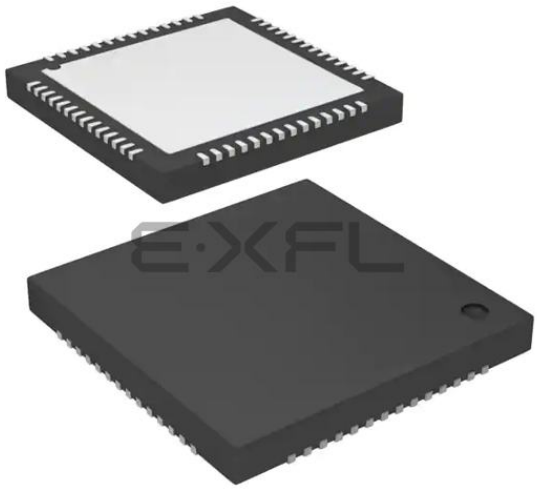


Welcome to E-XFL.COM

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

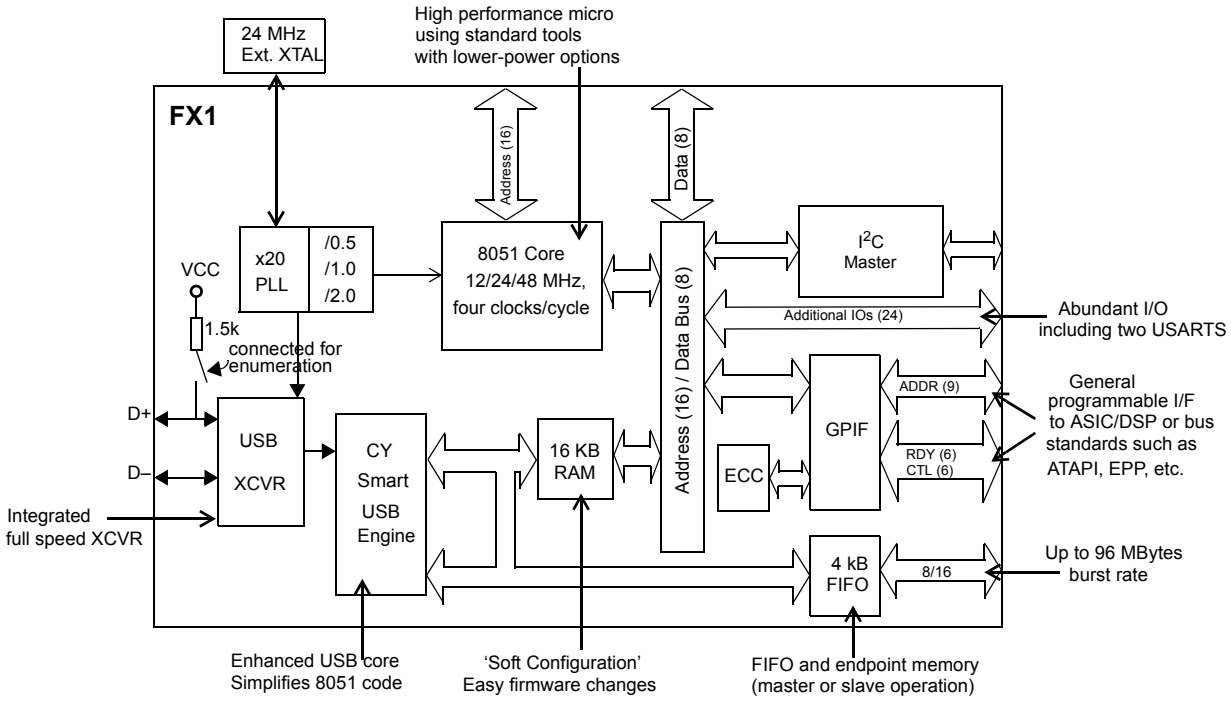
Embedded - Microcontrollers - Application Specific represents a category of microcontrollers designed with unique features and capabilities tailored to specific application needs. Unlike general-purpose microcontrollers, application-specific microcontrollers are optimized for particular tasks, offering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

What Are Embedded - Microcontrollers - Application Specific?

Application-specific microcontrollers are engineered to

Details	
Product Status	Obsolete
Applications	USB Microcontroller
Core Processor	8051
Program Memory Type	ROMless
Controller Series	CY7C647xx
RAM Size	16K x 8
Interface	I²C, USB, USART
Number of I/O	24
Voltage - Supply	3.15V ~ 3.45V
Operating Temperature	0°C ~ 70°C
Mounting Type	Surface Mount
Package / Case	56-VFQFN Exposed Pad
Supplier Device Package	56-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/infineon-technologies/cy7c64713-56ltxc

Logic Block Diagram



Functional Description

EZ-USB FX1™ (CY7C64713) is a full speed, highly integrated, USB microcontroller. By integrating the USB transceiver, Serial Interface Engine (SIE), enhanced 8051 microcontroller, and a programmable peripheral interface in a single chip, Cypress has created a very cost effective solution that provides superior time-to-market advantages.

The EZ-USB FX1 is more economical, because it incorporates the USB transceiver and provides a smaller footprint solution than the USB SIE or external transceiver implementations. With EZ-USB FX1, the Cypress Smart SIE handles most of the USB protocol in hardware, freeing the embedded microcontroller for application specific functions and decreasing the development time to ensure USB compatibility.

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

Four Pb-free packages are defined for the family: 56-pin SSOP, 56-pin QFN, 100-pin TQFP, and 128-pin TQFP.

