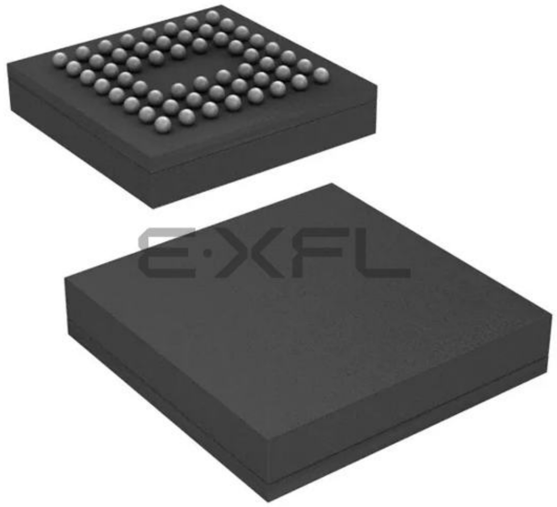
Infineon Technologies - CY7C68053-56BAXI Datasheet



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
Number of I/O	56
Voltage - Supply	1.71V ~ 1.89V
Operating Temperature	-40°C ~ 85°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/cy7c68053-56baxi

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 knowledge base article AN65209 - Getting Started with FX2LP.

- Overview: USB Portfolio, USB Roadmap
- USB 3.0 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 FX3 are:
  - AN65209 - Getting Started with FX2LP
  - AN15652 - Interfacing a Cypress MoBL-USB™ FX2LP18 with an Intel PXA27x Processor
  - AN6076 - Differences between EZ-USB® FX2LP™ and MoBL-USB™ FX2LP18
  - For complete list of Application notes, click here
- Code Examples:
  - USB Hi-Speed
- Technical Reference Manual (TRM):
  - MoBL-USB™ FX2LP18 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

MoBL-USB FX2LP18 Development Kit

The CY3687 MoBL-USB™ FX2LP18 Development Kit is a complete development resource for FX2LP18. It provides a platform to develop and test custom projects using FX2LP18. The development kit contains collateral materials for the firmware, hardware, and software aspects of a design using FX2LP18.

GPIF™ II 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 FX2LP18 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.

Functional Description

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

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

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

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

Applications

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

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

Functional Overview

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

USB Signaling Speed

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

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

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

8051 Microprocessor

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

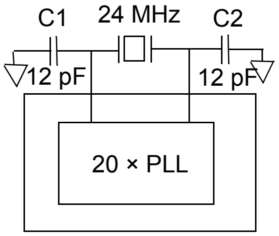
8051 Clock Frequency

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

- Parallel resonant
- Fundamental mode
- 500 μW drive level
- 12 pF (5% tolerance) load capacitors

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

Figure 1. Crystal Configuration



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

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

Special Function Registers

Certain 8051 Special Function Register (SFR) addresses are populated to provide fast access to critical FX2LP18 functions. These SFR additions are shown in Table 1 on page 5. Bold type indicates non standard, enhanced 8051 registers. The two SFR rows that end with '0' and '8' contain bit-addressable registers. The four I/O ports A–D use the SFR addresses used in the standard 8051 for ports 0–3, which are not implemented in FX2LP18. Because of the faster and more efficient SFR addressing, the FX2LP18 I/O ports are not addressable in external RAM space (using the MOVX instruction).

Table 1. Special Function Registers

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

I^{2}C™ Bus

FX2LP18 supports the I^{2}C bus as a master only at 100 or 400 KHz. SCL and SDA pins have open-drain outputs and hysteresis inputs. These signals must be pulled up to either V_{CC} or V_{CC_IO}, even if no I^{2}C device is connected. (Connecting to V_{CC_IO} may be more convenient.)

Buses

This 56-pin package has an 8- or 16-bit 'FIFO' bidirectional data bus, multiplexed on I/O ports B and D.

USB Boot Methods

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

ReNumeration™

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

When first plugged into USB, the FX2LP18 enumerates automatically and downloads firmware and USB descriptor tables over the USB cable. Next, the FX2LP18 enumerates again, this time as a device defined by the downloaded information. This patented two-step process, called ReNumeration™, happens instantly when the device is plugged in, with no hint that the initial download step has occurred.

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

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

Bus-Powered Applications

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

Note

[1] The I^{2}C bus SCL and SDA pins must be pulled up, even if an EEPROM is not connected. Otherwise this detection method does not work properly.

Interrupt System

The FX2LP18 interrupts are described in this section.

INT2 Interrupt Request and Enable Registers

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

USB Interrupt Autovectors

The main USB interrupt is shared by 27 interrupt sources. To save the code and processing time that is normally required to identify the individual USB interrupt source, the FX2LP18 provides a second level of interrupt vectoring, called 'Autovectoring.' When a USB interrupt is asserted, the FX2LP18 pushes the program counter onto its stack then jumps to address 0x0043, where it expects to find a 'jump' instruction to the USB interrupt service routine.

The FX2LP18 jump instruction is encoded, as shown in Table 2.

