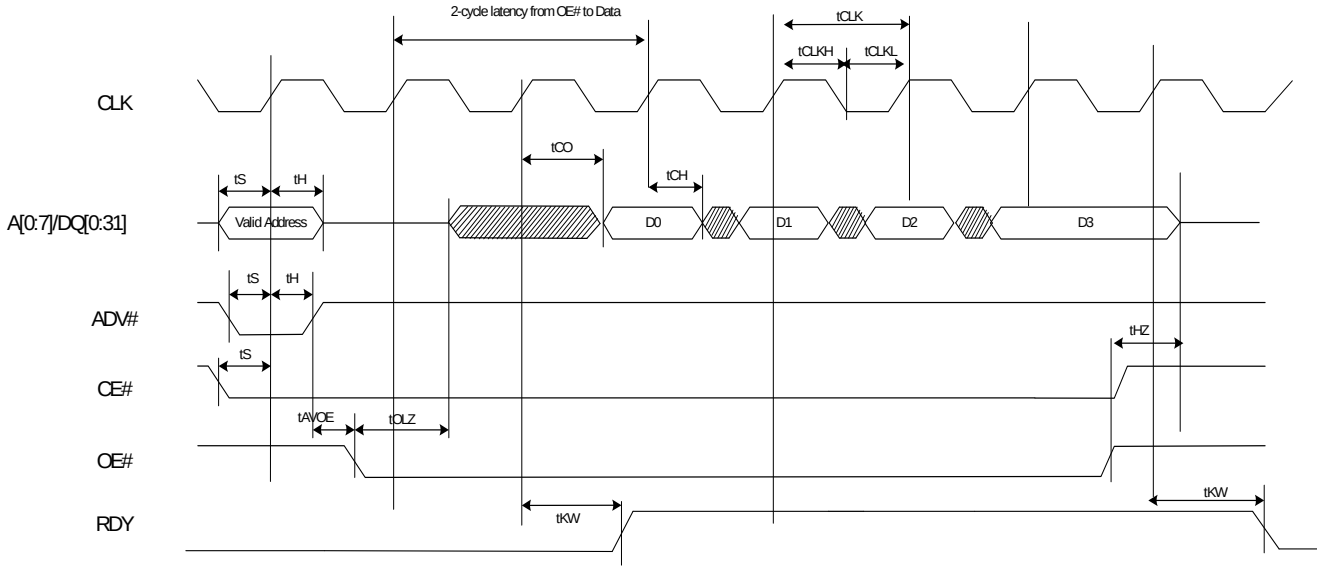
Note:
1. Multiple read cycles can be executed while keeping CE# low.
2. Read operation ends with either de-assertion of either OE# or CE#, whichever comes earlier.

Figure 22. ADMux Asynchronous Random Write



Note:
1. Multiple write cycles can be executed while keeping CE# low.
2. Write operation ends with de-assertion of either WE# or CE#, whichever comes earlier.

