EZ-USB NX2LP-Flex™ Flexible USB NAND Flash Controller

1.0 Silicon Features

Both CY7C68033/CY7C68034:

- Certified compliant for Bus- or Self-powered USB 2.0 operation (TID# 40490118)
- Single-chip, integrated USB 2.0 transceiver and smart SIE
- Ultra low power – 43 mA typical current draw in any mode
- Enhanced 8051 core
  - Firmware runs from internal RAM, which is downloaded from NAND flash at startup
  - No external EEPROM required
- 15 KBytes of on-chip Code/Data RAM
  - Default NAND firmware ~8 kB
  - Default free space ~7 kB
- Four programmable BULK/INTERRUPT/ISOCHRONOUS endpoints
  - Buffering options: double, triple, and quad
- Additional programmable (BULK/INTERRUPT) 64-byte endpoint
- SmartMedia Standard Hardware ECC generation with 1-bit correction and 2-bit detection
- GPIF (General Programmable Interface)
  - Allows direct connection to most parallel interfaces
  - Programmable waveform descriptors and configuration registers to define waveforms
  - Supports multiple Ready (RDY) inputs and Control (CTL) outputs
- 12 fully-programmable GPIO pins
- Integrated, industry-standard enhanced 8051
  - 48-MHz, 24-MHz, or 12-MHz CPU operation
  - Four clocks per instruction cycle
  - Three counter/timers
  - Expanded interrupt system
  - Two data pointers
- 3.3V operation with 5V tolerant inputs
- Vectored USB interrupts and GPIF/FIFO interrupts
- Separate data buffers for the Set-up and Data portions of a CONTROL transfer
- Integrated I^2C controller, runs at 100 or 400 kHz
- Four integrated FIFOs
  - Integrated glue logic and FIFOs lower system cost
  - Automatic conversion to and from 16-bit buses
  - Master or slave operation
  - Uses external clock or asynchronous strobes
  - Easy interface to ASIC and DSP ICs
- Available in space saving, 56-pin QFN package

CY7C68034:

- Ideal for battery powered applications
  - Suspend current: 100 μA (typ.)

CY7C68033:

- Ideal for non-battery powered applications
  - Suspend current: 300 μA (typ.)

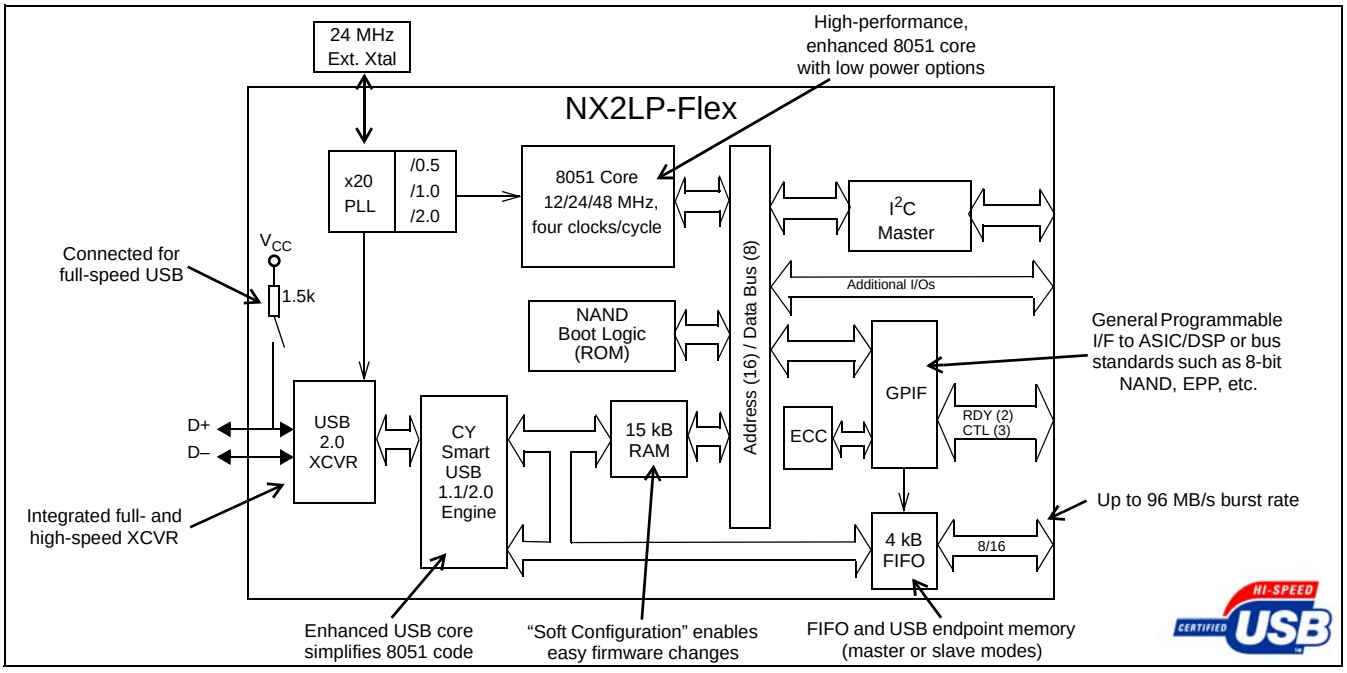


Figure 1-1. Block Diagram

1.1 Default NAND Firmware Features

Because the NX2LP-Flex™ is intended for NAND Flash-based USB mass storage applications, a default firmware image is included in the development kit with the following features:

- High (480-Mbps) or full (12-Mbps) speed USB support
- Both common NAND page sizes supported
  - 512 bytes for up to 1 Gb capacity
  - 2K bytes for up to 8 Gb capacity
- 12 configurable general-purpose I/O (GPIO) pins
  - 2 dedicated chip enable (CE#) pins
  - 6 configurable CE#/GPIO pins
    - Up to 8 NAND Flash single-device (single-die) chips are supported
    - Up to 4 NAND Flash dual-device (dual-die) chips are supported
    - Compile option allows unused CE# pins to be configured as GPIOs
  - 4 dedicated GPIO pins
- Industry standard ECC NAND Flash correction
  - 1-bit per 256-bit correction
  - 2-bit error detection
- Industry standard (SmartMedia) page management for wear leveling algorithm, bad block handling, and Physical to Logical management.
- 8-bit NAND Flash interface support
- Support for 30-ns, 50-ns, and 100-ns NAND Flash timing
- Complies with the USB Mass Storage Class Specification revision 1.0

The default firmware image implements a USB 2.0 NAND Flash controller. This controller adheres to the *Mass Storage Class Bulk-Only Transport Specification*. The USB port of the NX2LP-Flex is connected to a host computer directly or via the downstream port of a USB hub. Host software issues commands and data to the NX2LP-Flex and receives status and data from the NX2LP-Flex using standard USB protocol.

The default firmware image supports industry leading 8-bit NAND Flash interfaces and both common NAND page sizes of 512 and 2k bytes. Up to eight chip enable pins allow the NX2LP-Flex to be connected to up to eight single- or four dual-die NAND Flash chips.

Complete source code and documentation for the default firmware image are included in the NX2LP-Flex development kit to enable customization for meeting design requirements. Additionally, compile options for the default firmware allow for quick configuration of some features to decrease design effort and increase time-to-market advantages.

2.0 Overview

Cypress Semiconductor Corporation's (Cypress's) EZ-USB NX2LP-Flex (CY7C68033/CY7C68034) is a firmware-based, programmable version of the EZ-USB NX2LP™ (CY7C68023/CY7C68024), which is a fixed-function, low-power USB 2.0 NAND Flash controller. 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 enables feature-rich NAND Flash-based applications.

The ingenious architecture of NX2LP-Flex results in USB 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 small 56-pin QFN package. Because it incorporates the USB 2.0 transceiver, the NX2LP-Flex is more economical, providing a smaller footprint solution than external USB 2.0 SIE or transceiver implementations. With EZ-USB NX2LP-Flex, the Cypress Smart SIE handles most of the USB 1.1 and 2.0 protocol, freeing the embedded microcontroller for application-specific functions and decreasing development time while ensuring 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 UTOPIA, EPP, I^2C, PCMCIA, and most DSP processors.

3.0 Applications

The NX2LP-Flex allows designers to add extra functionality to basic NAND Flash mass storage designs, or to interface them with other peripheral devices. Applications may include:

- NAND Flash-based GPS devices
- NAND Flash-based DVB video capture devices
- Wireless pointer/presenter tools with NAND Flash storage
- NAND Flash-based MPEG/TV conversion devices
- Legacy conversion devices with NAND Flash storage
- NAND Flash-based cameras
- NAND Flash mass storage device with biometric (e.g., fingerprint) security
- Home PNA devices with NAND Flash storage
- Wireless LAN with NAND Flash storage
- NAND Flash-based MP3 players
- LAN networking with NAND Flash storage

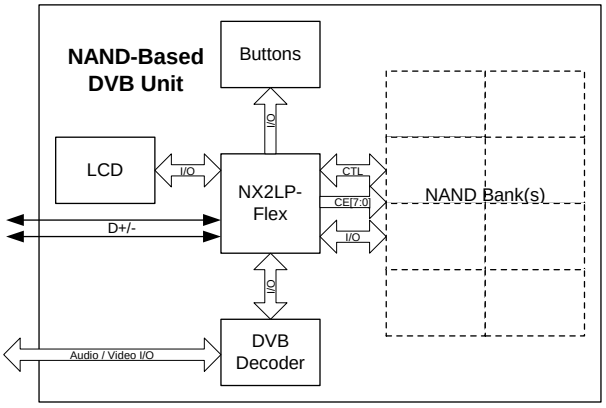


Figure 3-1. Example DVB Block Diagram

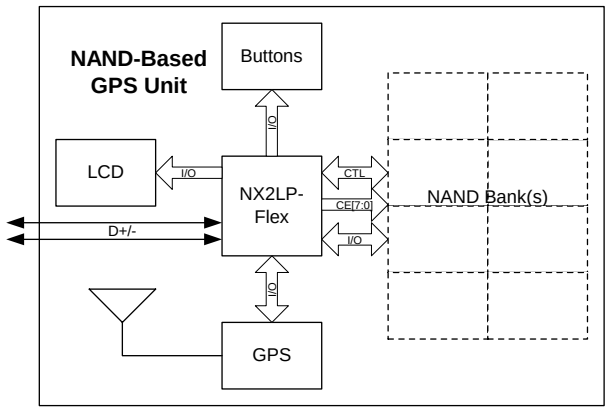


Figure 3-2. Example GPS Block Diagram

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. Please visit http://www.cypress.com for more information.

4.0 Functional Overview

4.1 USB Signaling Speed

NX2LP-Flex 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.

NX2LP-Flex does not support the low-speed signaling mode of 1.5 Mbps.

4.2 8051 Microprocessor

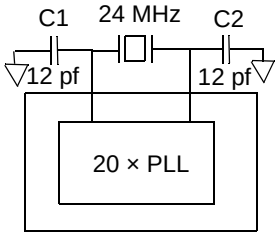
The 8051 microprocessor embedded in the NX2LP-Flex has 256 bytes of register RAM, an expanded interrupt system and three timer/counters.

4.2.1 8051 Clock Frequency

NX2LP-Flex 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, and internal counters divide it down for use as the 8051 clock. The default 8051 clock frequency is 12 MHz. The clock frequency of the 8051 can be changed by the 8051 through the CPUCS register, dynamically.



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

Figure 4-1. Crystal Configuration

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

4.2.2 Special Function Registers

Certain 8051 SFR addresses are populated to provide fast access to critical NX2LP-Flex functions. These SFR additions are shown in *Table 4-1*. Bold type indicates non-standard, enhanced 8051 registers. The two SFR rows that end with “0” and “8” contain bit-addressable registers. The four I/O ports A–D use the SFR addresses used in the standard 8051 for ports 0–3, which are not implemented in NX2LP-Flex. Because of the faster and more efficient SFR addressing, the NX2LP-Flex I/O ports are not addressable in external RAM space (using the MOVX instruction).

4.3 I^2C Bus

NX2LP supports the I^2C bus as a master only at 100-/400-kHz. SCL and SDA pins have open-drain outputs and hysteresis inputs. These signals must be pulled up to 3.3V, even if no I^2C device is connected. The I^2C bus is disabled at startup and only available for use after the initial NAND access.

Table 4-1. Special Function Registers

x	8x	9x	Ax	Bx	Cx	Dx	Ex	Fx
0	**IOA**	**IOB**	**IOC**	**IOD**	SCON1	PSW	ACC	B
1	SP	**EXIF**	**INT2CLR**	**IOE**	SBUF1			
2	DPL0	**MPAGE**	**INT4CLR**	**OEA**				
3	DPH0			**OEB**				
4	**DPL1**			**OEC**				
5	**DPH1**			**OED**				
6	**DPS**			**OEE**				
7	PCON							
8	TCON	SCON0	IE	IP	T2CON	**EICON**	**EIE**	**EIP**
9	TMOD	SBUF0						
A	TL0	**AUTOPTRH1**	**EP2468STAT**	**EP01STAT**	RCAP2L			
B	TL1	**AUTOPTRL1**	**EP24FIFOFLGS**	**GPIFTRIG**	RCAP2H			
C	TH0	**RESERVED**	**EP68FIFOFLGS**		TL2			
D	TH1	**AUTOPTRH2**		**GPIFSGLDATH**	TH2			
E	**CKCON**	**AUTOPTRL2**		**GPIFSGLDATLX**				
F		**RESERVED**	**AUTOPTRSET-UP**	**GPIFSGLDATLNOX**				

4.4 Buses

The NX2LP-Flex features an 8- or 16-bit “FIFO” bidirectional data bus, multiplexed on I/O ports B and D.

The default firmware image implements an 8-bit data bus in GPIF Master mode. It is recommended that additional interfaces added to the default firmware image use this 8-bit data bus.

4.5 Enumeration

During the start-up sequence, internal logic checks for the presence of NAND Flash with valid firmware. If valid firmware is found, the NX2LP-Flex loads it and operates according to the firmware. If no NAND Flash is detected, or if no valid firmware is found, the NX2LP-Flex uses the default values from internal ROM space for manufacturing mode operation. The two modes of operation are described in sections 4.5.1 and 4.5.2 below.

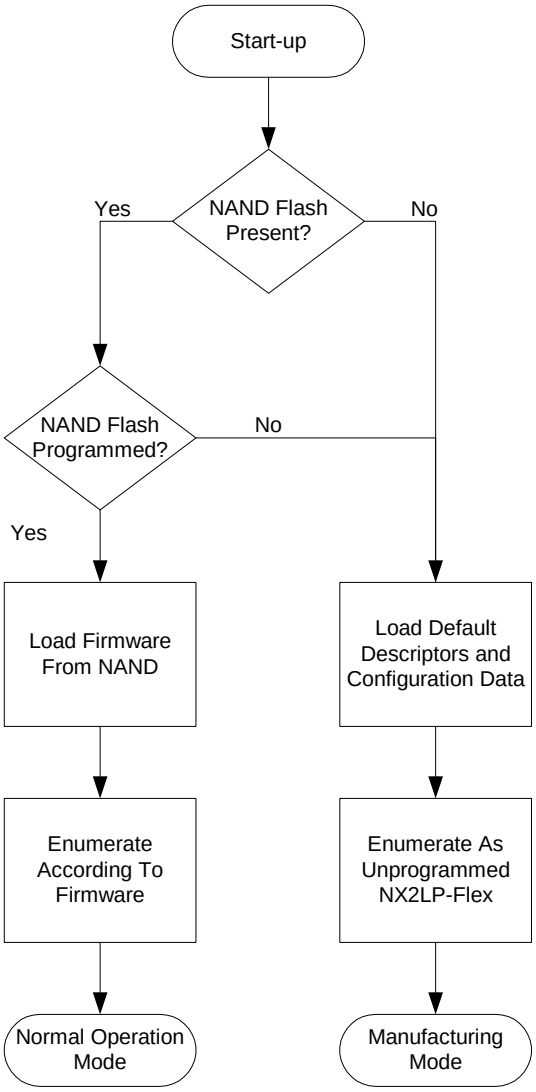


Figure 4-2. NX2LP-Flex Enumeration Sequence

4.5.1 Normal Operation Mode

In Normal Operation Mode, the NX2LP-Flex behaves as a USB 2.0 Mass Storage Class NAND Flash controller. This includes all typical USB device states (powered, configured, etc.). The USB descriptors are returned according to the data stored in the configuration data memory area. Normal read and write access to the NAND Flash is available in this mode.

4.5.2 Manufacturing Mode

In Manufacturing Mode, the NX2LP-Flex enumerates using the default descriptors and configuration data that are stored in internal ROM space. This mode allows for first-time programming of the configuration data memory area, as well as board-level manufacturing tests.

4.6 Default Silicon ID Values

To facilitate proper USB enumeration when no programmed NAND Flash is present, the NX2LP-Flex has default silicon ID values stored in ROM space. The default silicon ID values should only be used for development purposes. Cypress requires designers to use their own Vendor ID for final products. A Vendor ID is obtained through registration with the USB Implementor's Forum (USB-IF). Also, if the NX2LP-Flex is used as a mass storage class device, a unique USB serial number is required for each device in order to comply with the USB Mass Storage class specification.