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

Table 2. INT2 USB Interrupts

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	FX2LP18 ACK'd the control handshake
8	1C	–	Reserved
9	20	EP0-IN	EP0-IN ready to be loaded with data
10	24	EP0-OUT	EP0-OUT has USB data
11	28	EP1-IN	EP1-IN ready to be loaded with data
12	2C	EP1-OUT	EP1-OUT has USB data
13	30	EP2	IN: buffer available. OUT: buffer has data
14	34	EP4	IN: buffer available. OUT: buffer has data
15	38	EP6	IN: buffer available. OUT: buffer has data
16	3C	EP8	IN: buffer available. OUT: buffer has data
17	40	IBN	IN-Bulk-NAK (any IN endpoint)
18	44	–	Reserved
19	48	EP0PING	EP0 OUT was pinged and it NAK'd
20	4C	EP1PING	EP1 OUT was pinged and it NAK'd
21	50	EP2PING	EP2 OUT was pinged and it NAK'd
22	54	EP4PING	EP4 OUT was pinged and it NAK'd
23	58	EP6PING	EP6 OUT was pinged and it NAK'd
24	5C	EP8PING	EP8 OUT was pinged and it NAK'd
25	60	ERRLIMIT	Bus errors exceeded the programmed limit
26	64	–	
27	68	–	Reserved
28	6C	–	Reserved
29	70	EP2ISOERR	ISO EP2 OUT PID sequence error
30	74	EP4ISOERR	ISO EP4 OUT PID sequence error
31	78	EP6ISOERR	ISO EP6 OUT PID sequence error
32	7C	EP8ISOERR	ISO EP8 OUT PID sequence error

Reset and Wakeup

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

Reset Pin

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

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

Table 3. 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 then the 8051 receives a wakeup interrupt. This applies whether or not FX2LP18 is connected to the USB.

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

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

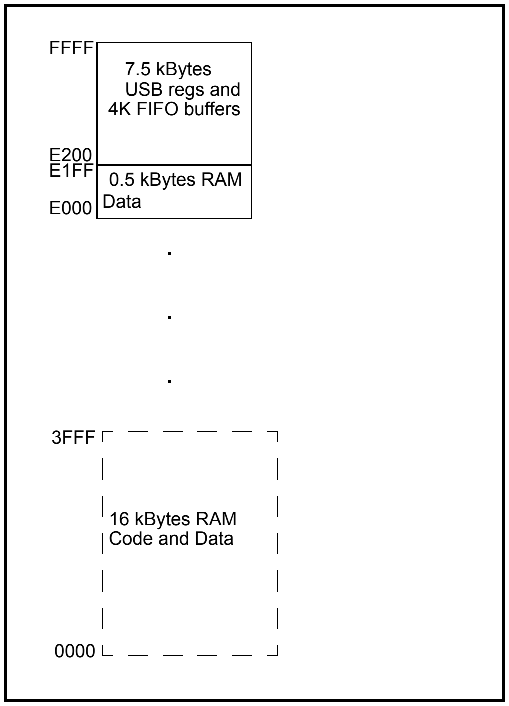
The second wakeup pin, WU2, can also be configured as a general purpose I/O pin. This allows a simple external R-C network to be used as a periodic wakeup source. Note that WAKEUP is active LOW by default.

Lowering Suspend Current

Good design practices for CMOS circuits dictate that any unused input pins must not be floating between V_{IL} and V_{IH}. Floating input pins will not damage the chip, but can substantially increase suspend current. To achieve the lowest suspend current, configure unused port pins as outputs. Connect unused input pins to ground. Some examples of pins that need attention during suspend are:

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

Figure 3. FX2LP18 Internal Code Memory


FFFF
7.5 kBytes USB regs and 4K FIFO buffers
E200
E1FF
0.5 kBytes RAM Data
E000
3FFF
16 kBytes RAM Code and Data
0000

Note

[2] If the external clock is powered at the same time as the CY7C680xx and has a stabilization wait period, it must be added to the 200 µs.

Program/Data RAM

This section describes the FX2LP18 RAM.

Size

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

Memory maps are shown in Figure 3 and Figure 4.

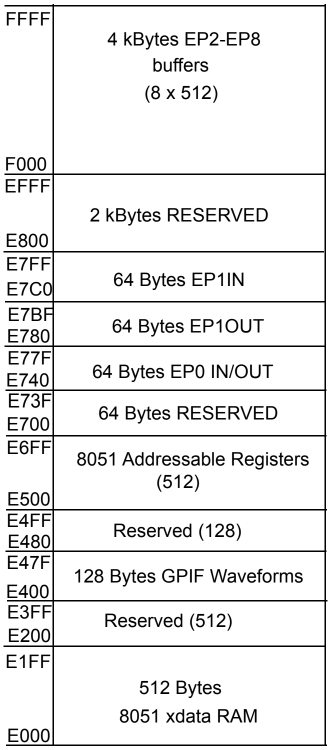
Internal Code Memory

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

- USB download
- USB upload
- Setup data pointer
- I2C interface boot load

Register Addresses

Figure 4. Register Address Memory



Endpoint RAM

This section describes the FX2LP18 Endpoint RAM.