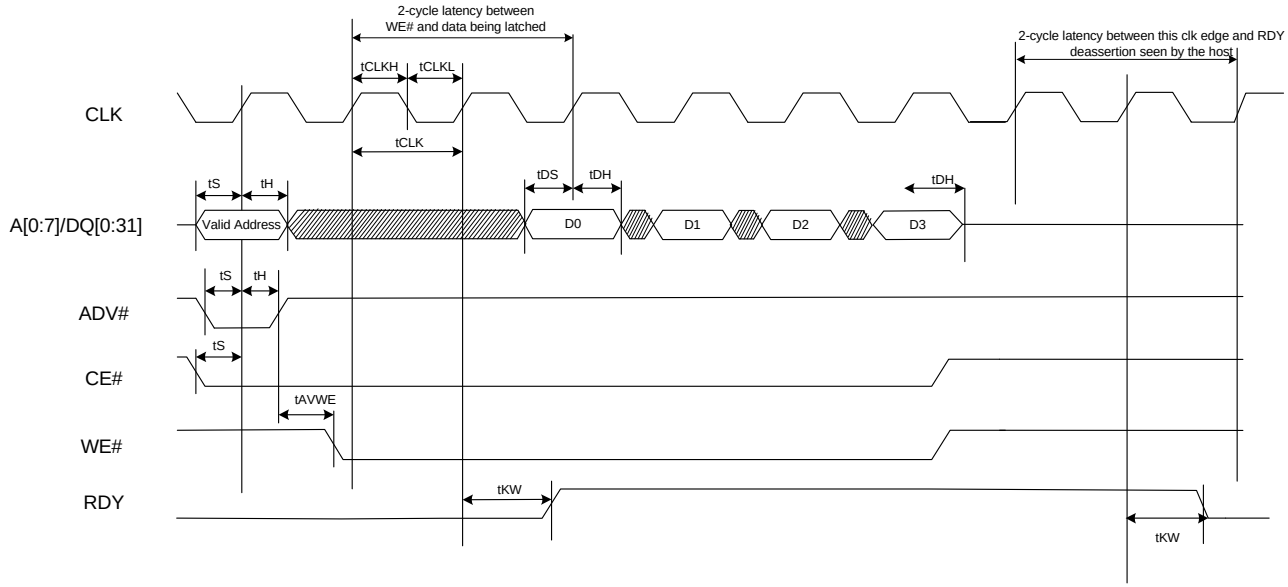
Figure 25. Synchronous ADMux Interface – Burst Read Timing



Note:

1) External P-Port processor and FX3 work operate on the same clock edge
2) External processor sees RDY assert 2 cycles after OE # asserts andand sees RDY deassert a cycle after the last burst data appears on the output
3) Valid output data appears 2 cycle after OE # asserted. The last burst data is held until OE # deasserts
4) Burst size of 4 is shown. Transfer size for the operation must be a multiple of burst size. Burst size is usually power of 2. RDY will not deassert in the middle of the burst.
5) External processor cannot deassert OE in the middle of a burst. If it does so, any bytes remaining in the burst packet could get lost.
6) Two cycle latency is shown for 0-100 MHz operation. Latency can be reduced by 1 cycle for operations at less than 50 MHz (this 1 cycle latency is not supported by the bootloader)

Figure 26. Sync ADMux Interface – Burst Write Timing



Note:

1) External P-Port processor and FX3 operate on the same clock edge
2) External processor sees RDY assert 2 cycles after WE # asserts and deasserts 3 cycles after the edge sampling the last burst data.
3) Transfer size for the operation must be a multiple of burst size. Burst size is usually power of 2. RDY will not deassert in the middle of the burst. Burst size of 4 is shown
4) External processor cannot deassert WE in the middle of a burst. If it does so, any bytes remaining in the burst packet could get lost.
5)Two cycle latency is shown for 0-100 MHz operation. Latency can be reduced by 1 cycle for operations at less than 50 MHz (this 1 cycle latency is not supported by the bootloader)

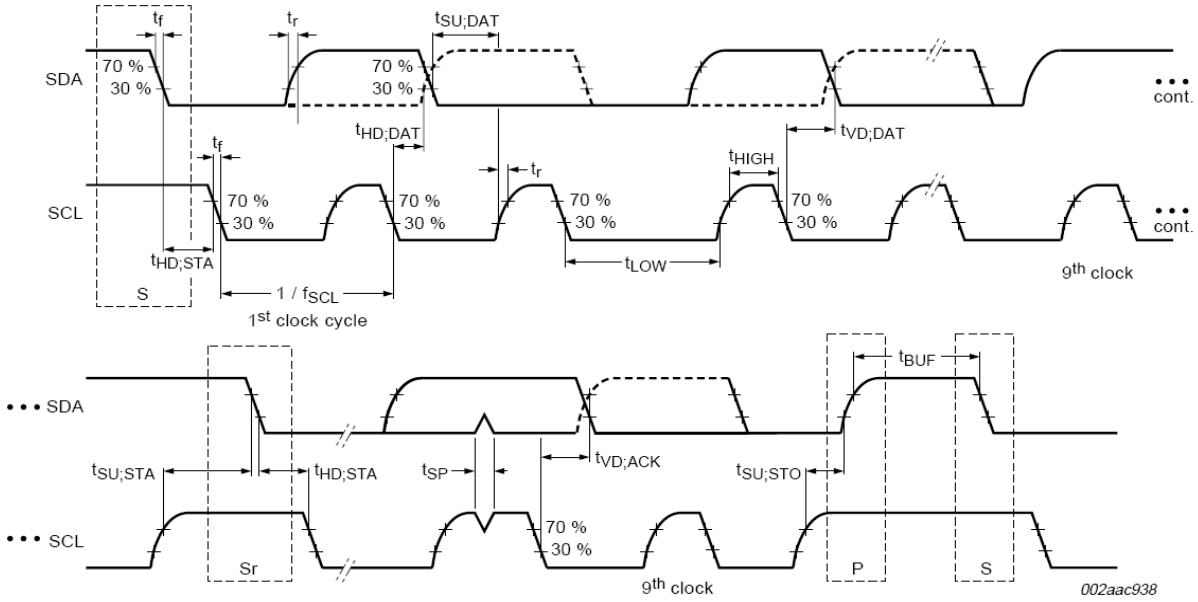
Table 15. Synchronous ADMux Timing Parameters[9]