Cypress provides all the software tools and drivers necessary for properly programming and testing the NX2LP-Flex. Please refer to the documentation in the development kit for more information on these topics.

Table 4-2. Default Silicon ID Values

Default VID/PID/DID		
Vendor ID	0x04B4	Cypress Semiconductor
Product ID	0x8613	EZ-USB® Default
Device release	0xAnnn	Depends on chip revision (nnn = chip revision, where first silicon = 001)

4.7 ReNumeration™

Cypress's ReNumeration™ feature is used in conjunction with the NX2LP-Flex manufacturing software tools to enable first-time NAND programming. It is only available when used in conjunction with the NX2LP-Flex Manufacturing tools, and is not enabled during normal operation.

4.8 Bus-powered Applications

The NX2LP-Flex fully supports bus-powered designs by enumerating with less than 100 mA, as required by the USB 2.0 specification.

4.9 Interrupt System

4.9.1 INT2 Interrupt Request and Enable Registers

NX2LP-Flex implements an autovector feature for INT2 and INT4. There are 27 INT2 (USB) vectors, and 14 INT4 (FIFO/GPIF) vectors. See the EZ-USB Technical Reference Manual (TRM) for more details.

4.9.2 USB-Interrupt Autovectors

The main USB interrupt is shared by 27 interrupt sources. To save the code and processing time that normally would be required to identify the individual USB interrupt source, the NX2LP-Flex provides a second level of interrupt vectoring, called Autovectoring. When a USB interrupt is asserted, the NX2LP-Flex pushes the program counter onto its stack then jumps to address 0x0500, where it expects to find a "jump" instruction to the USB Interrupt service routine.

Developers familiar with Cypress's programmable USB devices should note that these interrupt vector values differ from those used in other EZ-USB microcontrollers. This is due to the additional NAND boot logic that is present in the NX2LP-Flex ROM space. Also, these values are fixed and cannot be changed in the firmware.

Table 4-3. INT2 USB Interrupts

USB INTERRUPT TABLE FOR INT2			
Priority	**INT2VEC Value**	**Source**	**Notes**
1	0x500	SUDAV	Set-up Data Available
2	0x504	SOF	Start of Frame (or microframe)
3	0x508	SUTOK	Set-up Token Received
4	0x50C	SUSPEND	USB Suspend request
5	0x510	USB RESET	Bus reset
6	0x514	HISPEED	Entered high speed operation
7	0x518	EP0ACK	NX2LP ACK'd the CONTROL Handshake
8	0x51C		Reserved
9	0x520	EP0-IN	EP0-IN ready to be loaded with data
10	0x524	EP0-OUT	EP0-OUT has USB data
11	0x528	EP1-IN	EP1-IN ready to be loaded with data
12	0x52C	EP1-OUT	EP1-OUT has USB data
13	0x530	EP2	IN: buffer available. OUT: buffer has data
14	0x534	EP4	IN: buffer available. OUT: buffer has data
15	0x538	EP6	IN: buffer available. OUT: buffer has data
16	0x53C	EP8	IN: buffer available. OUT: buffer has data
17	0x540	IBN	IN-Bulk-NAK (any IN endpoint)
18	0x544		Reserved
19	0x548	EP0PING	EP0 OUT was Pinged and it NAK'd
20	0x54C	EP1PING	EP1 OUT was Pinged and it NAK'd
21	0x550	EP2PING	EP2 OUT was Pinged and it NAK'd
22	0x554	EP4PING	EP4 OUT was Pinged and it NAK'd
23	0x558	EP6PING	EP6 OUT was Pinged and it NAK'd
24	0x55C	EP8PING	EP8 OUT was Pinged and it NAK'd
25	0x560	ERRLIMIT	Bus errors exceeded the programmed limit
26	0x564		Reserved
27	0x568		Reserved
28	0x56C		Reserved
29	0x570	EP2ISOERR	ISO EP2 OUT PID sequence error
30	0x574	EP4ISOERR	ISO EP4 OUT PID sequence error
31	0x578	EP6ISOERR	ISO EP6 OUT PID sequence error
32	0x57C	EP8ISOERR	ISO EP8 OUT PID sequence error

If Autovectoring is enabled (AV2EN = 1 in the INTSET-UP register), the NX2LP-Flex substitutes its INT2VEC byte. Therefore, if the high byte ("page") of a jump-table address is preloaded at location 0x544, the automatically-inserted INT2VEC byte at 0x545 will direct the jump to the correct address out of the 27 addresses within the page.

4.9.3 FIFO/GPIF Interrupt (INT4)

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

Table 4-4. Individual FIFO/GPIF Interrupt Sources

Priority	INT4VEC Value	Source	Notes
1	0x580	EP2PF	Endpoint 2 Programmable Flag
2	0x584	EP4PF	Endpoint 4 Programmable Flag
3	0x588	EP6PF	Endpoint 6 Programmable Flag
4	0x58C	EP8PF	Endpoint 8 Programmable Flag
5	0x590	EP2EF	Endpoint 2 Empty Flag
6	0x594	EP4EF	Endpoint 4 Empty Flag
7	0x598	EP6EF	Endpoint 6 Empty Flag
8	0x59C	EP8EF	Endpoint 8 Empty Flag
9	0x5A0	EP2FF	Endpoint 2 Full Flag
10	0x5A4	EP4FF	Endpoint 4 Full Flag
11	0x5A8	EP6FF	Endpoint 6 Full Flag
12	0x5AC	EP8FF	Endpoint 8 Full Flag
13	0x5B0	GPIFDONE	GPIF Operation Complete
14	0x5B4	GPIFWF	GPIF Waveform

If Autovectoring is enabled (AV4EN = 1 in the INTSET-UP register), the NX2LP-Flex substitutes its INT4VEC byte. Therefore, if the high byte (“page”) of a jump-table address is preloaded at location 0x554, the automatically-inserted INT4VEC byte at 0x555 will direct the jump to the correct address out of the 14 addresses within the page. When the ISR occurs, the NX2LP-Flex pushes the program counter onto its stack then jumps to address 0x553, where it expects to find a “jump” instruction to the ISR Interrupt service routine.

4.10 Reset and Wakeup

4.10.1 Reset Pin

The input pin RESET#, will reset the NX2LP-Flex when asserted. This pin has hysteresis and is active LOW. When a crystal is used as the clock source for the NX2LP-Flex, the reset period must allow for the stabilization of the crystal and the PLL. This reset period should be approximately 5 ms after V_{CC} has reached 3.0V. If the crystal input pin is driven by a clock signal, the internal PLL stabilizes in 200 µs after V_{CC} has reached 3.0V[1]. *Figure 4-3* shows a power-on reset condition and a reset applied during operation. A power-on reset is defined as the time reset is asserted while power is being applied to the circuit. A powered reset is defined to be when the NX2LP-Flex has previously been powered on and operating and the RESET# pin is asserted.

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

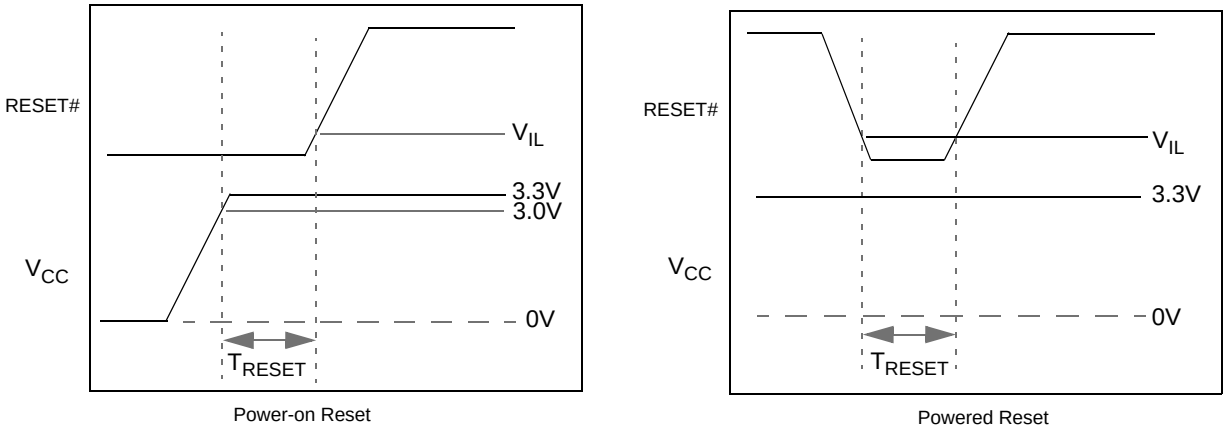


Figure 4-3. Reset Timing Plots

Note:
1. If the external clock is powered at the same time as the CY7C68033/CY7C68034 and has a stabilization wait period, it must be added to the 200 µs.

Table 4-5. Reset Timing Values

Condition	T_{RESET}
Power-on Reset with crystal	5 ms
Power-on Reset with external clock source	200 µs + Clock stability time
Powered Reset	200 µs

4.10.2 Wakeup Pins

The 8051 puts itself and the rest of the chip into a power-down mode by setting PCON.0 = 1. This stops the oscillator and PLL. When WAKEUP is asserted by external logic, the oscillator restarts, after the PLL stabilizes, and then the 8051 receives a wakeup interrupt. This applies whether or not NX2LP-Flex is connected to the USB.

The NX2LP-Flex 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 NX2LP-Flex and initiate a wakeup).
- External logic asserts the WAKEUP pin
- External logic asserts the PA3/WU2 pin.

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

4.11 Program/Data RAM

4.11.1 Internal ROM/RAM Size

The NX2LP-Flex has 1 kBytes ROM and 15 kBytes of internal program/data RAM, where PSEN#/RD# signals are internally ORed to allow the 8051 to access it as both program and data memory. No USB control registers appear in this space.

4.11.2 Internal Code Memory

This mode implements the internal block of RAM (starting at 0x0500) as combined code and data memory, as shown in *Figure 4-4*, below.

Only the internal and scratch pad RAM spaces have the following access:

- USB download (only supported by the Cypress Manufacturing Tool)
- Set-up data pointer
- NAND boot access.

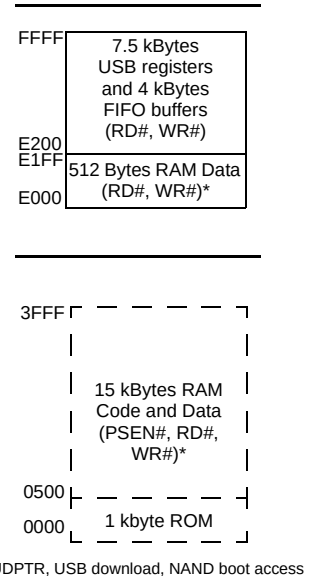


Figure 4-4. Internal Code Memory

4.12 Register Addresses

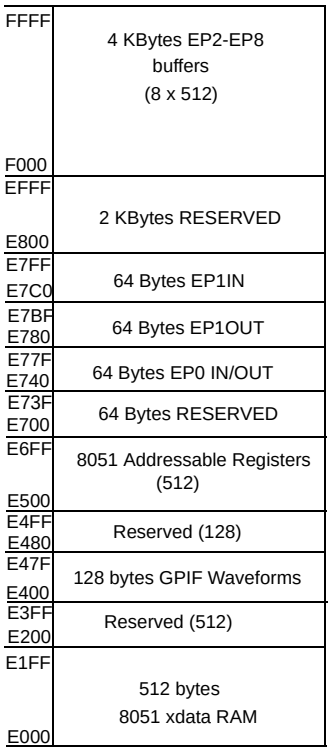


Figure 4-5. Internal Register Addresses

4.13 Endpoint RAM

4.13.1 Size

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

4.13.2 Organization

- EP0
  - Bidirectional endpoint zero, 64-byte buffer
- EP1IN, EP1OUT
  - 64-byte buffers, bulk or interrupt
- EP2,4,6,8
  - Eight 512-byte buffers, bulk, interrupt, or isochronous.
  - EP4 and EP8 can be double buffered, while EP2 and 6 can be either double, triple, or quad buffered.

For high-speed endpoint configuration options, see *Figure 4-6*.

4.13.3 Set-up Data Buffer

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

4.13.4 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 1 of the 12 configurations shown in the vertical columns. When operating in full-speed BULK mode, only the first 64 bytes of each buffer are used. For example, in high-speed the max 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 would be:

EP2–1024 double buffered; EP6–512 quad buffered (column 8 in *Figure 4-6*).

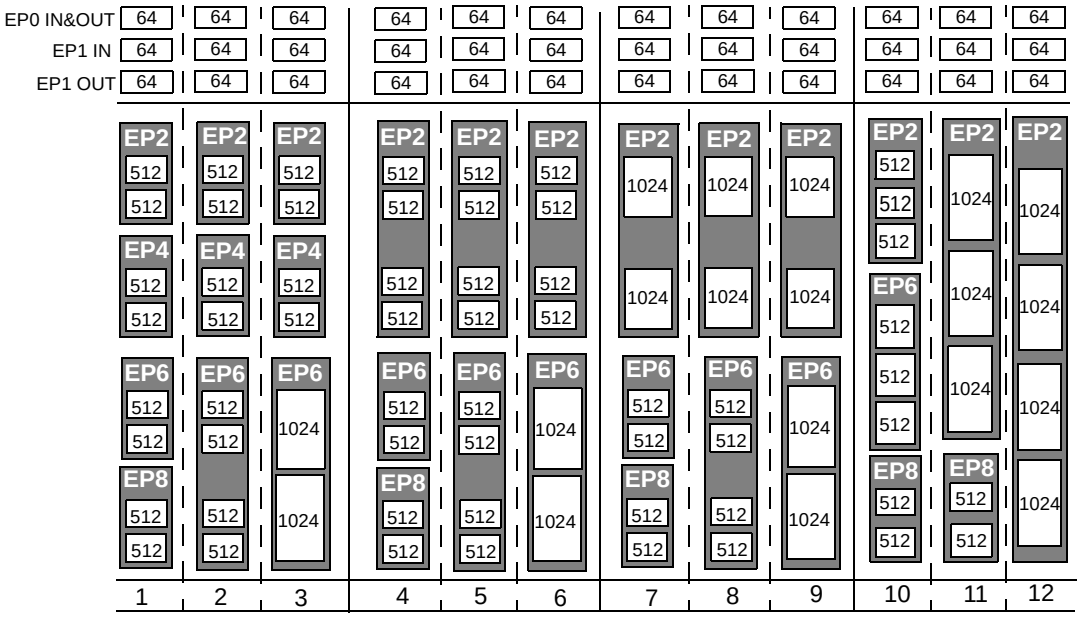


Figure 4-6. Endpoint Configuration

4.13.5 Default Full-Speed Alternate Settings

Table 4-6. Default Full-Speed Alternate Settings[2, 3]

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×)

Notes:
2. "0" means "not implemented."
3. "2×" means "double buffered."

4.13.6 Default High-Speed Alternate Settings

Table 4-7. Default High-Speed Alternate Settings[2, 3]

Alternate Setting	0	1	2	3
ep0	64	64	64	64
ep1out	0	512 bulk[4]	64 int	64 int
ep1in	0	512 bulk[4]	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×)

4.14 External FIFO Interface

4.14.1 Architecture

The NX2LP-Flex slave FIFO architecture has eight 512-byte blocks in the endpoint RAM that directly serve as FIFO memories, and are controlled by FIFO control signals (such as IFCLK, SLCS#, SLRD, SLWR, SLOE, PKTEND, and flags).

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

4.14.2 Master/Slave Control Signals

The NX2LP-Flex endpoint FIFOS are implemented as eight physically distinct 256x16 RAM blocks. The 8051/SIE can switch any of the RAM blocks between two domains, the USB (SIE) domain and the 8051-I/O Unit domain. This switching is done virtually instantaneously, giving essentially zero transfer time between “USB FIFOS” and “Slave FIFOS.” Since they are physically the same memory, no bytes are actually transferred between buffers.

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

The I/O control unit implements either an internal-master (M for master) or external-master (S for Slave) interface.

In Master (M) mode, the GPIF internally controls FIFOADR[1:0] to select a FIFO. The two RDY pins can be used as flag inputs from an external FIFO or other logic if desired. The GPIF can be run from either an internally derived clock or externally supplied clock (IFCLK), at a rate that transfers data up to 96 Megabytes/s (48-MHz IFCLK with 16-bit interface).

In Slave (S) mode, the NX2LP-Flex 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 from the internal clock source with the IFCLKSRC bit. Each endpoint can individually be selected for byte or word operation by an internal configuration bit, and a Slave FIFO Output Enable signal SLOE enables data of the selected width. External logic must insure that the output enable signal is inactive when writing data to a slave FIFO. The slave interface can also operate asynchronously, where the SLRD and SLWR signals act directly as strobes, rather than a clock qualifier as in synchronous mode. The signals SLRD, SLWR, SLOE and PKTEND are gated by the signal SLCS#.