Size

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

Organization

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

Setup Data Buffer

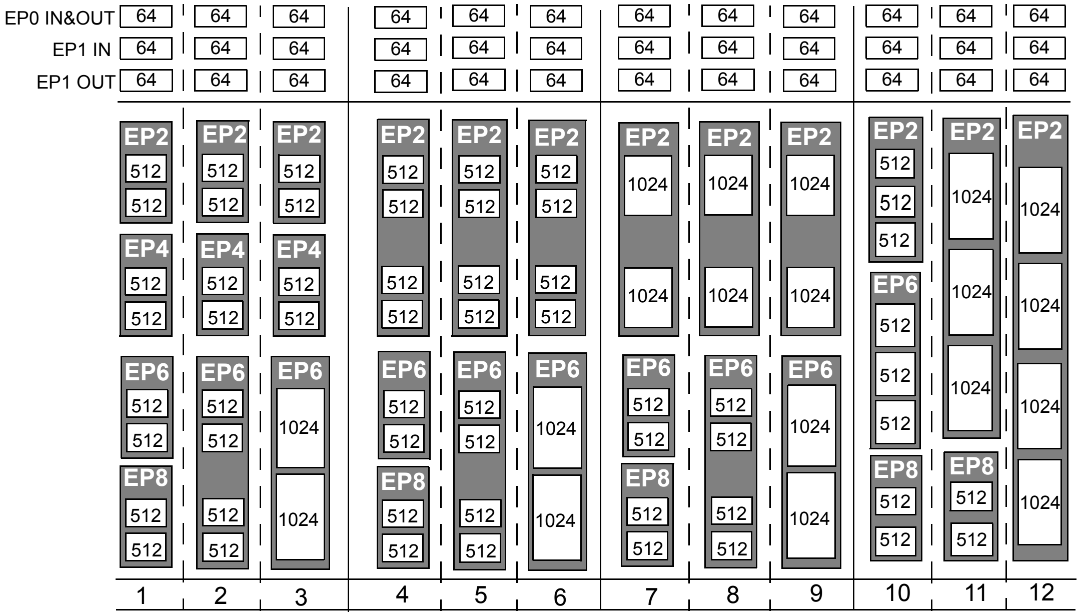
A separate 8-byte buffer at 0xE6B8-0xE6BF holds the setup data from a CONTROL transfer.

Endpoint Configurations (High Speed Mode)

Endpoints 0 and 1 are the same for every configuration. Endpoint 0 is the only CONTROL endpoint, and endpoint 1 can be either BULK or INTERRUPT. The endpoint buffers can be configured in any one of the 12 configurations shown in the vertical columns of Figure 5. When operating in full speed BULK mode only the first 64 bytes of each buffer are used. For example, in high speed the maximum packet size is 512 bytes, but in full speed it is 64 bytes. Even though a buffer is configured to be a 512 byte buffer, in full speed only the first 64 bytes are used. The unused endpoint buffer space is not available for other operations. An example endpoint configuration is:

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

Figure 5. Endpoint Configuration



Default Full Speed Alternate Settings

Table 4. Default Full Speed Alternate Settings[3, 4]

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

Default High Speed Alternate Settings

Table 5. Default High Speed Alternate Settings[3, 4]

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

Notes

[3] ‘0’ means ‘not implemented.’

[4] ‘2×’ means ‘double buffered.’

[5] Even though these buffers are 64 bytes, they are reported as 512 for USB 2.0 compliance. Nnever transfer packets larger than 64 bytes to EP1.

ECC Generation[6]

The MoBL-USB can calculate Error Correcting Codes (ECCs) 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 1-bit error or detect any 2-bit error.

ECC Implementation

The two ECC configurations are selected by the ECCM bit.

ECCM = 0

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

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

ECCM = 1

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

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

USB Uploads and Downloads

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

Autopointer Access

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

I^2C Controller

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

I^2C Port Pins

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

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

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

Table 6. Strap Boot EEPROM Address Lines to these Values

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

I^2C Interface Boot Load Access

At power on reset the I^2C interface boot loader loads the VID/PID/DID and configuration bytes and up to 16 kBytes of program/data. The available RAM spaces are 16 kBytes from 0x0000–0x3FFF and 512 bytes from 0xE000–0xE1FF. The 8051 is reset. I^2C interface boot loads only occur after power on reset.

I^2C Interface General Purpose Access

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

Notes

[6] To use the ECC logic, the GPIF or Slave FIFO interface must be configured for byte-wide operation.

[7] After the data is downloaded from the host, a ‘loader’ can execute from internal RAM in order to transfer downloaded data to external memory.

[8] This EEPROM does not have address pins.

CY7C68053 Pin Descriptions

Table 7. FX2LP18 Pin Descriptions[9]

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

Note

[9] Do not leave unused inputs floating. Tie either HIGH or LOW as appropriate. Only pull outputs up or down to ensure signals at power up and in standby. Do not drive any pins while the device is powered down.

Table 7. **FX2LP18 Pin Descriptions**[9] (continued)