Parameter	Description	Min	Max	Unit
FREQ	Interface clock frequency	–	100	MHz
tCLK	Clock period	10	–	ns
tCLKH	Clock HIGH time	4	–	ns
tCLKL	Clock LOW time	4	–	ns
tS	CE#/WE#/DQ setup time	2	–	ns
tH	CE#/WE#/DQ hold time	0.5	–	ns
tCH	Clock to data output hold time	0	–	ns
tDS	Data input setup time	2	–	ns
tDH	Clock to data input hold	0.5	–	ns
tAVDOE	ADV# HIGH to OE# LOW	0	–	ns
tAVDWE	ADV# HIGH to WE# LOW	0	–	ns
tHZ	CE# HIGH to Data HIGH-Z	–	8	ns
tOHZ	OE# HIGH to Data HIGH-Z	–	8	ns
tOLZ	OE# LOW to Data LOW-Z	0	–	ns
tKW	Clock to RDY valid	–	8	ns

Serial Peripherals Timing

I^2C Timing

Figure 27. I^2C Timing Definition



Note
9. All parameters guaranteed by design and validated through characterization.

Table 16. I²C Timing Parameters[10] (continued)

Parameter	Description	Min	Max	Units
I²C Fast Mode Plus Parameters (Not supported at I2C_VDDQ=1.2 V)				
fSCL	SCL clock frequency	0	1000	kHz
tHD:STA	Hold time START condition	0.26	–	µs
tLOW	LOW period of the SCL	0.5	–	µs
tHIGH	HIGH period of the SCL	0.26	–	µs
tSU:STA	Setup time for a repeated START condition	0.26	–	µs
tHD:DAT	Data hold time	0	–	µs
tSU:DAT	Data setup time	50	–	ns
tr	Rise time of both SDA and SCL signals	–	120	ns
tf	Fall time of both SDA and SCL signals	–	120	ns
tSU:STO	Setup time for STOP condition	0.26	–	µs
tBUF	Bus-free time between a STOP and START condition	0.5	–	µs
tVD:DAT	Data valid time	–	0.45	µs
tVD:ACK	Data valid ACK	–	0.55	µs
tSP	Pulse width of spikes that must be suppressed by input filter	0	50	ns

I²S Timing Diagram

Figure 28. I²S Transmit Cycle

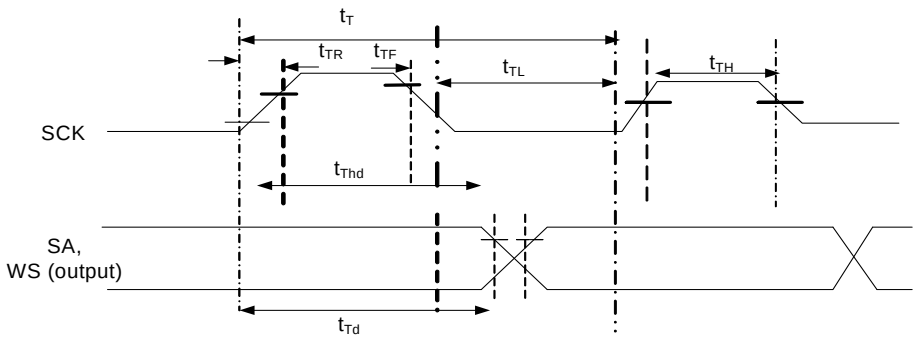


Table 17. I²S Timing Parameters[11]