4.14.3 GPIF and FIFO Clock Rates

An 8051 register bit selects one of two frequencies for the internally supplied interface clock: 30 MHz and 48 MHz. Alternatively, an externally supplied clock of 5 MHz–48 MHz feeding the IFCLK pin can be used as the interface clock. IFCLK can be configured to function as a clock output signal when the GPIF and FIFOs are internally clocked. An output enable bit in the IFCONFIG register turns this clock output off, if desired. Another bit within the IFCONFIG register will invert the IFCLK signal, whether internally or externally sourced.

The default NAND firmware image implements a 48-MHz internally supplied interface clock and disables the IFCLK output. The NAND boot logic uses the same configuration to implement 100-ns timing on the NAND bus to support proper detection of all NAND Flash types.

4.15 GPIF

The GPIF is a flexible 8- or 16-bit parallel interface driven by a user-programmable finite state machine. It allows the NX2LP-Flex to perform local bus mastering, and can implement a wide variety of protocols such as 8-bit NAND interface, printer parallel port, and Utopia. The default NAND firmware and boot logic utilizes GPIF functionality to interface with NAND Flash.

The GPIF on the NX2LP-Flex features three programmable control outputs (CTL) and two general-purpose ready inputs (RDY). The GPIF data bus width can be 8 or 16 bits. Because the default NAND firmware image implements an 8-bit data bus and up to 8 chip enable pins on the GPIF ports, it is recommended that designs based upon the default firmware image use an 8-bit data bus as well.

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

Note:
4. Even though these buffers are 64 bytes, they are reported as 512 for USB 2.0 compliance. The user must never transfer packets larger than 64 bytes to EP1.

waveform that will be executed to perform the desired data move between the NX2LP-Flex and the external device.

4.15.1 Three Control OUT Signals

The NX2LP-Flex exposes three control signals, CTL[2:0]. CTLx waveform edges can be programmed to make transitions as fast as once per clock (20.8 ns using a 48-MHz clock).

4.15.2 Two Ready IN Signals

The 8051 programs the GPIF unit to test the RDY pins for GPIF branching. The 56-pin package brings out two signals, RDY[1:0].

4.15.3 Long Transfer Mode

In GPIF 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 over-flow until the full number of requested transactions complete. The GPIF decrements the value in these registers to represent the current status of the transaction.

4.16 ECC Generation[5]

The NX2LP-Flex 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)
- One ECC calculated over 512 bytes.

The two ECC configurations described below are selected by the ECCM bit. The ECC can correct any one-bit error or detect any two-bit error.

4.16.1 ECCM = 0

Two 3-byte ECCs, each calculated over a 256-byte block of data. This configuration conforms to the SmartMedia Standard and is used by both the NAND boot logic and default NAND firmware image.

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

4.16.2 ECCM = 1

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

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

4.17 Autopointer Access

NX2LP-Flex 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. Also, the autopointers can point to any NX2LP-Flex register or endpoint buffer space.

4.18 I^2C Controller

NX2LP has one I^2C port that the 8051, once running uses to control external I^2C devices. The I^2C port operates in master mode only. The I^2C post is disabled at startup and only available for use after the initial NAND access.

4.18.1 I^2C Port Pins

The I^2C pins SCL and SDA must have external 2.2-kΩ pull-up resistors even if no EEPROM is connected to the NX2LP.

4.18.2 I^2C Interface General-Purpose Access

The 8051 can control peripherals connected to the I^2C bus using the I^2CTL and I^2DATA registers. NX2LP provides I^2C master control only and is never an I^2C slave.

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

5.0 Pin Assignments

Figure 5-1 and *Figure 5-2* identify all signals for the 56-pin NX2LP-Flex package.

Three modes of operation are available for the NX2LP-Flex: Port mode, GPIF Master mode, and Slave FIFO mode. These modes define the signals on the right edge of each column in *Figure 5-1*. The right-most column details the signal functionality from the default NAND firmware image, which actually utilizes GPIF Master mode. The signals on the left edge of the "Port" column are common to all modes of the NX2LP-Flex. The 8051 selects the interface mode using the IFCONFIG[1:0] register bits. Port mode is the power-on default configuration.

Figure 5-2 details the pinout of the 56-pin package and lists pin names for all modes of operation. Pin names with an asterisk (*) feature programmable polarity.

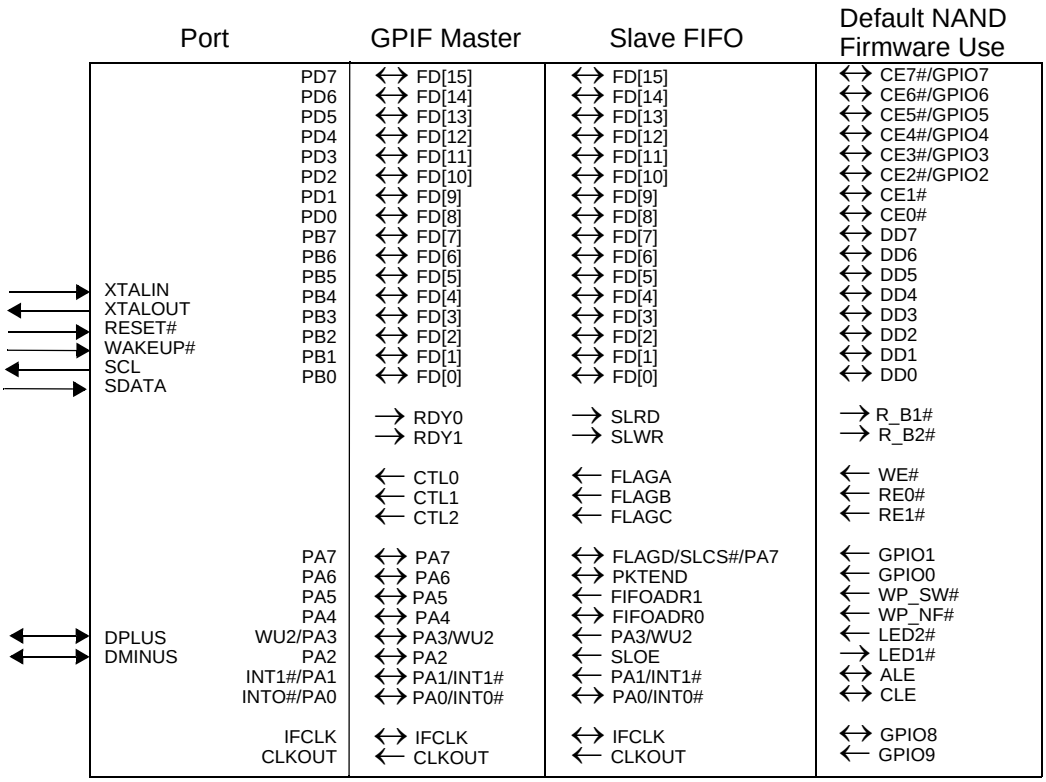


Figure 5-1. Port and Signal Mapping

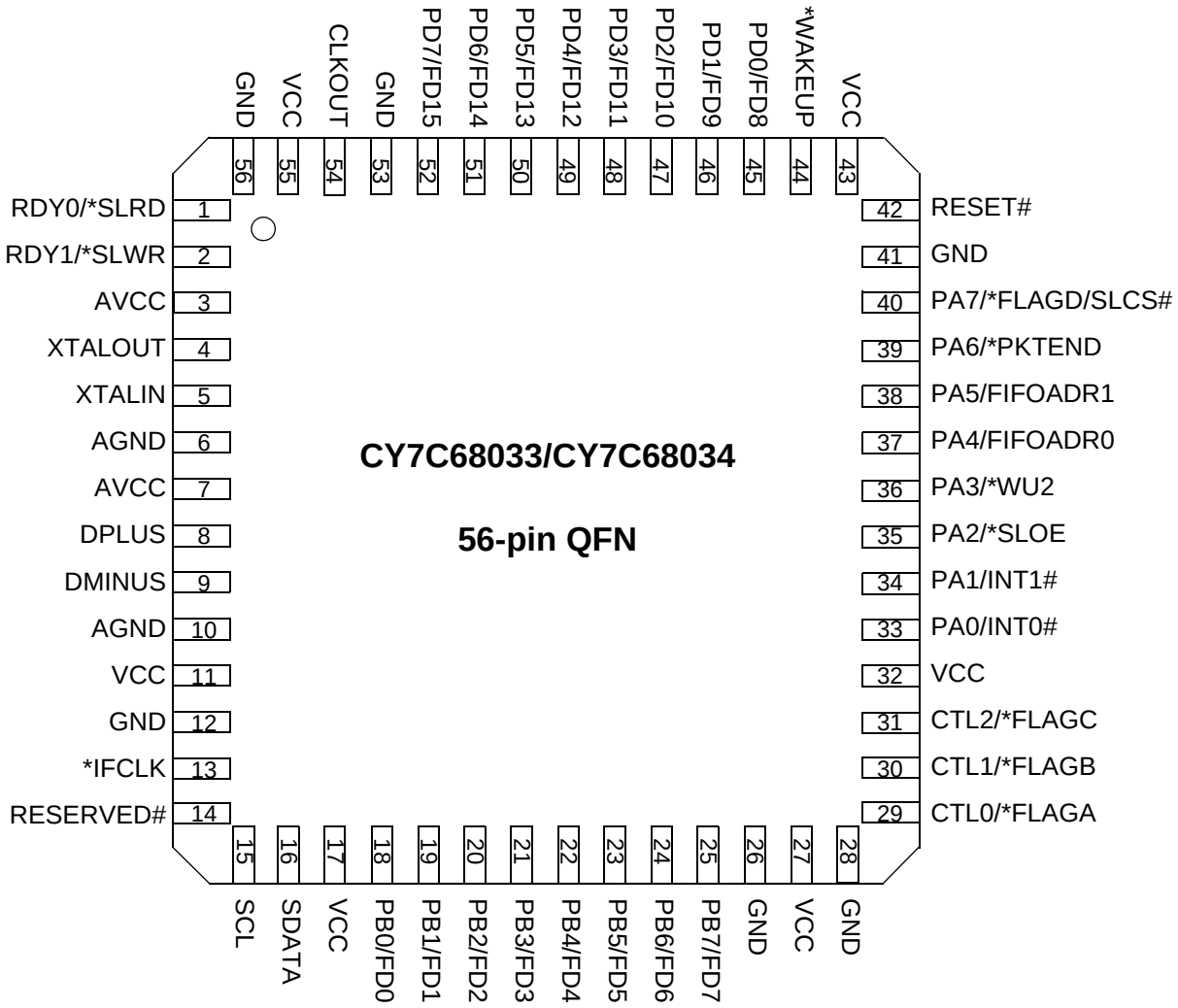


Figure 5-2. CY7C68033/CY7C68034 56-pin QFN Pin Assignment

5.1 Pin Descriptions

Table 5-1. NX2LP-Flex Pin Descriptions [6]

56 QFN Pin Number	Default Pin Name	NAND Firmware Usage	Pin Type	Default State	Description
9	DMINUS	N/A	I/O/Z	Z	**USB D– Signal**. Connect to the USB D– signal.
8	DPLUS	N/A	I/O/Z	Z	**USB D+ Signal**. Connect to the USB D+ signal.
42	RESET#	N/A	Input	N/A	**Active LOW Reset**. Resets the entire chip. See section 4.10 ”Reset and Wakeup” on page 7 for more details.
5	XTALIN	N/A	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. When driving from an external source, the driving signal should be a 3.3V square wave.
4	XTALOUT	N/A	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.
54	CLKOUT	GPIO9	O/Z	12 MHz	**CLKOUT:** 12-, 24-, or 48-MHz clock, phase locked to the 24-MHz input clock. The 8051 defaults to 12-MHz operation. The 8051 may three-state this output by setting CPUCS[1] = 1.
1	RDY0 or SLRD	R_B1#	Input	N/A	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **RDY0** is a GPIF input signal. **SLRD** is the input-only read strobe with programmable polarity (FIFOPINPOLAR[3]) for the slave FIFOs connected to FD[7:0] or FD[15:0]. **R_B1#** is a NAND Ready/Busy input signal.
2	RDY1 or SLWR	R_B2#	Input	N/A	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **RDY1** is a GPIF input signal. **SLWR** is the input-only write strobe with programmable polarity (FIFOPINPOLAR[2]) for the slave FIFOs connected to FD[7:0] or FD[15:0]. **R_B2#** is a NAND Ready/Busy input signal.
29	CTL0 or FLAGA	WE#	O/Z	H	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **CTL0** is a GPIF control output. **FLAGA** is a programmable slave-FIFO output status flag signal. Defaults to programmable for the FIFO selected by the FIFOADR[1:0] pins. **WE#** is the NAND write enable output signal.
30	CTL1 or FLAGB	RE0#	O/Z	H	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **CTL1** is a GPIF control output. **FLAGB** is a programmable slave-FIFO output status flag signal. Defaults to FULL for the FIFO selected by the FIFOADR[1:0] pins. **RE0#** is a NAND read enable output signal.
31	CTL2 or FLAGC	RE1#	O/Z	H	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **CTL2** is a GPIF control output. **FLAGC** is a programmable slave-FIFO output status flag signal. Defaults to EMPTY for the FIFO selected by the FIFOADR[1:0] pins. **RE1#** is a NAND read enable output signal.
13	IFCLK	GPIO8	I/O/Z	I	Interface Clock, used for synchronously clocking data into or out of the slave FIFOs. IFCLK also serves as a timing reference for all slave FIFO control signals and GPIF. When internal clocking is used (IFCONFIG[7] = 1) the IFCLK pin can be configured to output 30/48 MHz by bits IFCONFIG[5] and IFCONFIG[6]. IFCLK may be inverted, whether internally or externally sourced, by setting the bit IFCONFIG[4] =1.

Note:

6. Unused inputs should not be left floating. Tie either HIGH or LOW as appropriate. Outputs should only be pulled up or down to ensure signals at power-up and in standby. Note also that no pins should be driven while the device is powered down.

Table 5-1. NX2LP-Flex Pin Descriptions (continued)[6]

56 QFN Pin Number	Default Pin Name	NAND Firmware Usage	Pin Type	Default State	Description
14	Reserved#	N/A	Input	N/A	**Reserved**. Connect to ground.
15	SCL	N/A	OD	Z	Clock for the I^2C interface. Connect to VCC with a 2.2K resistor, even if no I^2C peripheral is attached.
16	SDATA	N/A	OD	Z	Data for the I^2C interface. Connect to VCC with a 2.2K resistor, even if no I^2C peripheral is attached.
44	WAKEUP	Unused	Input	N/A	**USB Wakeup**. If the 8051 is in suspend, asserting this pin starts up the oscillator and interrupts the 8051 to allow it to exit the suspend mode. Holding WAKEUP asserted inhibits the EZ-USB chip from suspending. This pin has programmable polarity, controlled by WAKEUP[4].
Port A					
33	PA0 or INT0#	CLE	I/O/Z	I (PA0)	Multiplexed pin whose function is selected by PORTACFG[0] **PA0** is a bidirectional IO port pin. **INT0#** is the active-LOW 8051 INT0 interrupt input signal, which is either edge triggered (IT0 = 1) or level triggered (IT0 = 0). **CLE** is the NAND Command Latch Enable signal.
34	PA1 or INT1#	ALE	I/O/Z	I (PA1)	Multiplexed pin whose function is selected by PORTACFG[1] **PA1** is a bidirectional IO port pin. **INT1#** is the active-LOW 8051 INT1 interrupt input signal, which is either edge triggered (IT1 = 1) or level triggered (IT1 = 0). **ALE** is the NAND Address Latch Enable signal.
35	PA2 or SLOE or	LED1#	I/O/Z	I (PA2)	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **PA2** is a bidirectional IO 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]. **LED1#** is the data activity indicator LED sink pin.
36	PA3 or WU2	LED2#	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. **LED2#** is the chip activity indicator LED sink pin.
37	PA4 or FIFOADR0	WP_NF#	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]. **WP_NF#** is the NAND write-protect control output signal.
38	PA5 or FIFOADR1	WP_SW#	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]. **WP_SW#** is the NAND write-protect switch input signal.
39	PA6 or PKTEND	GPIO0 (Input)	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 used to commit the FIFO packet data to the endpoint and whose polarity is programmable via FIFOPINPOLAR[5]. **GPIO1** is a general purpose I/O signal.