56 VFBGA	Name	Type	Default	Description
6F	PA4 or FIFOADR0	I/O/Z	I (PA4)	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **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].
8C	PA5 or FIFOADR1	I/O/Z	I (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].
7C	PA6 or PKTEND	I/O/Z	I (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 that commits the FIFO packet data to the endpoint and whose polarity is programmable using FIFOPINPOLAR.5.
6C	PA7 or FLAGD or SLCS#	I/O/Z	I (PA7)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and PORTACFG.7 bits. **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				
3H	PB0 or FD[0]	I/O/Z	I (PB0)	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **PB0** is a bidirectional I/O port pin. **FD[0]** is the bidirectional FIFO/GPIF data bus.
4F	PB1 or FD[1]	I/O/Z	I (PB1)	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **PB1** is a bidirectional I/O port pin. **FD[1]** is the bidirectional FIFO/GPIF data bus.
4H	PB2 or FD[2]	I/O/Z	I (PB2)	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **PB2** is a bidirectional I/O port pin. **FD[2]** is the bidirectional FIFO/GPIF data bus.
4G	PB3 or FD[3]	I/O/Z	I (PB3)	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **PB3** is a bidirectional I/O port pin. **FD[3]** is the bidirectional FIFO/GPIF data bus.
5H	PB4 or FD[4]	I/O/Z	I (PB4)	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **PB4** is a bidirectional I/O port pin. **FD[4]** is the bidirectional FIFO/GPIF data bus.
5G	PB5 or FD[5]	I/O/Z	I (PB5)	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **PB5** is a bidirectional I/O port pin. **FD[5]** is the bidirectional FIFO/GPIF data bus.
5F	PB6 or FD[6]	I/O/Z	I (PB6)	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **PB6** is a bidirectional I/O port pin. **FD[6]** is the bidirectional FIFO/GPIF data bus.
6H	PB7 or FD[7]	I/O/Z	I (PB7)	Multiplexed pin whose function is selected IFCONFIG[1:0]. **PB7** is a bidirectional I/O port pin. **FD[7]** is the bidirectional FIFO/GPIF data bus.

Table 8. FX2LP18 Register Summary (continued)

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access

Note

[10] Read and writes to these registers may require synchronization delay, see *MoBL-USB FX2LP18 Technical Reference Manual* for ‘Synchronization Delay.’

Table 8. FX2LP18 Register Summary (continued)

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
		Reserved											
E6D2	1	EP2GPIFFLGSEL[10]	Endpoint 2 GPIF flag select	0	0	0	0	0	0	FS1	FS0	00000000	RW
E6D3	1	EP2GPIFPFSTOP	Endpoint 2 GPIF stop transaction on program flag	0	0	0	0	0	0	0	FIFO2FLAG	00000000	RW
E6D4	1	EP2GPIFTRIG[10]	Endpoint 2 GPIF trigger	x	x	x	x	x	x	x	x	xxxxxxxx	W
	3	Reserved											
		Reserved											
		Reserved											
E6DA	1	EP4GPIFFLGSEL[10]	Endpoint 4 GPIF flag select	0	0	0	0	0	0	FS1	FS0	00000000	RW
E6DB	1	EP4GPIFPFSTOP	Endpoint 4 GPIF stop transaction on GPIF flag	0	0	0	0	0	0	0	FIFO4FLAG	00000000	RW
E6DC	1	EP4GPIFTRIG[10]	Endpoint 4 GPIF trigger	x	x	x	x	x	x	x	x	xxxxxxxx	W
	3	Reserved											
		Reserved											
		Reserved											
E6E2	1	EP6GPIFFLGSEL[10]	Endpoint 6 GPIF flag select	0	0	0	0	0	0	FS1	FS0	00000000	RW
E6E3	1	EP6GPIFPFSTOP	Endpoint 6 GPIF stop transaction on program flag	0	0	0	0	0	0	0	FIFO6FLAG	00000000	RW
E6E4	1	EP6GPIFTRIG[10]	Endpoint 6 GPIF trigger	x	x	x	x	x	x	x	x	xxxxxxxx	W
	3	Reserved											
		Reserved											
		Reserved											
E6EA	1	EP8GPIFFLGSEL[10]	Endpoint 8 GPIF flag select	0	0	0	0	0	0	FS1	FS0	00000000	RW
E6EB	1	EP8GPIFPFSTOP	Endpoint 8 GPIF stop transaction on program flag	0	0	0	0	0	0	0	FIFO8FLAG	00000000	RW
E6EC	1	EP8GPIFTRIG[10]	Endpoint 8 GPIF trigger	x	x	x	x	x	x	x	x	xxxxxxxx	W
	3	Reserved											
E6F0	1	XGPIFSGLDATH	GPIF Data H (16-bit mode only)	D15	D14	D13	D12	D11	D10	D9	D8	xxxxxxxx	RW
E6F1	1	XGPIFSGLDATLX	Read/Write GPIF Data L and trigger transaction	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
E6F2	1	XGPIFSGLDATL-NOX	Read GPIF Data L, no transaction trigger	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	R
E6F3	1	GPIFREADYCFG	Internal RDY, sync/async, RDY pin states	INTRDY	SAS	TCXRDY5	0	0	0	0	0	00000000	bbbrrrrr
E6F4	1	GPIFREADYSTAT	GPIF ready status	0	0	0	0	0	0	RDY1	RDY0	00xxxxxx	R
E6F5	1	GPIFABORT	Abort GPIF waveforms	x	x	x	x	x	x	x	x	xxxxxxxx	W
E6F6	2	Reserved											
		ENDPOINT BUFFERS											
E740	64	EP0BUF	EP0-IN/-OUT buffer	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
E780	64	EP10UTBUF	EP1-OUT buffer	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
E7C0	64	EP1INBUF	EP1-IN buffer	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
E800	2048	Reserved											RW
F000	1024	EP2FIFOBUF	512/1024-byte EP 2/Slave FIFO buffer (IN or OUT)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
F400	512	EP4FIFOBUF	512 byte EP 4/Slave FIFO buffer (IN or OUT)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
F600	512	Reserved											
F800	1024	EP6FIFOBUF	512/1024-byte EP 6/Slave FIFO buffer (IN or OUT)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
FC00	512	EP8FIFOBUF	512 byte EP 8/Slave FIFO buffer (IN or OUT)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
FE00	512	Reserved											
xxxx		I^{2}C Configuration Byte		0	DISCON	0	0	0	0	0	400KHZ	xxxxxxxx [12]	n/a

Note

[12] If no EEPROM is detected by the SIE then the default is 00000000.