Parameter	Description	Min	Max	Units
tT	I²S transmitter clock cycle	Ttr	–	ns
tTL	I²S transmitter cycle LOW period	0.35 Ttr	–	ns
tTH	I²S transmitter cycle HIGH period	0.35 Ttr	–	ns
tTR	I²S transmitter rise time	–	0.15 Ttr	ns
tTF	I²S transmitter fall time	–	0.15 Ttr	ns
tThd	I²S transmitter data hold time	0	–	ns
tTd	I²S transmitter delay time	–	0.8tT	ns
Note tT is selectable through clock gears. Max Ttr is designed for 96-kHz codec at 32 bits to be 326 ns (3.072 MHz).				

Note
11. All parameters guaranteed by design and validated through characterization.

Errata

This section describes the errata for Revision C of the FX3. Details include errata trigger conditions, scope of impact, available workaround, and silicon revision applicability. Contact your local Cypress Sales Representative if you have questions.

Part Numbers Affected

Part Number	Device Characteristics
CYUSB301x-xxxx	All Variants
CYUSB201x-xxxx	All Variants

Qualification Status

Product Status: Production

Errata Summary

The following table defines the errata applicability to available Rev. C EZ-USB FX3 SuperSpeed USB Controller family devices.

Items	[Part Number]	Silicon Revision	Fix Status
1. Turning off VIO1 during Normal, Suspend, and Standby modes causes the FX3 to stop working.	CYUSB301x-xxxx CYUSB201x-xxxx	Rev. C, B, ES	Workaround provided
2. USB enumeration failure in USB boot mode when FX3 is self-powered.	CYUSB301x-xxxx CYUSB201x-xxxx	Rev. C, B, ES	Workaround provided
3. Extra ZLP is generated by the COMMIT action in the GPIF II state.	CYUSB301x-xxxx CYUSB201x-xxxx	Rev. C, B, ES	Workaround provided
4. Invalid PID Sequence in USB 2.0 ISOC data transfer.	CYUSB301x-xxxx CYUSB201x-xxxx	Rev. C, B, ES	Workaround provided
5. USB data transfer errors are seen when ZLP is followed by data packet within same microframe.	CYUSB301x-xxxx CYUSB201x-xxxx	Rev. C, B, ES	Workaround provided
6. Bus collision is seen when the I2C block is used as a master in the I2C Multi-master configuration.	CYUSB301x-xxxx CYUSB201x-xxxx	Rev. C, B, ES	Use FX3 in single-master configuration

1. **Turning off VIO1 during Normal, Suspend, and Standby modes causes the FX3 to stop working.**

   ■Problem Definition
   Turning off the VIO1 during Normal, Suspend, and Standby modes will cause the FX3 to stop working.

   ■Parameters Affected
   N/A

   ■Trigger Conditions
   This condition is triggered when the VIO1 is turned off during Normal, Suspend, and Standby modes.

   ■Scope Of Impact
   FX3 stops working.

   ■Workaround
   VIO1 must stay on during Normal, Suspend, and Standby modes.

   ■Fix Status
   No fix. Workaround is required.

2. **USB enumeration failure in USB boot mode when FX3 is self-powered.**

   ■Problem Definition
   FX3 device may not enumerate in USB boot mode when it is self-powered. The bootloader is designed for bus power mode. It does not make use of the VBUS pin on the USB connector to detect the USB connection and expect that USB bus is connected to host if it is powered. If FX3 is not already connected to the USB host when it is powered, then it enters into low-power mode and does not wake up when connected to USB host.