Table 5-1. NX2LP-Flex Pin Descriptions (continued)[6]

56 QFN Pin Number	Default Pin Name	NAND Firmware Usage	Pin Type	Default State	Description
40	PA7 or FLAGD or SLCS#	GPIO1 (Input)	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 **GPIO0** is a general purpose I/O signal.
Port B					
18	PB0 or FD[0]	DD0	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. **DD0** is a bidirectional NAND data bus signal.
19	PB1 or FD[1]	DD1	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. **DD1** is a bidirectional NAND data bus signal.
20	PB2 or FD[2]	DD2	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. **DD2** is a bidirectional NAND data bus signal.
21	PB3 or FD[3]	DD3	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. **DD3** is a bidirectional NAND data bus signal.
22	PB4 or FD[4]	DD4	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. **DD4** is a bidirectional NAND data bus signal.
23	PB5 or FD[5]	DD5	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. **DD5** is a bidirectional NAND data bus signal.
24	PB6 or FD[6]	DD6	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. **DD6** is a bidirectional NAND data bus signal.
25	PB7 or FD[7]	DD7	I/O/Z	I (PB7)	Multiplexed pin whose function is selected by IFCONFIG[1:0]. **PB7** is a bidirectional I/O port pin. **FD[7]** is the bidirectional FIFO/GPIF data bus. **DD7** is a bidirectional NAND data bus signal.
PORT D					
45	PD0 or FD[8]	CE0#	I/O/Z	I (PD0)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. **FD[8]** is the bidirectional FIFO/GPIF data bus. **CE0#** is a NAND chip enable output signal.
46	PD1 or FD[9]	CE1#	I/O/Z	I (PD1)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. **FD[9]** is the bidirectional FIFO/GPIF data bus. **CE1#** is a NAND chip enable output signal.
47	PD2 or FD[10]	CE2# or GPIO2	I/O/Z	I (PD2)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. **FD[10]** is the bidirectional FIFO/GPIF data bus. **CE2#** is a NAND chip enable output signal. **GPIO2** is a general purpose I/O signal.

Table 5-1. NX2LP-Flex Pin Descriptions (continued)[6]

56 QFN Pin Number	Default Pin Name	NAND Firmware Usage	Pin Type	Default State	Description
48	PD3 or FD[11]	CE3# or GPIO3	I/O/Z	I (PD3)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. **FD[11]** is the bidirectional FIFO/GPIF data bus. **CE3#** is a NAND chip enable output signal. **GPIO3** is a general purpose I/O signal.
49	PD4 or FD[12]	CE4# or GPIO4	I/O/Z	I (PD4)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. **FD[12]** is the bidirectional FIFO/GPIF data bus. **CE4#** is a NAND chip enable output signal. **GPIO4** is a general purpose I/O signal.
50	PD5 or FD[13]	CE5# or GPIO5	I/O/Z	I (PD5)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. **FD[13]** is the bidirectional FIFO/GPIF data bus. **CE5#** is a NAND chip enable output signal. **GPIO5** is a general purpose I/O signal.
51	PD6 or FD[14]	CE6# or GPIO6	I/O/Z	I (PD6)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. **FD[14]** is the bidirectional FIFO/GPIF data bus. **CE6#** is a NAND chip enable output signal. **GPIO6** is a general purpose I/O signal.
52	PD7 or FD[15]	CE7# or GPIO7	I/O/Z	I (PD7)	Multiplexed pin whose function is selected by the IFCONFIG[1:0] and EPxFIFOCFG.0 (wordwide) bits. **FD[15]** is the bidirectional FIFO/GPIF data bus. **CE7#** is a NAND chip enable output signal. **GPIO7** is a general purpose I/O signal.
Power and Ground					
3 7	AVCC	N/A	Power	N/A	**Analog V_{CC}**. Connect this pin to 3.3V power source. This signal provides power to the analog section of the chip.
6 10	AGND	N/A	Ground	N/A	**Analog Ground**. Connect to ground with as short a path as possible.
11 17 27 32 43 55	VCC	N/A	Power	N/A	**V_{CC}**. Connect to 3.3V power source.
12 26 28 41 53 56	GND	N/A	Ground	N/A	**Ground**.

6.0 Register Summary

NX2LP-Flex register bit definitions are described in the EZ-USB TRM in greater detail. Some registers that are listed here and in the TRM do not apply to the NX2LP-Flex. They are kept here for consistency reasons only. Registers that do not apply to the NX2LP-Flex should be left at their default power-up values.

Table 6-1. NX2LP-Flex Register Summary

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
		GPIF Waveform Memories											
E400	128	WAVEDATA	GPIF Waveform Descriptor 0, 1, 2, 3 data	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
E480	128	reserved											
		GENERAL CONFIGURATION											
E50D		GPCR2	General Purpose Configuration Register 2	reserved	reserved	reserved	FULL_SPEED_ONLY	reserved	reserved	reserved	reserved	00000000	R
E600	1	CPUCS	CPU Control & Status	0	0	PORTCSTB	CLKSPD1	CLKSPD0	CLKINV	CLKOE	8051RES	00000010	rrbbbbbr
E601	1	IFCONFIG	Interface Configuration (Ports, GPIF, slave FIFOs)	IFCLKSRC	3048MHZ	IFCLKOE	IFCLKPOL	ASYNC	GSTATE	IFCFG1	IFCFG0	10000000	RW
E602	1	PINFLAGSAB[7]	Slave FIFO FLAGA and FLAGB Pin Configuration	FLAGB3	FLAGB2	FLAGB1	FLAGB0	FLAGA3	FLAGA2	FLAGA1	FLAGA0	00000000	RW
E603	1	PINFLAGSCD[7]	Slave FIFO FLAGC and FLAGD Pin Configuration	FLAGD3	FLAGD2	FLAGD1	FLAGD0	FLAGC3	FLAGC2	FLAGC1	FLAGC0	00000000	RW
E604	1	FIFORESET[7]	Restore FIFOS to default state	NAKALL	0	0	0	EP3	EP2	EP1	EP0	xxxxxxxx	W
E605	1	BREAKPT	Breakpoint Control	0	0	0	0	BREAK	BPPULSE	BPEN	0	00000000	rrrrbbbr
E606	1	BPADDRH	Breakpoint Address H	A15	A14	A13	A12	A11	A10	A9	A8	xxxxxxxx	RW
E607	1	BPADDRL	Breakpoint Address L	A7	A6	A5	A4	A3	A2	A1	A0	xxxxxxxx	RW
E608	1	UART230	230 Kbaud internally generated ref. clock	0	0	0	0	0	0	230UART1	230UART0	00000000	rrrrrrbb
E609	1	FIFOPINPOLAR[7]	Slave FIFO Interface pins polarity	0	0	PKTEND	SLOE	SLRD	SLWR	EF	FF	00000000	rrbbbbbb
E60A	1	REVID	Chip Revision	rv7	rv6	rv5	rv4	rv3	rv2	rv1	rv0	RevA 00000001	R
E60B	1	REVCTL[7]	Chip Revision Control	0	0	0	0	0	0	dyn_out	enh_pkt	00000000	rrrrrrbb
		UDMA											
E60C	1	GPIFHOLDAMOUNT	MSTB Hold Time (for UDMA)	0	0	0	0	0	0	HOLDTIME1	HOLDTIME0	00000000	rrrrrrbb
	3	reserved											
		ENDPOINT CONFIGURATION											
E610	1	EP1OUTCFG	Endpoint 1-OUT Configuration	VALID	0	TYPE1	TYPE0	0	0	0	0	10100000	brbbrrrr
E611	1	EP1INCFG	Endpoint 1-IN Configuration	VALID	0	TYPE1	TYPE0	0	0	0	0	10100000	brbbrrrr
E612	1	EP2CFG	Endpoint 2 Configuration	VALID	DIR	TYPE1	TYPE0	SIZE	0	BUF1	BUF0	10100010	bbbbbrbb
E613	1	EP4CFG	Endpoint 4 Configuration	VALID	DIR	TYPE1	TYPE0	0	0	0	0	10100000	bbbbrrrr
E614	1	EP6CFG	Endpoint 6 Configuration	VALID	DIR	TYPE1	TYPE0	SIZE	0	BUF1	BUF0	11100010	bbbbbrbb
E615	1	EP8CFG	Endpoint 8 Configuration	VALID	DIR	TYPE1	TYPE0	0	0	0	0	11100000	bbbbrrrr
	2	reserved											
E618	1	EP2FIFOCFG[7]	Endpoint 2 / slave FIFO configuration	0	INFM1	OEP1	AUTOOUT	AUTOIN	ZEROLENIN	0	WORDWIDE	00000101	rbbbbbrb
E619	1	EP4FIFOCFG[7]	Endpoint 4 / slave FIFO configuration	0	INFM1	OEP1	AUTOOUT	AUTOIN	ZEROLENIN	0	WORDWIDE	00000101	rbbbbbrb
E61A	1	EP6FIFOCFG[7]	Endpoint 6 / slave FIFO configuration	0	INFM1	OEP1	AUTOOUT	AUTOIN	ZEROLENIN	0	WORDWIDE	00000101	rbbbbbrb
E61B	1	EP8FIFOCFG[7]	Endpoint 8 / slave FIFO configuration	0	INFM1	OEP1	AUTOOUT	AUTOIN	ZEROLENIN	0	WORDWIDE	00000101	rbbbbbrb
E61C	4	reserved											
E620	1	EP2AUTOINLENH[7]	Endpoint 2 AUTOIN Packet Length H	0	0	0	0	0	PL10	PL9	PL8	00000010	rrrrrbbb
E621	1	EP2AUTOINLENL[7]	Endpoint 2 AUTOIN Packet Length L	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
E622	1	EP4AUTOINLENH[7]	Endpoint 4 AUTOIN Packet Length H	0	0	0	0	0	0	PL9	PL8	00000010	rrrrrrbb
E623	1	EP4AUTOINLENL[7]	Endpoint 4 AUTOIN Packet Length L	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
E624	1	EP6AUTOINLENH[7]	Endpoint 6 AUTOIN Packet Length H	0	0	0	0	0	PL10	PL9	PL8	00000010	rrrrrbbb
E625	1	EP6AUTOINLENL[7]	Endpoint 6 AUTOIN Packet Length L	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
E626	1	EP8AUTOINLENH[7]	Endpoint 8 AUTOIN Packet Length H	0	0	0	0	0	0	PL9	PL8	00000010	rrrrrrbb
E627	1	EP8AUTOINLENL[7]	Endpoint 8 AUTOIN Packet Length L	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000	RW
E628	1	ECCCFG	ECC Configuration	0	0	0	0	0	0	0	ECCM	00000000	rrrrrrrb
E629	1	ECCRESET	ECC Reset	x	x	x	x	x	x	x	x	00000000	W
E62A	1	ECC1B0	ECC1 Byte 0 Address	LINE15	LINE14	LINE13	LINE12	LINE11	LINE10	LINE9	LINE8	00000000	R
E62B	1	ECC1B1	ECC1 Byte 1 Address	LINE7	LINE6	LINE5	LINE4	LINE3	LINE2	LINE1	LINE0	00000000	R

Note:

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

Table 6-1. NX2LP-Flex Register Summary (continued)

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
E62C	1	ECC1B2	ECC1 Byte 2 Address	COL5	COL4	COL3	COL2	COL1	COL0	LINE17	LINE16	00000000	R
E62D	1	ECC2B0	ECC2 Byte 0 Address	LINE15	LINE14	LINE13	LINE12	LINE11	LINE10	LINE9	LINE8	00000000	R
E62E	1	ECC2B1	ECC2 Byte 1 Address	LINE7	LINE6	LINE5	LINE4	LINE3	LINE2	LINE1	LINE0	00000000	R
E62F	1	ECC2B2	ECC2 Byte 2 Address	COL5	COL4	COL3	COL2	COL1	COL0	0	0	00000000	R
E630 H.S.	1	EP2FIFOPFH[7]	Endpoint 2 / slave FIFO Programmable Flag H	DECIS	PKTSTAT	IN:PKTS[2] OUT:PFC12	IN:PKTS[1] OUT:PFC11	IN:PKTS[0] OUT:PFC10	0	PFC9	PFC8	10001000	bbbbbrbb
E630 F.S.	1	EP2FIFOPFH[7]	Endpoint 2 / slave FIFO Programmable Flag H	DECIS	PKTSTAT	OUT:PFC12	OUT:PFC11	OUT:PFC10	0	PFC9	IN:PKTS[2] OUT:PFC8	10001000	bbbbbrbb
E631 H.S.	1	EP2FIFOPFL[7]	Endpoint 2 / slave FIFO Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E631 F.S	1	EP2FIFOPFL[7]	Endpoint 2 / slave FIFO Programmable Flag L	IN:PKTS[1] OUT:PFC7	IN:PKTS[0] OUT:PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E632 H.S.	1	EP4FIFOPFH[7]	Endpoint 4 / slave FIFO Programmable Flag H	DECIS	PKTSTAT	0	IN: PKTS[1] OUT:PFC10	IN: PKTS[0] OUT:PFC9	0	0	PFC8	10001000	bbrbbrrb
E632 F.S	1	EP4FIFOPFH[7]	Endpoint 4 / slave FIFO Programmable Flag H	DECIS	PKTSTAT	0	OUT:PFC10	OUT:PFC9	0	0	PFC8	10001000	bbrbbrrb
E633 H.S.	1	EP4FIFOPFL[7]	Endpoint 4 / slave FIFO Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E633 F.S	1	EP4FIFOPFL[7]	Endpoint 4 / slave FIFO Programmable Flag L	IN: PKTS[1] OUT:PFC7	IN: PKTS[0] OUT:PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E634 H.S.	1	EP6FIFOPFH[7]	Endpoint 6 / slave FIFO Programmable Flag H	DECIS	PKTSTAT	IN:PKTS[2] OUT:PFC12	IN:PKTS[1] OUT:PFC11	IN:PKTS[0] OUT:PFC10	0	PFC9	PFC8	00001000	bbbbbrbb
E634 F.S	1	EP6FIFOPFH[7]	Endpoint 6 / slave FIFO Programmable Flag H	DECIS	PKTSTAT	OUT:PFC12	OUT:PFC11	OUT:PFC10	0	PFC9	IN:PKTS[2] OUT:PFC8	00001000	bbbbbrbb
E635 H.S.	1	EP6FIFOPFL[7]	Endpoint 6 / slave FIFO Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E635 F.S	1	EP6FIFOPFL[7]	Endpoint 6 / slave FIFO Programmable Flag L	IN:PKTS[1] OUT:PFC7	IN:PKTS[0] OUT:PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E636 H.S.	1	EP8FIFOPFH[7]	Endpoint 8 / slave FIFO Programmable Flag H	DECIS	PKTSTAT	0	IN: PKTS[1] OUT:PFC10	IN: PKTS[0] OUT:PFC9	0	0	PFC8	00001000	bbrbbrrb
E636 F.S	1	EP8FIFOPFH[7]	Endpoint 8 / slave FIFO Programmable Flag H	DECIS	PKTSTAT	0	OUT:PFC10	OUT:PFC9	0	0	PFC8	00001000	bbrbbrrb
E637 H.S.	1	EP8FIFOPFL[7]	Endpoint 8 / slave FIFO Programmable Flag L	PFC7	PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
E637 F.S	1	EP8FIFOPFL[7]	Endpoint 8 / slave FIFO Programmable Flag L	IN: PKTS[1] OUT:PFC7	IN: PKTS[0] OUT:PFC6	PFC5	PFC4	PFC3	PFC2	PFC1	PFC0	00000000	RW
	8	reserved											
E640	1	EP2ISOINPKTS	EP2 (if ISO) IN Packets per frame (1-3)	AADJ	0	0	0	0	0	INPPF1	INPPF0	00000001	brrrrrbb
E641	1	EP4ISOINPKTS	EP4 (if ISO) IN Packets per frame (1-3)	AADJ	0	0	0	0	0	INPPF1	INPPF0	00000001	brrrrrrr
E642	1	EP6ISOINPKTS	EP6 (if ISO) IN Packets per frame (1-3)	AADJ	0	0	0	0	0	INPPF1	INPPF0	00000001	brrrrrbb
E643	1	EP8ISOINPKTS	EP8 (if ISO) IN Packets per frame (1-3)	AADJ	0	0	0	0	0	INPPF1	INPPF0	00000001	brrrrrrr
E644	4	reserved											
E648	1	INPKTEND[7]	Force IN Packet End	Skip	0	0	0	EP3	EP2	EP1	EP0	xxxxxxxx	W
E649	7	OUTPKTEND[7]	Force OUT Packet End	Skip	0	0	0	EP3	EP2	EP1	EP0	xxxxxxxx	W
		INTERRUPTS											
E650	1	EP2FIFOIE[7]	Endpoint 2 slave FIFO Flag Interrupt Enable	0	0	0	0	EDGEPF	PF	EF	FF	00000000	RW
E651	1	EP2FIFOIRQ[7,8]	Endpoint 2 slave FIFO Flag Interrupt Request	0	0	0	0	0	PF	EF	FF	00000000	rrrrrbbb
E652	1	EP4FIFOIE[7]	Endpoint 4 slave FIFO Flag Interrupt Enable	0	0	0	0	EDGEPF	PF	EF	FF	00000000	RW
E653	1	EP4FIFOIRQ[7,8]	Endpoint 4 slave FIFO Flag Interrupt Request	0	0	0	0	0	PF	EF	FF	00000000	rrrrrbbb
E654	1	EP6FIFOIE[7]	Endpoint 6 slave FIFO Flag Interrupt Enable	0	0	0	0	EDGEPF	PF	EF	FF	00000000	RW
E655	1	EP6FIFOIRQ[7,8]	Endpoint 6 slave FIFO Flag Interrupt Request	0	0	0	0	0	PF	EF	FF	00000000	rrrrrbbb
E656	1	EP8FIFOIE[7]	Endpoint 8 slave FIFO Flag Interrupt Enable	0	0	0	0	EDGEPF	PF	EF	FF	00000000	RW
E657	1	EP8FIFOIRQ[7,8]	Endpoint 8 slave FIFO Flag Interrupt Request	0	0	0	0	0	PF	EF	FF	00000000	rrrrrbbb
E658	1	IBNIE	IN-BULK-NAK Interrupt Enable	0	0	EP8	EP6	EP4	EP2	EP1	EP0	00000000	RW
E659	1	IBNIRQ[8]	IN-BULK-NAK interrupt Request	0	0	EP8	EP6	EP4	EP2	EP1	EP0	00xxxxxx	rrbbbbbb
E65A	1	NAKIE	Endpoint Ping-NAK / IBN Interrupt Enable	EP8	EP6	EP4	EP2	EP1	EP0	0	IBN	00000000	RW
E65B	1	NAKIRQ[8]	Endpoint Ping-NAK / IBN Interrupt Request	EP8	EP6	EP4	EP2	EP1	EP0	0	IBN	xxxxxx0x	bbbbbbrb
E65C	1	USBIE	USB Int Enables	0	EP0ACK	HSGRANT	URES	SUSP	SUTOK	SOF	SUDAV	00000000	RW
E65D	1	USBIRQ[8]	USB Interrupt Requests	0	EP0ACK	HSGRANT	URES	SUSP	SUTOK	SOF	SUDAV	0xxxxxxx	rbbbbbbb
E65E	1	EPIE	Endpoint Interrupt Enables	EP8	EP6	EP4	EP2	EP1OUT	EP1IN	EP0OUT	EP0IN	00000000	RW
E65F	1	EPIRQ[8]	Endpoint Interrupt Requests	EP8	EP6	EP4	EP2	EP1OUT	EP1IN	EP0OUT	EP0IN	0	RW

Note:
8. The register can only be reset, it cannot be set.