Table 10. GPIF Synchronous Signals Parameters with Externally Sourced IFCLK[18]

Parameter	Description	Min	Max	Unit
t_{IFCLK}	IFCLK period[19]	20.83	200	ns
t_{SRY}	RDY_X to clock setup time	2.9	–	ns
t_{RYH}	Clock to RDY_X	3.7	–	ns
t_{SGD}	GPIF data to clock setup time	3.2	–	ns
t_{DAH}	GPIF data hold time	4.5	–	ns
t_{XGD}	Clock to GPIF data output propagation delay	–	15	ns
t_{XCTL}	Clock to CTL_X output propagation delay	–	13.06	ns

Slave FIFO Synchronous Read

Figure 9. Slave FIFO Synchronous Read Timing Diagram[20]

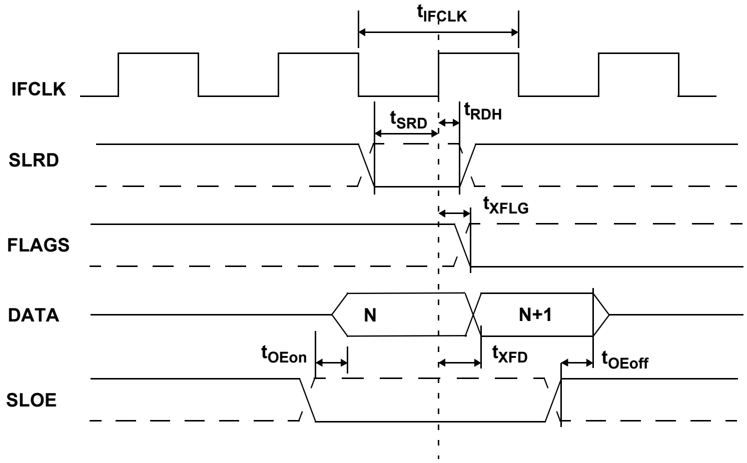


Table 11. Slave FIFO Synchronous Read Parameters with Internally Sourced IFCLK[18]

Parameter	Description	Min	Max	Unit
t_{IFCLK}	IFCLK period	20.83	–	ns
t_{SRD}	SLRD to clock setup time	18.7	–	ns
t_{RDH}	Clock to SLRD hold time	0	–	ns
t_{OEon}	SLOE turn-on to FIFO data valid	–	10.5	ns
t_{OEoff}	SLOE turn-off to FIFO data hold	2.15	10.5	ns
t_{XFLG}	Clock to FLAGS output propagation delay	–	9.5	ns
t_{XFD}	Clock to FIFO data output propagation delay	–	11	ns

Notes

[18] GPIF asynchronous RDY_x signals have a minimum setup time of 50 ns when using internal 48 MHz IFCLK.

[19] IFCLK must not exceed 48 MHz.

[20] Dashed lines denote signals with programmable polarity.

Slave FIFO Synchronous Write

Figure 11. Slave FIFO Synchronous Write Timing Diagram[24]

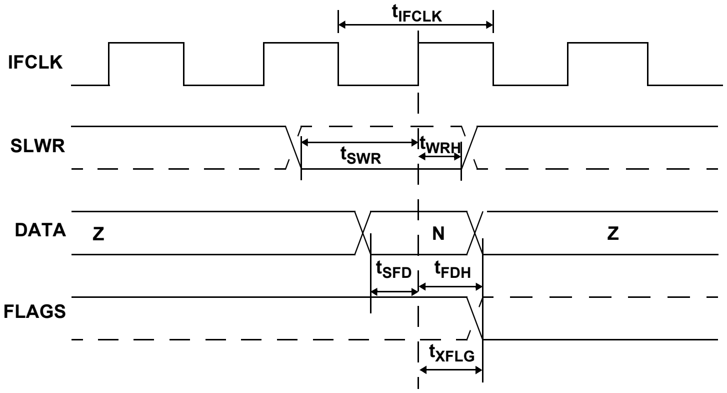


Table 14. Slave FIFO Synchronous Write Parameters with Internally Sourced IFCLK[25]

Parameter	Description	Min	Max	Unit
t_{IFCLK}	IFCLK period	20.83	–	ns
t_{SWR}	SLWR to clock setup time	18.1	–	ns
t_{WRH}	Clock to SLWR hold time	0	–	ns
t_{SFD}	FIFO data to clock setup time	10.64	–	ns
t_{FDH}	Clock to FIFO data hold time	0	–	ns
t_{XFLG}	Clock to FLAGS output propagation time	–	9.5	ns

Table 15. Slave FIFO Synchronous Write Parameters with Externally Sourced IFCLK[26]

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

Notes

[24] Dashed lines denote signals with programmable polarity.

[25] GPIF asynchronous RDY_x signals have a minimum setup time of 50 ns when using internal 48 MHz IFCLK.

[26] Read and writes to these registers may require synchronization delay, see *MoBL-USB FX2LP18 Technical Reference Manual* for 'Synchronization Delay.'