   ■Parameters Affected
   N/A

■**Trigger Conditions**
This condition is triggered when FX3 is self-powered in USB boot mode.

■**Scope Of Impact**
Device does not enumerate

■**Workaround**
Reset the device after connecting to USB host.

■**Fix Status**
No fix. Workaround is required.

3. **Extra ZLP is generated by the COMMIT action in the GPIF II state.**

■**Problem Definition**
When COMMIT action is used in a GPIF-II state without IN_DATA action then an extra Zero Length Packet (ZLP) is committed along with the data packets.

■**Parameters Affected**
N/A

■**Trigger Conditions**
This condition is triggered when COMMIT action is used in a state without IN_DATA action.

■**Scope Of Impact**
Extra ZLP is generated.

■**Workaround**
Use IN_DATA action along with COMMIT action in the same state.

■**Fix Status**
No fix. Workaround is required.

4. **Invalid PID Sequence in USB 2.0 ISOC data transfer.**

■**Problem Definition**
When the FX3 device is functioning as a high speed USB device with high bandwidth isochronous endpoints, the PID sequence of the ISO data packets is governed solely by the isomult setting. The length of the data packet is not considered while generating the PID sequence during each microframe. For example, even if a short packet is being sent on an endpoint with MULT set to 2; the PID used will be DATA2

■**Parameters Affected**
N/A

■**Trigger Conditions**
This condition is triggered when high bandwidth ISOC transfer endpoints are used.

■**Scope Of Impact**
ISOC data transfers failure.

■**Workaround**
This problem can be worked around by reconfiguring the endpoint with a lower isomult setting prior to sending short packets, and then switching back to the original value.

■**Fix Status**
No fix. Workaround is required.

5. **USB data transfer errors are seen when ZLP is followed by data packet within same microframe.**

■**Problem Definition**
Some data transfer errors may be seen if a Zero Length Packet is followed very quickly (within one microframe or 125 us) by another data packet on a burst enabled USB IN endpoint operating at super speed.

■**Parameters Affected**
N/A

■**Trigger Conditions**
This condition is triggered in SuperSpeed transfer with ZLPs

■**Scope Of Impact**
Data failure and lower data speed.

■**Workaround**
The solution is to ensure that some time is allowed to elapse between a ZLP and the next data packet on burst enabled USB IN endpoints. If this cannot be ensured at the data source, the CyU3PDmaChannelSetSuspend() API can be used to suspend the

Document History Page (continued)