Table 6-1. NX2LP-Flex Register Summary (continued)

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
E660	1	GPIFIE[7]	GPIF Interrupt Enable	0	0	0	0	0	0	GPIFWF	GPIFDONE	00000000	RW
E661	1	GPIFIRQ[7]	GPIF Interrupt Request	0	0	0	0	0	0	GPIFWF	GPIFDONE	000000xx	RW
E662	1	USBERRIE	USB Error Interrupt Enables	ISOEP8	ISOEP6	ISOEP4	ISOEP2	0	0	0	ERRLIMIT	00000000	RW
E663	1	USBERRIRQ[8]	USB Error Interrupt Requests	ISOEP8	ISOEP6	ISOEP4	ISOEP2	0	0	0	ERRLIMIT	0000000x	bbbbrrrb
E664	1	ERRCNTLIM	USB Error counter and limit	EC3	EC2	EC1	EC0	LIMIT3	LIMIT2	LIMIT1	LIMIT0	xxxx0100	rrrrbbbb
E665	1	CLRERRCNT	Clear Error Counter EC3:0	x	x	x	x	x	x	x	x	xxxxxxxx	W
E666	1	INT2IVEC	Interrupt 2 (USB) Autovector	0	I2V4	I2V3	I2V2	I2V1	I2V0	0	0	00000000	R
E667	1	INT4IVEC	Interrupt 4 (slave FIFO & GPIF) Autovector	1	0	I4V3	I4V2	I4V1	I4V0	0	0	10000000	R
E668	1	INTSET-UP	Interrupt 2&4 set-up	0	0	0	0	AV2EN	0	INT4SRC	AV4EN	00000000	RW
E669	7	reserved											
		INPUT / OUTPUT											
E670	1	PORTACFG	I/O PORTA Alternate Configuration	FLAGD	SLCS	0	0	0	0	INT1	INT0	00000000	RW
E671	1	PORTCCFG	I/O PORTC Alternate Configuration	GPIFA7	GPIFA6	GPIFA5	GPIFA4	GPIFA3	GPIFA2	GPIFA1	GPIFA0	00000000	RW
E672	1	PORTECFG	I/O PORTE Alternate Configuration	GPIFA8	T2EX	INT6	RXD1OUT	RXD0OUT	T2OUT	T1OUT	T0OUT	00000000	RW
E673	4	XTALINSRC	XTALIN Clock Source	0	0	0	0	0	0	0	EXTCLK	00000000	rrrrrrrb
E677	1	reserved											
E678	1	I2CS	I^2C Bus Control & Status	START	STOP	LASTRD	ID1	ID0	BERR	ACK	DONE	000xx000	bbbrrrrr
E679	1	I2DAT	I^2C Bus Data	d7	d6	d5	d4	d3	d2	d1	d0	xxxxxxxx	RW
E67A	1	I2CTL	I^2C Bus Control	0	0	0	0	0	0	STOPIE	400kHz	00000000	RW
E67B	1	XAUTODAT1	Autoptr1 MOVX access, when APTREN=1	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
E67C	1	XAUTODAT2	Autoptr2 MOVX access, when APTREN=1	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
		UDMA CRC											
E67D	1	UDMACRCH[7]	UDMA CRC MSB	CRC15	CRC14	CRC13	CRC12	CRC11	CRC10	CRC9	CRC8	01001010	RW
E67E	1	UDMACRCL[7]	UDMA CRC LSB	CRC7	CRC6	CRC5	CRC4	CRC3	CRC2	CRC1	CRC0	10111010	RW
E67F	1	UDMACRC-QUALIFIER	UDMA CRC Qualifier	QENABLE	0	0	0	QSTATE	QSIGNAL2	QSIGNAL1	QSIGNAL0	00000000	brrrbbbb
		USB CONTROL											
E680	1	USBCS	USB Control & Status	HSM	0	0	0	DISCON	NOSYNSOF	RENUM	SIGRSUME	x0000000	rrrrbbbb
E681	1	SUSPEND	Put chip into suspend	x	x	x	x	x	x	x	x	xxxxxxxx	W
E682	1	WAKEUPCS	Wakeup Control & Status	WU2	WU	WU2POL	WUPOL	0	DPEN	WU2EN	WUEN	xx000101	bbbbrbbb
E683	1	TOGCTL	Toggle Control	Q	S	R	IO	EP3	EP2	EP1	EP0	x0000000	rrrbbbbb
E684	1	USBFRAMEH	USB Frame count H	0	0	0	0	0	FC10	FC9	FC8	00000xxx	R
E685	1	USBFRAMEL	USB Frame count L	FC7	FC6	FC5	FC4	FC3	FC2	FC1	FC0	xxxxxxxx	R
E686	1	MICROFRAME	Microframe count, 0-7	0	0	0	0	0	MF2	MF1	MF0	00000xxx	R
E687	1	FNADDR	USB Function address	0	FA6	FA5	FA4	FA3	FA2	FA1	FA0	0xxxxxxx	R
E688	2	reserved											
		ENDPOINTS											
E68A	1	EP0BCH[7]	Endpoint 0 Byte Count H	(BC15)	(BC14)	(BC13)	(BC12)	(BC11)	(BC10)	(BC9)	(BC8)	xxxxxxxx	RW
E68B	1	EP0BCL[7]	Endpoint 0 Byte Count L	(BC7)	BC6	BC5	BC4	BC3	BC2	BC1	BC0	xxxxxxxx	RW
E68C	1	reserved											
E68D	1	EP1OUTBC	Endpoint 1 OUT Byte Count	0	BC6	BC5	BC4	BC3	BC2	BC1	BC0	0xxxxxxx	RW
E68E	1	reserved											
E68F	1	EP1INBC	Endpoint 1 IN Byte Count	0	BC6	BC5	BC4	BC3	BC2	BC1	BC0	0xxxxxxx	RW
E690	1	EP2BCH[7]	Endpoint 2 Byte Count H	0	0	0	0	0	BC10	BC9	BC8	00000xxx	RW
E691	1	EP2BCL[7]	Endpoint 2 Byte Count L	BC7/SKIP	BC6	BC5	BC4	BC3	BC2	BC1	BC0	xxxxxxxx	RW
E692	2	reserved											
E694	1	EP4BCH[7]	Endpoint 4 Byte Count H	0	0	0	0	0	0	BC9	BC8	000000xx	RW
E695	1	EP4BCL[7]	Endpoint 4 Byte Count L	BC7/SKIP	BC6	BC5	BC4	BC3	BC2	BC1	BC0	xxxxxxxx	RW
E696	2	reserved											
E698	1	EP6BCH[7]	Endpoint 6 Byte Count H	0	0	0	0	0	BC10	BC9	BC8	00000xxx	RW
E699	1	EP6BCL[7]	Endpoint 6 Byte Count L	BC7/SKIP	BC6	BC5	BC4	BC3	BC2	BC1	BC0	xxxxxxxx	RW
E69A	2	reserved											
E69C	1	EP8BCH[7]	Endpoint 8 Byte Count H	0	0	0	0	0	0	BC9	BC8	000000xx	RW
E69D	1	EP8BCL[7]	Endpoint 8 Byte Count L	BC7/SKIP	BC6	BC5	BC4	BC3	BC2	BC1	BC0	xxxxxxxx	RW
E69E	2	reserved											
E6A0	1	EP0CS	Endpoint 0 Control and Status	HSNAK	0	0	0	0	0	BUSY	STALL	10000000	bbbbbbrb
E6A1	1	EP1OUTCS	Endpoint 1 OUT Control and Status	0	0	0	0	0	0	BUSY	STALL	00000000	bbbbbbrb
E6A2	1	EP1INCS	Endpoint 1 IN Control and Status	0	0	0	0	0	0	BUSY	STALL	00000000	bbbbbbrb
E6A3	1	EP2CS	Endpoint 2 Control and Status	0	NPAK2	NPAK1	NPAK0	FULL	EMPTY	0	STALL	00101000	rrrrrrrb

Table 6-1. NX2LP-Flex Register Summary (continued)

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
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	Set-up Data Pointer high address byte	A15	A14	A13	A12	A11	A10	A9	A8	xxxxxxxx	RW
E6B4	1	SUDPTRL	Set-up Data Pointer low address byte	A7	A6	A5	A4	A3	A2	A1	0	xxxxxxx0	bbbbbbbr
E6B5	1	SUDPTRCTL	Set-up Data Pointer Auto Mode	0	0	0	0	0	0	0	SDPAUTO	00000001	RW
	2	reserved											
E6B8	8	SET-UPDAT	8 bytes of set-up 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[7]	GPIF Address H	0	0	0	0	0	0	0	GPIFA8	00000000	RW
E6C5	1	GPIFADRL[7]	GPIF Address L	GPIFA7	GPIFA6	GPIFA5	GPIFA4	GPIFA3	GPIFA2	GPIFA1	GPIFA0	00000000	RW
		FLOWSTATE											
E6C6	1	FLOWSTATE	Flowstate Enable and Selector	FSE	0	0	0	0	FS2	FS1	FS0	00000000	brrrrbbb
E6C7	1	FLOWLOGIC	Flowstate Logic	LFUNC1	LFUNC0	TERMA2	TERMA1	TERMA0	TERMB2	TERMB1	TERMB0	00000000	RW
E6C8	1	FLOWEQ0CTL	CTL-Pin States in Flowstate (when Logic = 0)	CTL0E3	CTL0E2	CTL0E1/ CTL5	CTL0E0/ CTL4	CTL3	CTL2	CTL1	CTL0	00000000	RW
E6C9	1	FLOWEQ1CTL	CTL-Pin States in Flowstate (when Logic = 1)	CTL0E3	CTL0E2	CTL0E1/ CTL5	CTL0E0/ CTL4	CTL3	CTL2	CTL1	CTL0	00000000	RW
E6CA	1	FLOWHOLDOFF	Holdoff Configuration	HOPERIOD3	HOPERIOD2	HOPERIOD1	HOPERIOD 0	HOSTATE	HOCTL2	HOCTL1	HOCTL0	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[7]	GPIF Transaction Count Byte 3	TC31	TC30	TC29	TC28	TC27	TC26	TC25	TC24	00000000	RW
E6CF	1	GPIFTCB2[7]	GPIF Transaction Count Byte 2	TC23	TC22	TC21	TC20	TC19	TC18	TC17	TC16	00000000	RW
E6D0	1	GPIFTCB1[7]	GPIF Transaction Count Byte 1	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	00000000	RW
E6D1	1	GPIFTCB0[7]	GPIF Transaction Count Byte 0	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	00000001	RW
	2	reserved										00000000	RW

Table 6-1. NX2LP-Flex Register Summary (continued)

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
		reserved											
		reserved											
E6D2	1	EP2GPIFFLGSEL[7]	Endpoint 2 GPIF Flag select	0	0	0	0	0	0	FS1	FS0	00000000	RW
E6D3	1	EP2GPIFPFSTOP	Endpoint 2 GPIF stop transaction on prog. flag	0	0	0	0	0	0	0	FIFO2FLAG	00000000	RW
E6D4	1	EP2GPIFTRIG[7]	Endpoint 2 GPIF Trigger	x	x	x	x	x	x	x	x	xxxxxxxx	W
	3	reserved											
		reserved											
		reserved											
E6DA	1	EP4GPIFFLGSEL[7]	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[7]	Endpoint 4 GPIF Trigger	x	x	x	x	x	x	x	x	xxxxxxxx	W
	3	reserved											
		reserved											
		reserved											
E6E2	1	EP6GPIFFLGSEL[7]	Endpoint 6 GPIF Flag select	0	0	0	0	0	0	FS1	FS0	00000000	RW
E6E3	1	EP6GPIFPFSTOP	Endpoint 6 GPIF stop transaction on prog. flag	0	0	0	0	0	0	0	FIFO6FLAG	00000000	RW
E6E4	1	EP6GPIFTRIG[7]	Endpoint 6 GPIF Trigger	x	x	x	x	x	x	x	x	xxxxxxxx	W
	3	reserved											
		reserved											
		reserved											
E6EA	1	EP8GPIFFLGSEL[7]	Endpoint 8 GPIF Flag select	0	0	0	0	0	0	FS1	FS0	00000000	RW
E6EB	1	EP8GPIFPFSTOP	Endpoint 8 GPIF stop transaction on prog. flag	0	0	0	0	0	0	0	FIFO8FLAG	00000000	RW
E6EC	1	EP8GPIFTRIG[7]	Endpoint 8 GPIF Trigger	x	x	x	x	x	x	x	x	xxxxxxxx	W
	3	reserved											
E6F0	1	XGPIFSGLDATH	GPIF Data H (16-bit mode only)	D15	D14	D13	D12	D11	D10	D9	D8	xxxxxxxx	RW
E6F1	1	XGPIFSGLDATLX	Read/Write GPIF Data L & trigger transaction	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
E6F2	1	XGPIFSGLDATL-NOX	Read GPIF Data L, no transaction trigger	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	R
E6F3	1	GPIFREADYCFG	Internal RDY, Sync/Async, RDY pin states	INTRDY	SAS	TCXRDY5	0	0	0	0	0	00000000	bbbrrrrr
E6F4	1	GPIFREADYSTAT	GPIF Ready Status	0	0	RDY5	RDY4	RDY3	RDY2	RDY1	RDY0	00xxxxxx	R
E6F5	1	GPIFABORT	Abort GPIF Waveforms	x	x	x	x	x	x	x	x	xxxxxxxx	W
E6F6	2	reserved											
		ENDPOINT BUFFERS											
E740	64	EP0BUF	EP0-IN/-OUT buffer	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
E780	64	EP10UTBUF	EP1-OUT buffer	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
E7C0	64	EP1INBUF	EP1-IN buffer	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
	2048	reserved											RW
F000	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		I²C Configuration Byte		0	DISCON	0	0	0	0	0	400KHZ	xxxxxxxx [10]	n/a
		Special Function Registers (SFRs)											
80	1	IOA[9]	Port A (bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
81	1	SP	Stack Pointer	D7	D6	D5	D4	D3	D2	D1	D0	00000111	RW
82	1	DPL0	Data Pointer 0 L	A7	A6	A5	A4	A3	A2	A1	A0	00000000	RW
83	1	DPH0	Data Pointer 0 H	A15	A14	A13	A12	A11	A10	A9	A8	00000000	RW
84	1	DPL1[9]	Data Pointer 1 L	A7	A6	A5	A4	A3	A2	A1	A0	00000000	RW
85	1	DPH1[9]	Data Pointer 1 H	A15	A14	A13	A12	A11	A10	A9	A8	00000000	RW
86	1	DPS[9]	Data Pointer 0/1 select	0	0	0	0	0	0	0	SEL	00000000	RW
87	1	PCON	Power Control	SMOD0	x	1	1	x	x	x	IDLE	00110000	RW

Notes:

9. SFRs not part of the standard 8051 architecture.
10. If no NAND is detected by the SIE then the default is 00000000.

Table 6-1. NX2LP-Flex Register Summary (continued)