Slave FIFO Synchronous Packet End Strobe

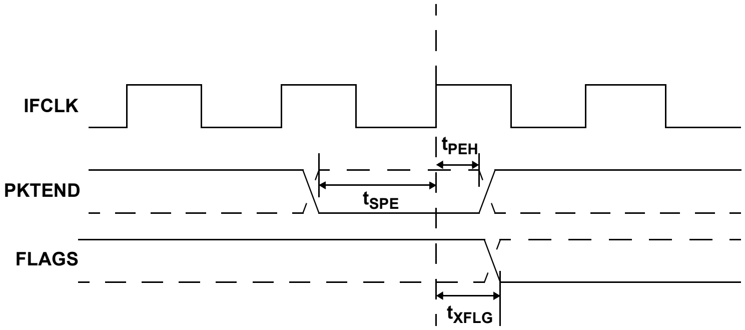


Figure 13. Slave FIFO Synchronous Packet End Strobe Timing Diagram[29]

Table 17. Slave FIFO Synchronous Packet End Strobe Parameters with Internally Sourced IFCLK[30]

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 18. Slave FIFO Synchronous Packet End Strobe Parameters with Externally Sourced IFCLK[30]

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	3.04	–	ns
t_{XFLG}	Clock to FLAGS output propagation delay	–	13.5	ns

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

Although there are no specific timing requirements for the PKTEND assertion, there is a specific corner case condition that needs attention while using the PKTEND to commit a one byte/word packet. There is an additional timing requirement to be met when the FIFO is configured to operate in auto mode and you want to send two packets back to back: a full packet (full defined as the number of bytes in the FIFO meeting the level set in AUTOINLEN register) committed automatically followed by a short one byte/word packet committed manually using the PKTEND pin. In this scenario, make sure to assert PKTEND at least one clock cycle after the rising edge that caused the last byte/word to be clocked into the previous auto committed packet. Figure 14 shows this scenario. X is the value the AUTOINLEN register is set to when the IN endpoint is configured to be in auto mode.

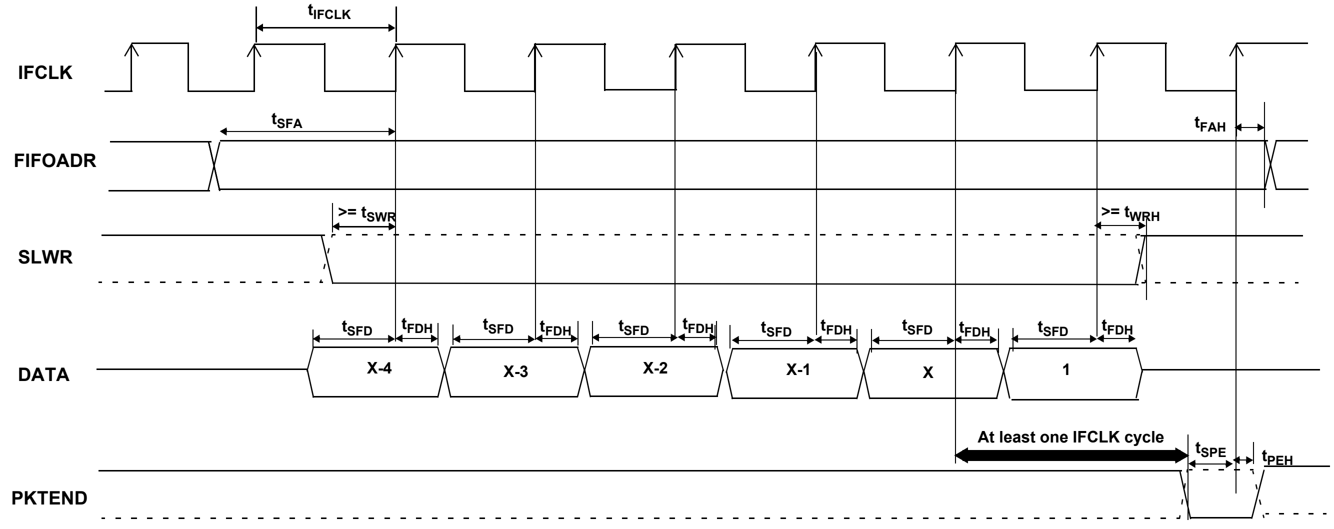
Figure 14 shows a scenario where two packets are committed. The first packet is committed automatically when the number of bytes in the FIFO reaches X (value set in AUTOINLEN register) and the second one byte/word short packet is committed manually using PKTEND. Note that there is at least one IFCLK cycle timing between the assertion of PKTEND and clocking of the last byte of the previous packet (causing the packet to be committed automatically). Failing to adhere to this timing, results in the FX2LP18 failing to send the one byte/word short packet.

Notes

[29] Dashed lines denote signals with programmable polarity.

[30] Read and writes to these registers may require synchronization delay, see *MoBL-USB FX2LP18 Technical Reference Manual* for ‘Synchronization Delay.’