Document Title: CYUSB301X/CYUSB201X, EZ-USB® FX3: SuperSpeed USB Controller Document Number: 001-52136				
Revision	**ECN**	**Orig. of Change**	**Submission Date**	**Description of Change**
*E	3204393	OSG	03/24/2011	Updated Slave FIFO protocol and added ZLP signaling protocol Changed GPIFII asynchronous tDO parameter Changed Async Slave FIFO tOE parameter Changed Async Slave FIFO tRDO parameter Added tCOE parameter to GPIFII Sync mode timing parameters Renamed GPIFII Sync mode tDO to tCO and tDO_ss0 to tCO_ss0 Modified description of GPIFII Sync tCO (previously tDO) parameter Changed tAH(address hold time) parameter in Async Slave FIFO modes to be with respect to rising edge of SLWR#/SLRD# instead of falling edge. Correspondingly, changed the tAH number. Removed 24 bit data bus support for GPIFII.
*F	3219493	OSG	04/07/2011	Minor ECN - Release to web. No content changes.
*G	3235250	GSZ	04/20/2011	Minor updates in Features.
*H	3217917	OSG	04/06/2011	Updated GPIFII Synchronous Timing diagram. Added SPI Boot option. Corrected values of R_USB2 and R_USB3. Corrected TCK and TRST# pull-up/pull-down configuration. Minor updates to block diagrams. Corrected Synchronous Slave FIFO tDH parameter.
*I	3305568	DSG	07/07/2011	Minor ECN - Correct ECN number in revision *F. No content changes.
*J	3369042	OSG	12/06/2011	Changed datasheet status from Preliminary to Final. Changed tWRPE parameter to 2ns Updated tRR and tRPW for crystal input Added clarification regarding I_{OZ} and I_{IX} Updated Sync SLave FIFO Read timing diagram Updated SPI timing diagram Removed tGRANULARITY parameter Updated I2S Timing diagram and tTd parameter Updated 121-ball BGA package diagram. Added clarification regarding VCC in DC Specifications table In Power Modes description, stated that VIO1 cannot be turned off at any time if the GPIFII is used in the application Updated Absolute Maximum Ratings Added requirement for by-pass capacitor on $U3RX_{VDDQ}$ and $U3TX_{VDDQ}$ Updated tPEI parameter in Async Slave FIFO timing table Updated Sync Slave FIFO write and read timing diagrams Updated I2C interface tVD:ACK parameter for 1MHz operation Clarified that CTL[15] is not usable as a GPIO
*K	3534275	OSG	02/24/2012	Corrected typo in the block diagram.
*L	3649782	OSG	08/16/2012	Changed part number to CYUSB301X. Added 256 KB range for embedded SRAM. Updated Functional Overview, Other Interfaces, and Clocking sections. Added Pin List for CYUSB3011 and CYUSB3013 parts. Updated Ordering Information with new part numbers.
*M	3848148	OSG	12/20/2012	Updated 121-ball BGA package diagram to current revision.
*N	4016006	OSG	05/31/2013	Updated Features (Added 131-ball WLCSP under Package option). Updated Pin Configurations (Added FX3 131-ball WLCSP Ball Map (Figure 7)). Updated Pin Description (Updated Table 7). Updated Electrical Specifications (Included Commercial Temperature Range related information). Updated Operating Conditions (Included Commercial Temperature Range related information). Updated Package Diagram (Added 131-ball WLCSP Package Diagram (Figure 32)). Updated Ordering Information (Updated part numbers).

Document History Page (continued)

Document Title: CYUSB301X/CYUSB201X, EZ-USB® FX3: SuperSpeed USB Controller Document Number: 001-52136				
Revision	**ECN**	**Orig. of Change**	**Submission Date**	**Description of Change**
*O	4368374	RSKV	05/02/2014	Updated Package Diagram: spec 001-62221 – Changed revision from *B to *C. Updated to new template. Completing Sunset Review.
*P	4474200	ANOP	08/14/2014	Added CYUSB201x MPNs, ball map, and pin list to the datasheet.
*Q	4668496	DBIR	02/24/2015	Updated Features. Updated Logic Block Diagram. Updated Functional Description: Added “For a complete list of related documentation, click here.” at the end. Added More Information. Updated Functional Overview: Updated Application Examples: Updated Figure 1. Updated Figure 2. Updated USB Interface: Updated description. Removed Figure “USB Interface Signals”. Updated Pin Configurations: Updated Figure 6. Updated Reset: Updated Hard Reset: Updated description. Updated Pin Description: Updated Table 7: Updated entire table. Modified CVDDQ power domain description. Removed Table “CYUSB3011 and CYUSB3013 Pin List (GPIF II with 16-bit Data Bus Width)”. Removed Table “CYUSB2014 Pin List (GPIF II with 32-bit Data Bus Width)”. Updated Electrical Specifications: Updated DC Specifications: Added ISS parameter and its details. Updated Slave FIFO Interface: Updated Synchronous Slave FIFO Read Sequence Description: Updated Figure 12. Updated Synchronous Slave FIFO Write Sequence Description: Updated Figure 13. Updated Table 11. Updated AC Timing Parameters: Added Host Processor Interface (P-Port) Timing. Updated Acronyms. Added Errata. Replaced West Bridge Benicia with FX3.
*R	4703347	AMDK	03/27/2015	Updated Slave FIFO Interface: Updated Synchronous Slave FIFO Read Sequence Description: Updated Figure 12. Updated Synchronous Slave FIFO Write Sequence Description: Updated Figure 13. Updated Table 11: Updated minimum value of tSSD parameter. Added tACCD, tFAD parameters and their details.
*S	5160624	AJAI	04/07/2016	Removed ISS parameter. Added item 6 in Errata.