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
88	1	TCON	Timer/Counter Control (bit addressable)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000	RW
89	1	TMOD	Timer/Counter Mode Control	GATE	CT	M1	M0	GATE	CT	M1	M0	00000000	RW
8A	1	TL0	Timer 0 reload L	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
8B	1	TL1	Timer 1 reload L	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
8C	1	TH0	Timer 0 reload H	D15	D14	D13	D12	D11	D10	D9	D8	00000000	RW
8D	1	TH1	Timer 1 reload H	D15	D14	D13	D12	D11	D10	D9	D8	00000000	RW
8E	1	CKCON[9]	Clock Control	x	x	T2M	T1M	T0M	MD2	MD1	MD0	00000001	RW
8F	1	reserved											
90	1	IOB[9]	Port B (bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
91	1	EXIF[9]	External Interrupt Flag(s)	IE5	IE4	I²CINT	USBNT	1	0	0	0	00001000	RW
92	1	MPAGE[9]	Upper Addr Byte of MOVX using @R0 / @R1	A15	A14	A13	A12	A11	A10	A9	A8	00000000	RW
93	5	reserved											
98	1	SCON0	Serial Port 0 Control (bit addressable)	SM0_0	SM1_0	SM2_0	REN_0	TB8_0	RB8_0	TI_0	RI_0	00000000	RW
99	1	SBUF0	Serial Port 0 Data Buffer	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
9A	1	AUTOPTRH1[9]	Autopointer 1 Address H	A15	A14	A13	A12	A11	A10	A9	A8	00000000	RW
9B	1	AUTOPTRL1[9]	Autopointer 1 Address L	A7	A6	A5	A4	A3	A2	A1	A0	00000000	RW
9C	1	reserved											
9D	1	AUTOPTRH2[9]	Autopointer 2 Address H	A15	A14	A13	A12	A11	A10	A9	A8	00000000	RW
9E	1	AUTOPTRL2[9]	Autopointer 2 Address L	A7	A6	A5	A4	A3	A2	A1	A0	00000000	RW
9F	1	reserved											
A0	1	IOC[9]	Port C (bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
A1	1	INT2CLR[9]	Interrupt 2 clear	x	x	x	x	x	x	x	x	xxxxxxxx	W
A2	1	INT4CLR[9]	Interrupt 4 clear	x	x	x	x	x	x	x	x	xxxxxxxx	W
A3	5	reserved											
A8	1	IE	Interrupt Enable (bit addressable)	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0	00000000	RW
A9	1	reserved											
AA	1	EP2468STAT[9]	Endpoint 2,4,6,8 status flags	EP8F	EP8E	EP6F	EP6E	EP4F	EP4E	EP2F	EP2E	01011010	R
AB	1	EP24FIFOFLGS[9]	Endpoint 2,4 slave FIFO status flags	0	EP4PF	EP4EF	EP4FF	0	EP2PF	EP2EF	EP2FF	00100010	R
AC	1	EP68FIFOFLGS[9]	Endpoint 6,8 slave FIFO status flags	0	EP8PF	EP8EF	EP8FF	0	EP6PF	EP6EF	EP6FF	01100110	R
AD	2	reserved											
AF	1	AUTOPTRSET-UP[9]	Autopointer 1&2 set-up	0	0	0	0	0	APTR2INC	APTR1INC	APTREN	00000110	RW
B0	1	IOD[9]	Port D (bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
B1	1	IOE[9]	Port E (NOT bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
B2	1	OEA[9]	Port A Output Enable	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
B3	1	OEB[9]	Port B Output Enable	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
B4	1	OEC[9]	Port C Output Enable	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
B5	1	OED[9]	Port D Output Enable	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
B6	1	OEE[9]	Port E Output Enable	D7	D6	D5	D4	D3	D2	D1	D0	00000000	RW
B7	1	reserved											
B8	1	IP	Interrupt Priority (bit addressable)	1	PS1	PT2	PS0	PT1	PX1	PT0	PX0	10000000	RW
B9	1	reserved											
BA	1	EP01STAT[9]	Endpoint 0&1 Status	0	0	0	0	0	EP1INBSY	EP1OUTBSY	EP0BSY	00000000	R
BB	1	GPIFTRIG[9, 7]	Endpoint 2,4,6,8 GPIF slave FIFO Trigger	DONE	0	0	0	0	RW	EP1	EP0	10000xxx	brrrrbbb
BC	1	reserved											
BD	1	GPIFSGLDATH[9]	GPIF Data H (16-bit mode only)	D15	D14	D13	D12	D11	D10	D9	D8	xxxxxxxx	RW
BE	1	GPIFSGLDATLX[9]	GPIF Data L w/ Trigger	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
BF	1	GPIFSGLDATLNOX[9]	GPIF Data L w/ No Trigger	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	R
C0	1	SCON1[9]	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[9]	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											

Table 6-1. NX2LP-Flex Register Summary (continued)

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
D0	1	PSW	Program Status Word (bit addressable)	CY	AC	F0	RS1	RS0	OV	F1	P	00000000	RW
D1	7	reserved											
D8	1	EICON[9]	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[9]	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[9]	External Interrupt Priority Control	1	1	1	PX6	PX5	PX4	PI²C	PUSB	11100000	RW
F9	7	reserved											

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

7.0 Absolute Maximum Ratings

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

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

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

DC Input Voltage to Any Input Pin ..... +5.25V[11]

DC Voltage Applied to Outputs in High Z State ..... –0.5V to V_{CC} + 0.5V

Power Dissipation ..... 300 mW

Static Discharge Voltage ..... > 2000V

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

Note:
11. Applying power to I/O pins when the chip is not powered is not recommended.

8.0 Operating Conditions

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

Supply Voltage ..... +3.00V to +3.60V

Ground Voltage 0V

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

9.0 DC Characteristics

Table 9-1. DC Characteristics

Parameter	Description	Conditions	Min.	Typ.	Max.	Unit
V_{CC}	Supply Voltage		3.00	3.3	3.60	V
V_{CC} Ramp Up	0 to 3.3V		200			µs
V_{IH}	Input HIGH Voltage		2		5.25	V
V_{IL}	Input LOW Voltage		–0.5		0.8	V
V_{IH_X}	Crystal Input HIGH Voltage		2		5.25	V
V_{IL_X}	Crystal Input LOW Voltage		–0.5		0.8	V
I_I	Input Leakage Current	$0< V_{IN} < V_{CC}$			±10	µA
V_{OH}	Output Voltage HIGH	I_{OUT} = 4 mA	2.4			V
V_{OL}	Output LOW Voltage	I_{OUT} = –4 mA			0.4	V
I_{OH}	Output Current HIGH				4	mA
I_{OL}	Output Current LOW				4	mA
C_{IN}	Input Pin Capacitance	Except D+/D–			10	pF
		D+/D–			15	pF
I_{SUSP}	Suspend Current	Connected		300	380[12]	µA
	CY7C68034	Disconnected		100	150[12]	µA
	Suspend Current	Connected		0.5	1.2[12]	mA
	CY7C68033	Disconnected		0.3	1.0[12]	mA
I_{CC}	Supply Current	8051 running, connected to USB HS		43		mA
		8051 running, connected to USB FS		35		mA
$I_{UNCONFIG}$	Unconfigured Current	Before bMaxPower granted by host		43		mA
T_{RESET}	Reset Time After Valid Power	V_{CC} min = 3.0V	5.0			ms
	Pin Reset After powered on		200			µs

9.1 USB Transceiver

USB 2.0-compliant in full- and high-speed modes.

10.0 AC Electrical Characteristics

10.1 USB Transceiver

USB 2.0-compliant in full- and high-speed modes.

Note:
12. Measured at Max V_{CC}, 25°C.

10.2 GPIF Synchronous Signals

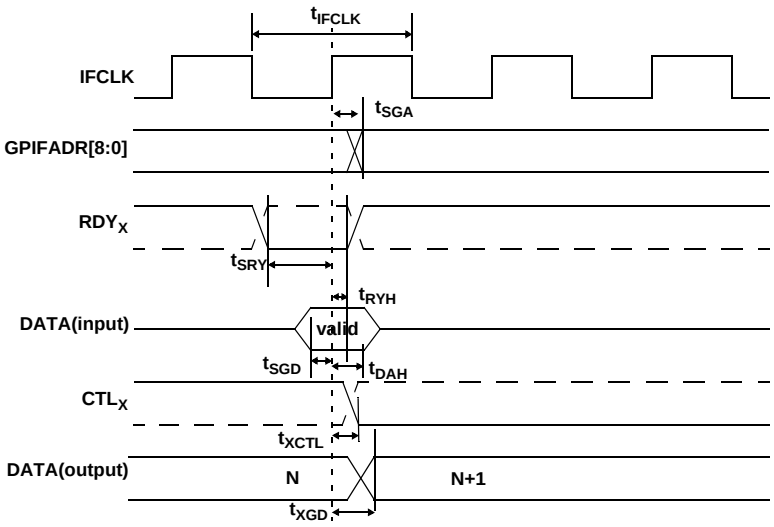


Figure 10-1. GPIF Synchronous Signals Timing Diagram[13]

Table 10-1. GPIF Synchronous Signals Parameters with Internally Sourced IFCLK[13, 14]

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

Table 10-2. GPIF Synchronous Signals Parameters with Externally Sourced IFCLK[14]

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

Notes:
13. Dashed lines denote signals with programmable polarity.
14. GPIF asynchronous RDY_x signals have a minimum Set-up time of 50 ns when using internal 48-MHz IFCLK.
15. IFCLK must not exceed 48 MHz.

10.3 Slave FIFO Synchronous Read

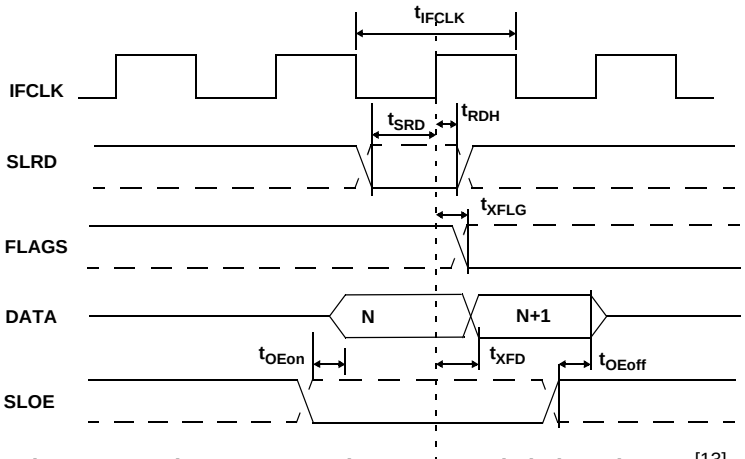


Figure 10-2. Slave FIFO Synchronous Read Timing Diagram[13]

Table 10-3. Slave FIFO Synchronous Read Parameters with Internally Sourced IFCLK[14]

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

Table 10-4. Slave FIFO Synchronous Read Parameters with Externally Sourced IFCLK[14]

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

10.4 Slave FIFO Asynchronous Read

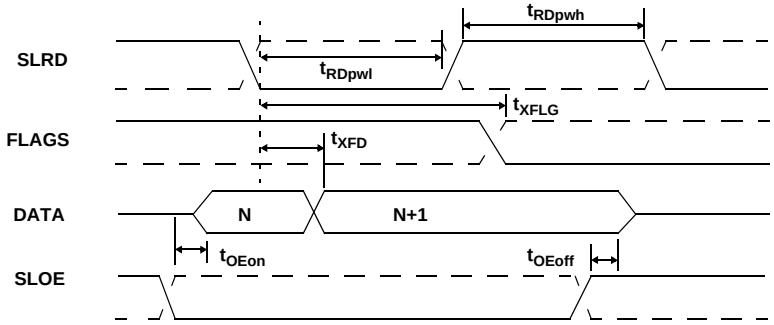


Figure 10-3. Slave FIFO Asynchronous Read Timing Diagram[13]

Table 10-5. Slave FIFO Asynchronous Read Parameters[16]

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

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

10.5 Slave FIFO Synchronous Write

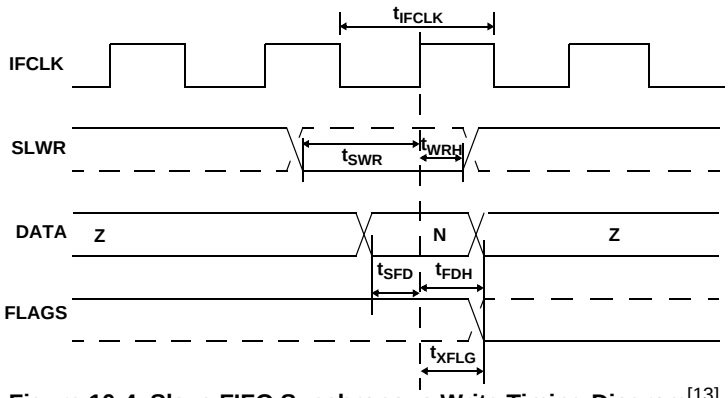


Figure 10-4. Slave FIFO Synchronous Write Timing Diagram[13]

Table 10-6. Slave FIFO Synchronous Write Parameters with Internally Sourced IFCLK[14]

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

Table 10-7. Slave FIFO Synchronous Write Parameters with Externally Sourced IFCLK[14]

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

10.6 Slave FIFO Asynchronous Write

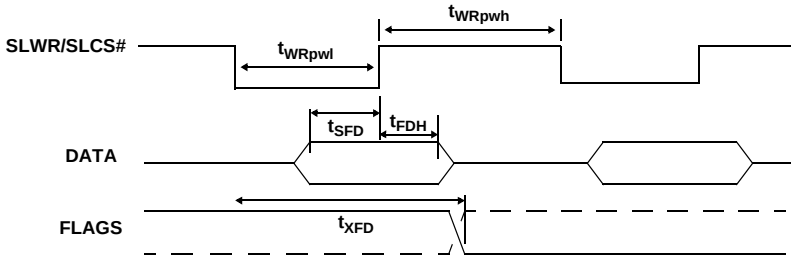


Figure 10-5. Slave FIFO Asynchronous Write Timing Diagram[13]

Table 10-8. Slave FIFO Asynchronous Write Parameters with Internally Sourced IFCLK [16]

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

10.7 Slave FIFO Synchronous Packet End Strobe

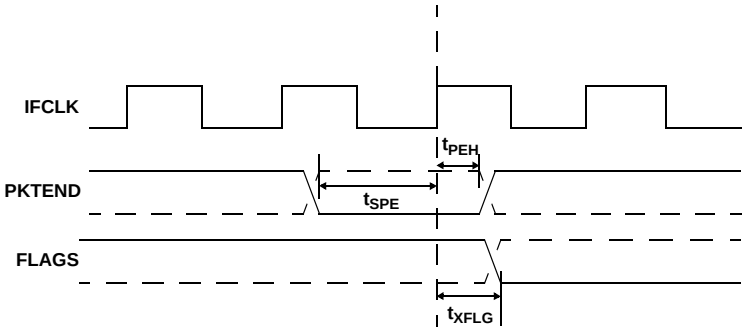


Figure 10-6. Slave FIFO Synchronous Packet End Strobe Timing Diagram[13]

Table 10-9. Slave FIFO Synchronous Packet End Strobe Parameters with Internally Sourced IFCLK [14]

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

Table 10-10. Slave FIFO Synchronous Packet End Strobe Parameters with Externally Sourced IFCLK [14]

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

There is no specific timing requirement that needs to be met for asserting PKTEND pin with regards to asserting SLWR. PKTEND can be asserted with the last data value clocked into the FIFOs or thereafter. The only consideration is the set-up 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 that needs to be met when the FIFO is configured to operate

in auto mode and it is desired to send two packets back to back: a full packet (full defined as the number of bytes in the FIFO meeting the level set in AUTOINLEN register) committed automatically followed by a short one byte/word packet committed manually using the PKTEND pin. In this particular scenario, user must make sure to assert PKTEND at least one clock cycle after the rising edge that caused the last byte/word to be clocked into the previous auto committed packet. *Figure 10-7* below shows this scenario. X is the value the AUTOINLEN register is set to when the IN endpoint is configured to be in auto mode.

Figure 10-7 shows a scenario where two packets are being committed. The first packet gets committed automatically when the number of bytes in the FIFO reaches X (value set in AUTOINLEN register) and the second one byte/word short packet being committed manually using PKTEND. Note that there is at least one IFCLK cycle timing between the assertion of PKTEND and clocking of the last byte of the previous packet (causing the packet to be committed automatically). Not adhering to this timing will result in the NX2LP-Flex failing to send the one byte/word short packet.