Figure 14. Slave FIFO Synchronous Write Sequence and Timing Diagram[31]



Slave FIFO Asynchronous Packet End Strobe

Figure 15. Slave FIFO Asynchronous Packet End Strobe Timing Diagram[31]

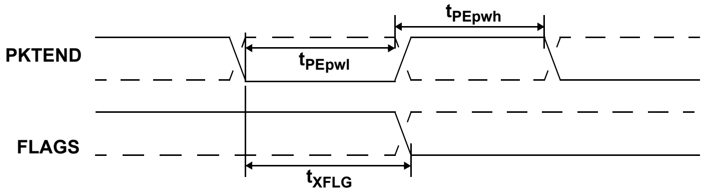


Table 19. Slave FIFO Asynchronous Packet End Strobe Parameters[32]

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 16. Slave FIFO Output Enable Timing Diagram[33]

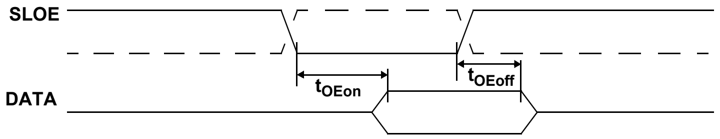


Table 20. Slave FIFO Output Enable Parameters

Parameter	Description	Min	Max	Unit
t_{OEon}	SLOE assert to FIFO data output	–	10.5	ns
t_{OEoff}	SLOE deassert to FIFO data hold	2.15	10.5	ns

Notes

[31] Dashed lines denote signals with programmable polarity.
[32] Slave FIFO asynchronous parameter values use internal IFCLK setting at 48 MHz.

Although there are no specific timing requirements for the PKTEND assertion, there is a specific corner case condition that needs attention while using the PKTEND to commit a one byte/word packet. Additional timing requirements exist when the FIFO is configured to operate in auto mode and you want to send two packets: a full packet (full defined as the number of bytes in the FIFO meeting the level set in AUTOINLEN register) committed automatically followed by a short one byte/word packet committed manually using the PKTEND pin. In this case, the external master must make sure to assert the PKTEND pin at least one clock cycle after the rising edge that caused the last byte/word to be clocked into the previous auto committed packet (the packet with the number of bytes equal to what is set in the AUTOINLEN register). Refer to Figure 14 on page 34 for further details about this timing.

Sequence Diagram of a Single and Burst Asynchronous Read

Figure 23. Slave FIFO Asynchronous Read Sequence and Timing Diagram[38]

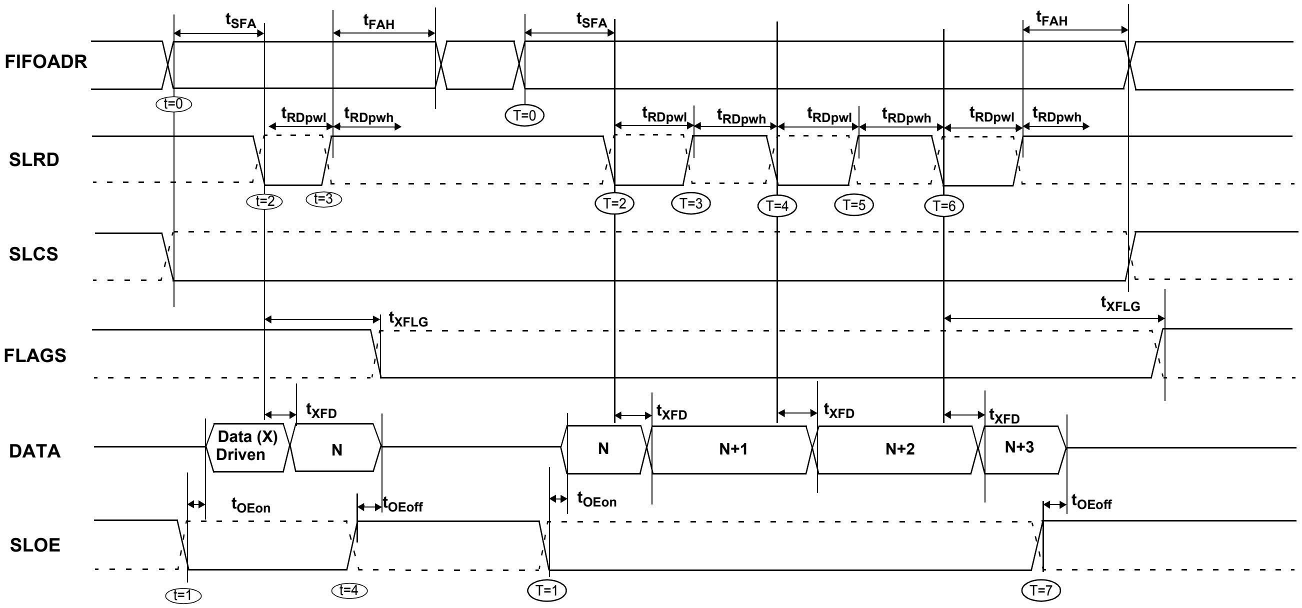


Figure 24. Slave FIFO Asynchronous Read Sequence of Events Diagram

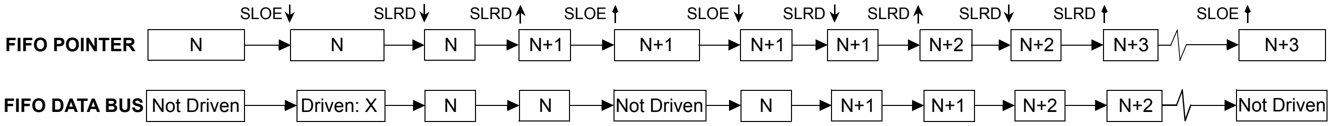


Figure 23 illustrates the timing relationship of the SLAVE FIFO signals during an asynchronous FIFO read. It shows a single read followed by a burst read.

- At t = 0, the FIFO address is stable and the SLCS signal is asserted.
- At t = 1, SLOE is asserted. This results in the data bus being driven. The data that is driven on to the bus is previous data; it is data that was in the FIFO from a prior read cycle.
- At t = 2, SLRD is asserted. The SLRD must meet the minimum active pulse of t_{RDpwl} and minimum inactive pulse width of t_{RDpwh}. If SLCS is used then, SLCS must be asserted before SLRD is asserted (that is, 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 23, data N is the first valid data read from the FIFO. For data to appear on the data bus during the read cycle (for example, SLRD is asserted), SLOE MUST be in an asserted state. SLRD and SLOE can also be tied together.

The same sequence of events is also shown for a burst read marked with T = 0 through 5.

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

Note

[38] Dashed lines denote signals with programmable polarity.

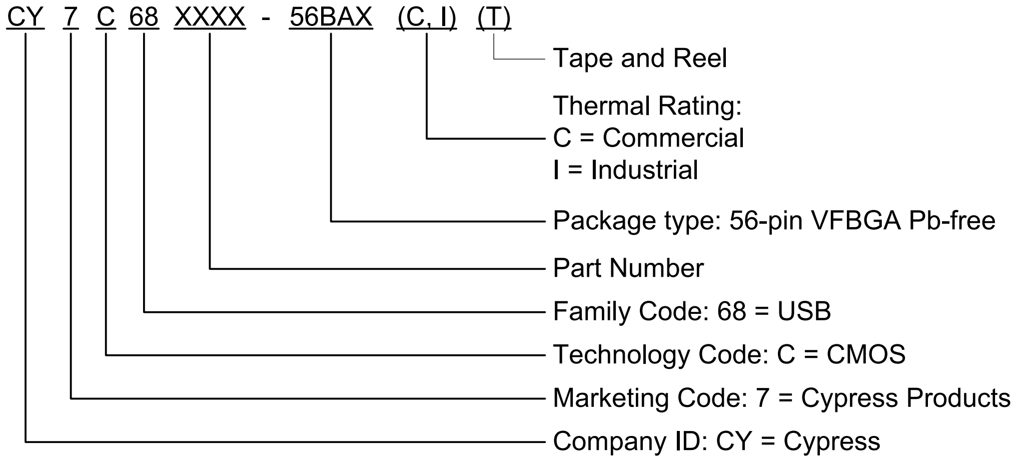
Ordering Information

Table 23 lists the key package features and ordering codes. The table contains only the parts that are currently available. If you do not see what you are looking for, contact your local sales representative. For more information, visit the Cypress website at www.cypress.com and refer to the product summary page at http://www.cypress.com/products.

Table 23. Key Features and Ordering Information

Ordering Code	Package Type	RAM Size	Number of Prog I/Os	8051 Address/Data Buses
CY7C68053-56BAXI	56-ball VFBGA (Pb-free)	16 K	24	–
Development Tool Kit				
CY3687	MoBL-USB FX2LP18 Development Kit			

Ordering Code Definitions


CY 7 C 68 XXXX - 56BAX (C, I) (T)
Tape and Reel
Thermal Rating:
C = Commercial
I = Industrial
Package type: 56-pin VFBGA Pb-free
Part Number
Family Code: 68 = USB
Technology Code: C = CMOS
Marketing Code: 7 = Cypress Products
Company ID: CY = Cypress

PCB Layout Recommendations

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

- At least a four-layer impedance controlled board is required to maintain signal quality.
- Specify impedance targets (ask your board vendor what they can achieve).
- To control impedance, maintain trace widths and trace spacing to within specifications.
- Minimize stubs to minimize reflected signals.
- Connections between the USB connector shell and signal ground must be done near the USB connector.
- Bypass or flyback caps on VBus, near connector, are recommended.
- DPLUS and DMINUS trace lengths must be kept within 2 mm of each other in length, with preferred length of 20 to 30 mm.
- Maintain a solid ground plane under the DPLUS and DMINUS traces. Do not allow the plane to be split under these traces.
- It is preferable to have no vias placed on the DPLUS or DMINUS trace routing.
- Isolate the DPLUS and DMINUS traces from all other signal traces by no less than 10 mm.

Acronyms

Table 24. Acronyms Used in this Document

Acronym	Description
ATA	Advanced Technology Attachment
ASIC	Application Specific Integrated Circuit
CPU	Central Processing Unit
DID	Device Identifier
DSP	Digital Signal Processor
ECC	Error Correction Code
EEPROM	Electrically Erasable Programmable Read Only Memory
EPP	Enhanced Parallel Port
FIFO	First In First Out
GPIF	General Programmable Interface
GPIO	General Purpose Input/Output
I/O	Input/Output
I2C	Inter-Integrated Circuit
PDA	Personal Digital Assistant
PLL	Phase-Locked Loop
PID	Product Identifier
RAM	Random Access Memory
SIE	Serial Interface Engine
SOF	Start Of Frame
USB	Universal Serial Bus
VID	Vendor Identifier
VFBGA	Very Fine-Pitch Ball Grid Array
UTOPIA	Universal Test and Operations Physical-Layer

Document Conventions

Units of Measure

Table 25. Units of Measure

Symbol	Unit of Measure
KHz	kilohertz
Mbytes	megabyte
MHz	megahertz
µA	microampere
µs	microsecond
µW	microwatt
mA	milliampere
mW	milliwatt
ns	nanosecond
ppm	parts per million
pF	picofarad
V	volt