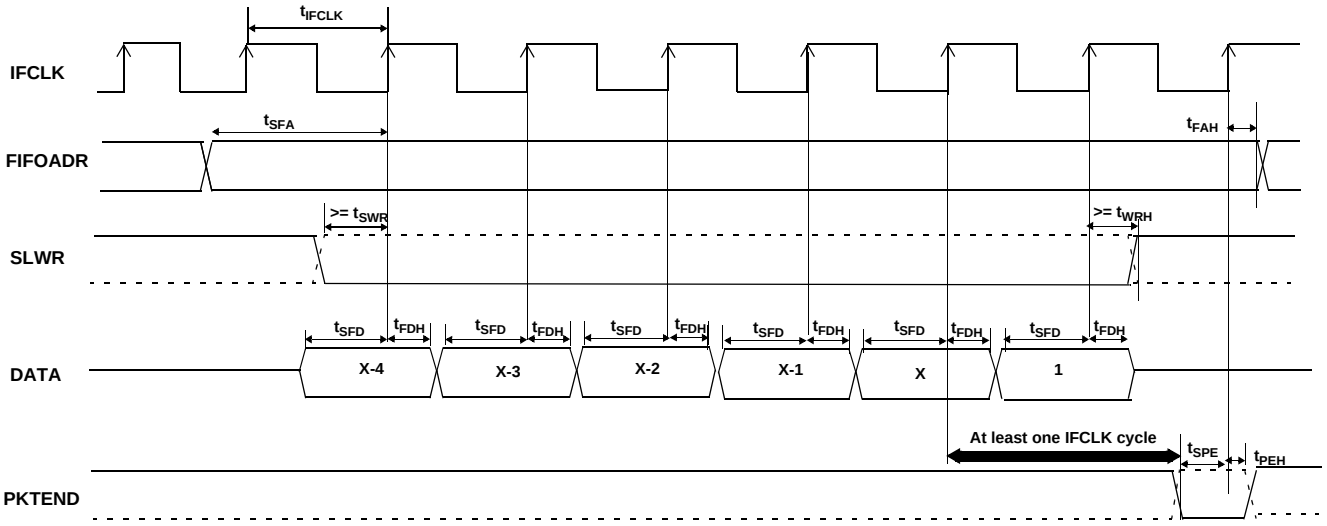


Figure 10-7. Slave FIFO Synchronous Write Sequence and Timing Diagram[13]

10.8 Slave FIFO Asynchronous Packet End Strobe

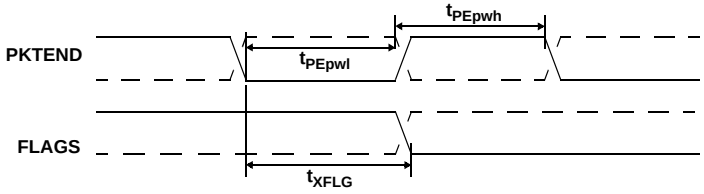


Figure 10-8. Slave FIFO Asynchronous Packet End Strobe Timing Diagram[13]

Table 10-11. Slave FIFO Asynchronous Packet End Strobe Parameters[16]

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

10.9 Slave FIFO Output Enable

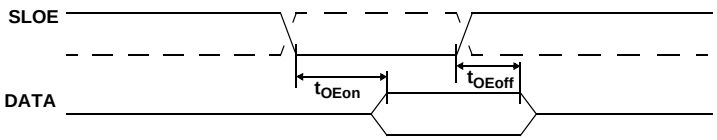


Figure 10-9. Slave FIFO Output Enable Timing Diagram[13]

Table 10-12. Slave FIFO Output Enable Parameters

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

10.10 Slave FIFO Address to Flags/Data

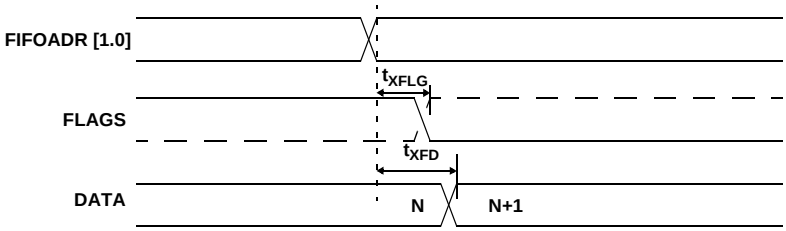


Figure 10-10. Slave FIFO Address to Flags/Data Timing Diagram[13]

Table 10-13. Slave FIFO Address to Flags/Data Parameters

Parameter	Description	Min.	Max.	Unit
t_{XFLG}	FIFOADR[1:0] to FLAGS Output Propagation Delay		10.7	ns
t_{XFD}	FIFOADR[1:0] to FIFODATA Output Propagation Delay		14.3	ns

10.11 Slave FIFO Synchronous Address

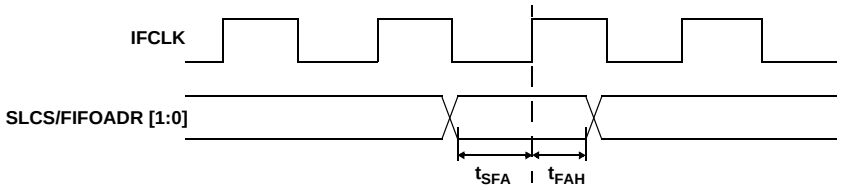


Figure 10-11. Slave FIFO Synchronous Address Timing Diagram[13]

Table 10-14. Slave FIFO Synchronous Address Parameters[14]

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

10.12 Slave FIFO Asynchronous Address

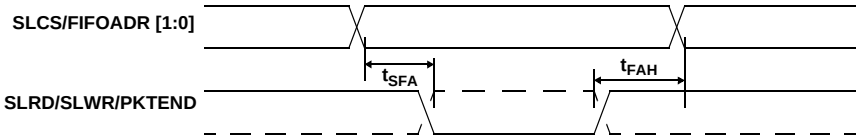


Figure 10-12. Slave FIFO Asynchronous Address Timing Diagram[13]

Table 10-15. Slave FIFO Asynchronous Address Parameters[16]

Parameter	Description	Min.	Max.	Unit
t_{SFA}	FIFOADR[1:0] to SLRD/SLWR/PKTEND Set-up Time	10		ns
t_{FAH}	RD/WR/PKTEND to FIFOADR[1:0] Hold Time	10		ns

10.13 Sequence Diagram

10.13.1 Single and Burst Synchronous Read Example

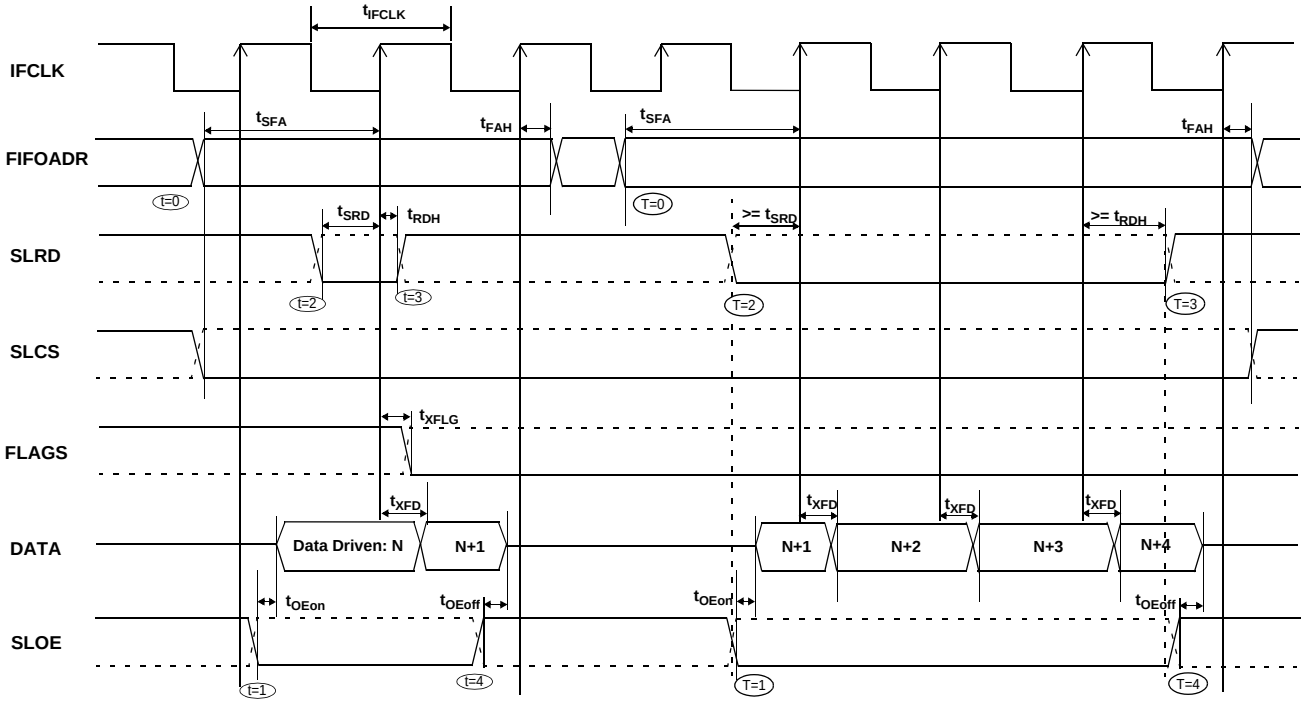


Figure 10-13. Slave FIFO Synchronous Read Sequence and Timing Diagram[13]

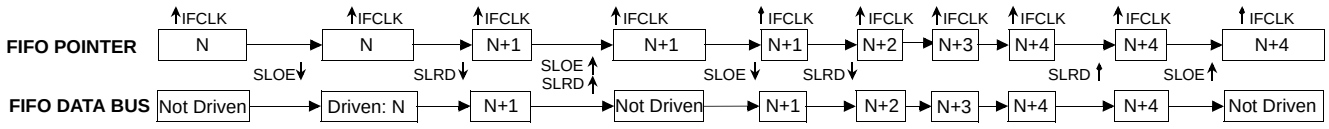


Figure 10-14. Slave FIFO Synchronous Sequence of Events Diagram

Figure 10-13 shows the timing relationship of the SLAVE FIFO signals during a synchronous FIFO read using IFCLK as the synchronizing clock. The diagram illustrates a single read followed by a burst read.

- At t = 0 the FIFO address is stable and the signal SLCS is asserted (SLCS may be tied low in some applications). **Note:** t_{SFA} has a minimum of 25 ns. This means when IFCLK is running at 48 MHz, the FIFO address set-up time is more than one IFCLK cycle.
- At t = 1, SLOE is asserted. SLOE is an output enable only, whose sole function is to drive the data bus. The data that is driven on the bus is the data that the internal FIFO pointer is currently pointing to. In this example it is the first data value in the FIFO. **Note:** the data is pre-fetched and is driven on the bus when SLOE is asserted.
- At t = 2, SLRD is asserted. SLRD must meet the set-up time of t_{SRD} (time from asserting the SLRD signal to the rising edge of the IFCLK) and maintain a minimum hold time of t_{RDH} (time from the IFCLK edge to the deassertion of the SLRD signal). If the SLCS signal is used, it must be asserted with SLRD, or before SLRD is asserted (i.e., the SLCS and SLRD signals must both be asserted to start a valid read condition).
- The FIFO pointer is updated on the rising edge of the IFCLK, while SLRD is asserted. This starts the propagation of data from the newly addressed location to the data bus. After a propagation delay of t_{XFD} (measured from the rising edge of IFCLK) the new data value is present. N is the first data value read from the FIFO. In order to have data on the FIFO data bus, SLOE MUST also be asserted.

The same sequence of events are shown for a burst read and are marked with the time indicators of T = 0 through 5. **Note:** For the burst mode, the SLRD and SLOE are left asserted during the entire duration of the read. In the burst read mode, when SLOE is asserted, data indexed by the FIFO pointer is on the data bus. During the first read cycle, on the rising edge of the clock the FIFO pointer is updated and increments to point to address N+1. For each subsequent rising edge of IFCLK, while the SLRD is asserted, the FIFO pointer is incremented and the next data value is placed on the data bus.

10.13.2 Single and Burst Synchronous Write

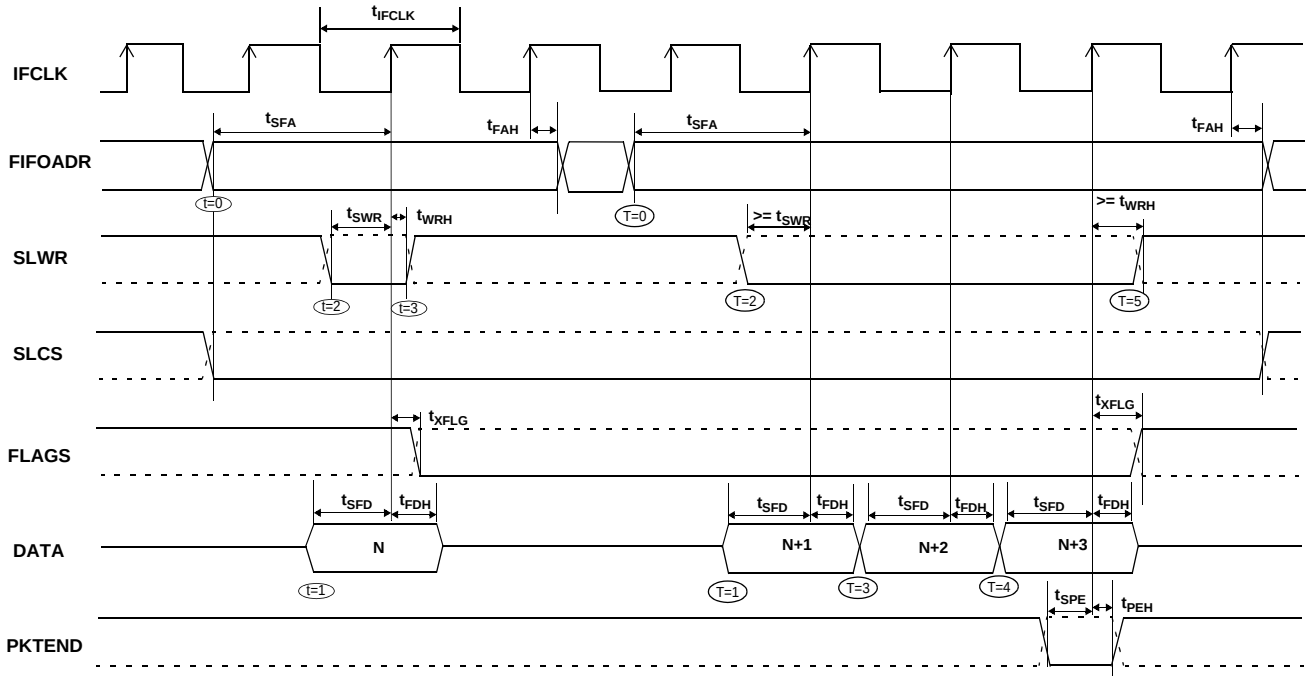


Figure 10-15. Slave FIFO Synchronous Write Sequence and Timing Diagram[13]

The *Figure 10-15* shows the timing relationship of the SLAVE FIFO signals during a synchronous write using IFCLK as the synchronizing clock. The diagram illustrates a single write followed by burst write of 3 bytes and committing all 4 bytes as a short packet using the PKTEND pin.

- At t = 0 the FIFO address is stable and the signal SLCS is asserted. (SLCS may be tied low in some applications.) **Note:** t_{SFA} has a minimum of 25 ns. This means when IFCLK is running at 48 MHz, the FIFO address set-up time is more than one IFCLK cycle.
- At t = 1, the external master/peripheral must output the data value onto the data bus with a minimum set up time of t_{SFD} before the rising edge of IFCLK.
- At t = 2, SLWR is asserted. The SLWR must meet the set-up time of t_{SWR} (time from asserting the SLWR signal to the rising edge of IFCLK) and maintain a minimum hold time of t_{WRH} (time from the IFCLK edge to the deassertion of the SLWR signal). If SLCS signal is used, it must be asserted with SLWR or before SLWR is asserted (i.e., the SLCS and SLWR signals must both be asserted to start a valid write condition).
- While the SLWR is asserted, data is written to the FIFO and on the rising edge of the IFCLK, the FIFO pointer is incremented. The FIFO flag will also be updated after a delay of t_{XFLG} from the rising edge of the clock.

The same sequence of events are also shown for a burst write and are marked with the time indicators of T = 0 through 5. **Note:** For the burst mode, SLWR and SLCS are left asserted for the entire duration of writing all the required data values. In this burst write mode, once the SLWR is asserted, the data on the FIFO data bus is written to the FIFO on every rising edge of IFCLK. The FIFO pointer is updated on each rising edge of IFCLK. In *Figure 10-15*, once the four bytes are written to the FIFO, SLWR is deasserted. The short 4-byte packet can be committed to the host by asserting the PKTEND signal.

There is no specific timing requirement that needs to be met for asserting the PKTEND signal with regards to asserting the SLWR signal. PKTEND can be asserted with the last data value or thereafter. The only requirement is that the set-up time t_{SPE} and the hold time t_{PEH} must be met. In the scenario of *Figure 10-15*, the number of data values committed includes the last value written to the FIFO. In this example, both the data value and the PKTEND signal are clocked on the same rising edge of IFCLK. PKTEND can also be asserted in subsequent clock cycles. The FIFOADDR lines should be held constant during the PKTEND assertion.

Although there are no specific timing requirement for the PKTEND assertion, there is a specific corner case condition that needs attention while using the PKTEND to commit a one byte/word packet. Additional timing requirements exists when the FIFO is configured to operate in auto mode and it is desired to send two packets: a full packet (full defined as the number of bytes in the FIFO meeting the level set in AUTOINLEN register) committed automatically followed by a short one byte/word packet committed manually using the PKTEND pin. In this case, the external master must make sure to assert the PKTEND pin at least one clock cycle after the rising edge that caused the last byte/word to be clocked into the previous auto committed packet (the packet with the number of bytes equal to what is set in the AUTOINLEN register). Refer to *Figure 10-7* for further details on this timing.

10.13.3 Sequence Diagram of a Single and Burst Asynchronous Read

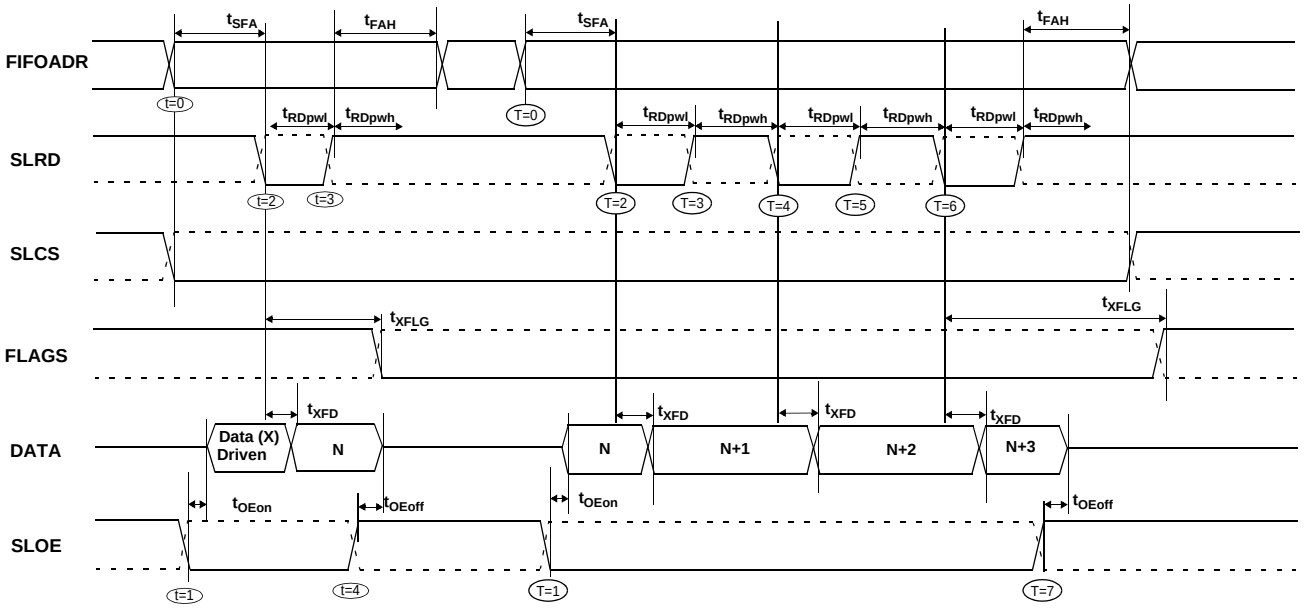


Figure 10-16. Slave FIFO Asynchronous Read Sequence and Timing Diagram[13]

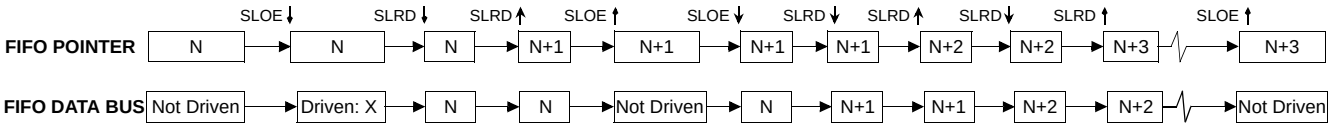


Figure 10-17. Slave FIFO Asynchronous Read Sequence of Events Diagram

Figure 10-16 diagrams the timing relationship of the SLAVE FIFO signals during an asynchronous FIFO read. It shows a single read followed by a burst read.

- At t = 0 the FIFO address is stable and the SLCS signal is asserted.
- At t = 1, SLOE is asserted. This results in the data bus being driven. The data that is driven on to the bus is previous data, it data that was in the FIFO from a prior read cycle.
- At t = 2, SLRD is asserted. The SLRD must meet the minimum active pulse of t_{RDpwl} and minimum de-active pulse width of t_{RDpwh}. If SLCS is used then, SLCS must be in asserted with SLRD or before SLRD is asserted (i.e., the SLCS and SLRD signals must both be asserted to start a valid read condition).
- The data that will be driven, after asserting SLRD, is the updated data from the FIFO. This data is valid after a propagation delay of t_{XFD} from the activating edge of SLRD. In *Figure 10-16*, data N is the first valid data read from the FIFO. For data to appear on the data bus during the read cycle (i.e.,SLRD is asserted), SLOE MUST be in an asserted state. SLRD and SLOE can also be tied together.

The same sequence of events is also shown for a burst read marked with T = 0 through 5. **Note:** In burst read mode, during SLOE is assertion, the data bus is in a driven state and outputs the previous data. Once SLRD is asserted, the data from the FIFO is driven on the data bus (SLOE must also be asserted) and then the FIFO pointer is incremented.

10.13.4 Sequence Diagram of a Single and Burst Asynchronous Write

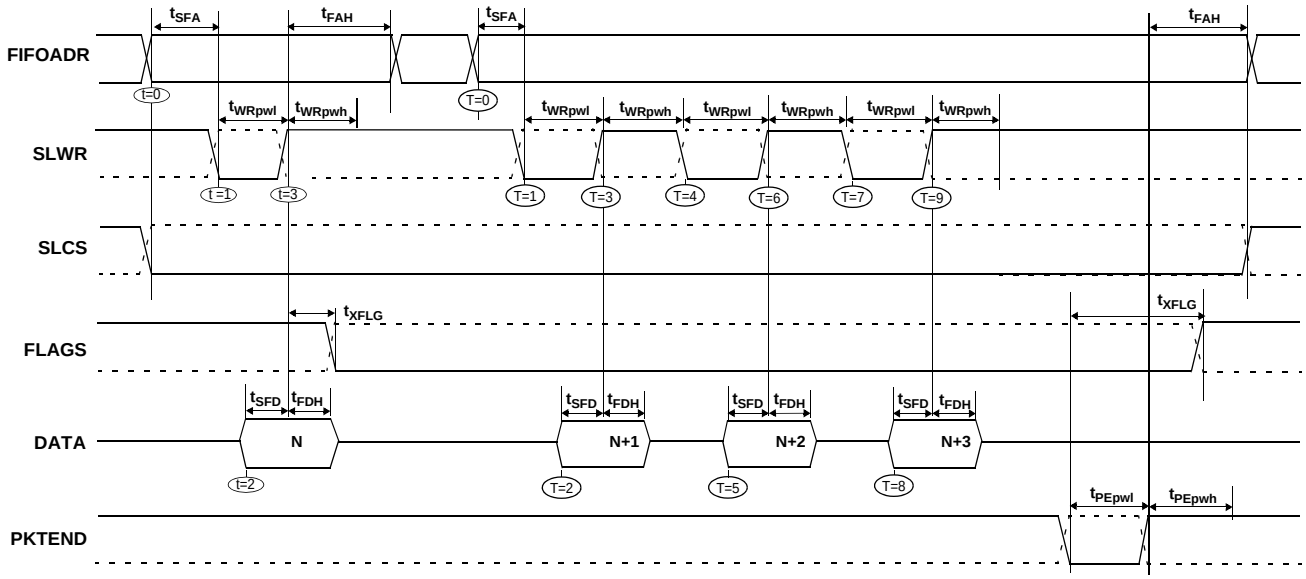


Figure 10-18. Slave FIFO Asynchronous Write Sequence and Timing Diagram[13]

Figure 10-18 diagrams the timing relationship of the SLAVE FIFO write in an asynchronous mode. The diagram shows a single write followed by a burst write of 3 bytes and committing the 4-byte-short packet using PKTEND.

- At t = 0 the FIFO address is applied, insuring that it meets the set-up time of t_{SFA}. If SLCS is used, it must also be asserted (SLCS may be tied low in some applications).
- At t = 1 SLWR is asserted. SLWR must meet the minimum active pulse of t_{WRpwl} and minimum de-active pulse width of t_{WRpwh}. If the SLCS is used, it must be in asserted with SLWR or before SLWR is asserted.
- At t = 2, data must be present on the bus t_{SFD} before the deasserting edge of SLWR.
- At t = 3, deasserting SLWR will cause the data to be written from the data bus to the FIFO and then increments the FIFO pointer. The FIFO flag is also updated after t_{XFLG} from the deasserting edge of SLWR.

The same sequence of events are shown for a burst write and is indicated by the timing marks of T = 0 through 5. **Note:** In the burst write mode, once SLWR is deasserted, the data is written to the FIFO and then the FIFO pointer is incremented to the next byte in the FIFO. The FIFO pointer is post incremented.

In *Figure 10-18* once the four bytes are written to the FIFO and SLWR is deasserted, the short 4-byte packet can be committed to the host using the PKTEND. The external device should be designed to not assert SLWR and the PKTEND signal at the same time. It should be designed to assert the PKTEND after SLWR is deasserted and met the minimum deasserted pulse width. The FIFOADDR lines are to be held constant during the PKTEND assertion.

11.0 Ordering Information

Table 11-1. Ordering Information

Ordering Code	Description
Silicon for battery-powered applications	
CY7C68034-56LFXC	8x8 mm, 56 QFN – Lead-free
Silicon for non-battery-powered applications	
CY7C68033-56LFXC	8x8 mm, 56 QFN – Lead-free
Development Kit	
CY3686	EZ-USB NX2LP-Flex Development Kit

12.0 Package Diagrams

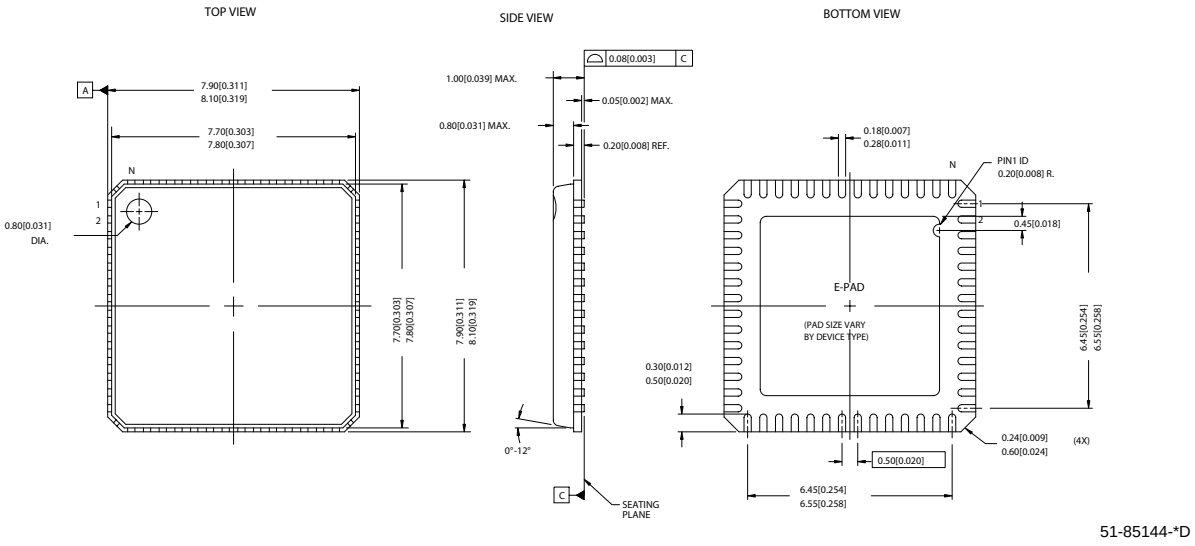


Figure 12-1. 56-Lead QFN 8 x 8 mm LF56A

13.0 PCB Layout Recommendations[17]

The following recommendations should be followed to ensure reliable high-performance operation:

- At least a four-layer impedance controlled boards is recommended to maintain signal quality.
- Specify impedance targets (ask your board vendor what they can achieve) to meet USB specifications.
- To control impedance, maintain trace widths and trace spacing.
- Minimize any stubs to avoid reflected signals.
- Connections between the USB connector shell and signal ground must be done near the USB connector.
- Bypass/flyback caps on VBUS, near connector, are recommended.
- DPLUS and DMINUS trace lengths should be kept to within 2 mm of each other in length, with preferred length of 20–30 mm.
- Maintain a solid ground plane under the DPLUS and DMINUS traces. Do not allow the plane to be split under these traces.
- No vias should be placed on the DPLUS or DMINUS trace routing unless absolutely necessary.
- Isolate the DPLUS and DMINUS traces from all other signal traces as much as possible.

Note:

17. Source for recommendations: *EZ-USB FX2™PCB Design Recommendations*, http://www.cypress.com/cfuploads/support/app_notes/FX2_PCB.pdf and *High Speed USB Platform Design Guidelines*, http://www.usb.org/developers/docs/hs_usb_pdg_r1_0.pdf.

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

Electrical contact of the part to the Printed Circuit Board (PCB) is made by soldering the leads on the bottom surface of the package to the PCB. Hence, special attention is required to the heat transfer area below the package to provide a good thermal bond to the circuit board. A Copper (Cu) fill is to be designed into the PCB as a thermal pad under the package. Heat is transferred from the NX2LP-Flex to the PCB through the device's metal paddle on the bottom side of the package. It is then conducted from the PCB's thermal pad to the inner ground plane by a 5 x 5 array of vias. A via is a plated through hole in the PCB with a finished diameter of 13 mil. The QFN's metal die paddle must be soldered to the PCB's thermal pad. Solder mask is placed on the board top side over each via to resist solder flow into the via. The mask on the top side also minimizes outgassing during the solder reflow process.

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

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

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

Figure 14-1 below displays a cross-sectional area underneath the package. The cross section is of only one via. The solder paste template needs to be designed to allow at least 50% solder coverage. The thickness of the solder paste template should be 5 mil. It is recommended that "No Clean" type 3 solder paste is used for mounting the part. Nitrogen purge is recommended during reflow.

Figure 14-2 is a plot of the solder mask pattern and *Figure 14-3* displays an X-Ray image of the assembly (darker areas indicate solder).

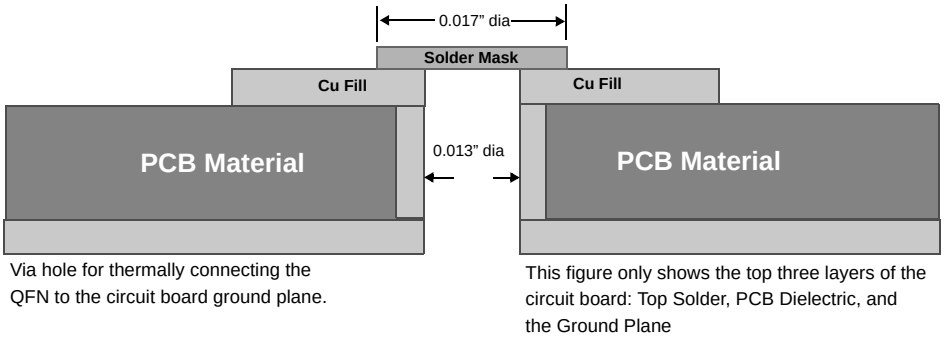


Figure 14-1. Cross-section of the Area Underneath the QFN Package

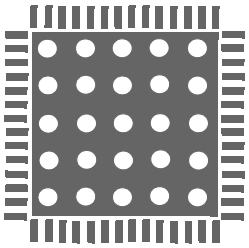


Figure 14-2. Plot of the Solder Mask (White Area)

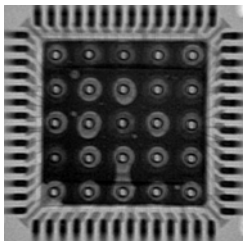


Figure 14-3. X-ray Image of the Assembly

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

Document History Page

Document Title: CY7C68033/CY7C68034 EZ-USB NX2LP-Flex™ Flexible USB NAND Flash Controller Document #: 001-04247 Rev. *C				
REV.	**ECN NO.**	**Issue Date**	**Orig. of Change**	**Description of Change**
**	388499	See ECN	GIR	Preliminary draft.
*A	394699	See ECN	XUT	Minor Change: Upload data sheet to external website. Publicly announcing the parts. No physical changes to document were made.
*B	400518	See ECN	GIR	Took ‘Preliminary’ off the top of all pages. Corrected the first bulleted item. Corrected Figure 3-2 caption. Added new logo.
*C	433952	See ECN	RGL	Added I^2C functionality.