Applications

- DSL modems
- ATA interface
- Memory card readers
- Legacy conversion devices
- Home PNA
- Wireless LAN
- MP3 players
- Networking

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

Functional Overview

USB Signaling Speed

FX1 operates at one 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.

FX1 does not support the low speed signaling mode of 1.5 Mbps or the high speed mode of 480 Mbps.

8051 Microprocessor

The 8051 microprocessor embedded in the FX1 family has 256 bytes of register RAM, an expanded interrupt system, three timer/counters, and two USARTs.

8051 Clock Frequency

FX1 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 the 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 is dynamically changed by the 8051 through the CPUCS register.

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

USARTS

FX1 contains two standard 8051 USARTs, addressed by Special Function Register (SFR) bits. The USART interface pins are available on separate I/O pins, and are not multiplexed with port pins.

UART0 and UART1 can operate using an internal clock at 230 KBaud with no more than 1% baud rate error. 230 KBaud operation is achieved by an internally derived clock source that generates overflow pulses at the appropriate time. The internal clock adjusts for the 8051 clock rate (48, 24, 12 MHz) such that it always presents the correct frequency for 230-KBaud operation.[1]

Special Function Registers

Certain 8051 SFR addresses are populated to provide fast access to critical FX1 functions. These SFR additions are shown in Table 1 on page 5. Bold type indicates non-standard, enhanced 8051 registers. The two SFR rows that end with ‘0’ and ‘8’ contain bit addressable registers. The four I/O ports A–D use the SFR addresses used in the standard 8051 for ports 0–3, which are not implemented in the FX1. Because of the faster and more efficient SFR addressing, the FX1 I/O ports are not addressable in the external RAM space (using the MOVX instruction).

Note
1. 115-KBaud operation is also possible by programming the 8051 SMOD0 or SMOD1 bits to a ‘1’ for UART0 and UART1, respectively.

ReNumeration™

Because the FX1's configuration is soft, one chip can take on the identities of multiple distinct USB devices.

When first plugged into the USB, the FX1 enumerates automatically and downloads firmware and the USB descriptor tables over the USB cable. Next, the FX1 enumerates again, this time as a device defined by the downloaded information. This patented two step process, called ReNumeration, happens instantly when the device is plugged in, with no indication that the initial download step has occurred.

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

Before reconnecting, the firmware sets or clears the RENUM bit to indicate if the firmware or the Default USB Device handles device requests over endpoint zero:

- RENUM = 0, the Default USB Device handles device requests
- RENUM = 1, the firmware handles device requests

Bus-powered Applications

The FX1 fully supports bus powered designs by enumerating with less than 100 mA as required by the USB specification.

Interrupt System

INT2 Interrupt Request and Enable Registers

FX1 implements an autovector feature for INT2 and INT4. There are 27 INT2 (USB) vectors, and 14 INT4 (FIFO/GPIF) vectors. See EZ-USB Technical Reference Manual (TRM) for more details.

USB-Interrupt Autovectors

The main USB interrupt is shared by 27 interrupt sources. The FX1 provides a second level of interrupt vectoring, called Autovectoring, to save code and processing time that is normally required to identify the individual USB interrupt source. When a USB interrupt is asserted, the FX1 pushes the program counter on to its stack and then jumps to address 0x0043, where it expects to find a "jump" instruction to the USB Interrupt service routine.

The FX1 jump instruction is encoded as shown in Table 3.

If Autovectoring is enabled (AV2EN = 1 in the INTSETUP register), the FX1 substitutes its INT2VEC byte. Therefore, if the high byte ("page") of a jump table address is preloaded at location 0x0044, the automatically inserted INT2VEC byte at 0x0045 directs the jump to the correct address out of the 27 addresses within the page.

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, such as the USB Interrupt, can employ autovectoring. Table 4 on page 7 shows the priority and INT4VEC values for the 14 FIFO/GPIF interrupt sources.

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, such as the USB Interrupt, can employ autovectoring. Table 4 on page 7 shows the priority and INT4VEC values for the 14 FIFO/GPIF interrupt sources.

Table 3. INT2 USB Interrupts

USB INTERRUPT TABLE FOR INT2			
Priority	**INT2VEC Value**	**Source**	**Notes**
1	00	SUDAV	Setup Data Available
2	04	SOF	Start of Frame
3	08	SUTOK	Setup Token Received
4	0C	SUSPEND	USB Suspend request
5	10	USB RESET	Bus reset
6	14		Reserved
7	18	EP0ACK	FX1 ACK'd the CONTROL Handshake
8	1C		Reserved
9	20	EP0-IN	EP0-IN ready to be loaded with data
10	24	EP0-OUT	EP0-OUT has USB data
11	28	EP1-IN	EP1-IN ready to be loaded with data
12	2C	EP1-OUT	EP1-OUT has USB data
13	30	EP2	IN: buffer available. OUT: buffer has data
14	34	EP4	IN: buffer available. OUT: buffer has data
15	38	EP6	IN: buffer available. OUT: buffer has data
16	3C	EP8	IN: buffer available. OUT: buffer has data
17	40	IBN	IN-Bulk-NAK (any IN endpoint)

Reset and Wakeup

Reset Pin

The input pin, RESET#, resets the FX1 when asserted. This pin has hysteresis and is active LOW. When a crystal is used with the CY7C64713, the reset period must allow for the stabilization of the crystal and the PLL. This reset period must be approximately 5 ms after VCC has reached 3.0 Volts. If the crystal input pin is driven by a clock signal the internal PLL stabilizes in 200 µs after VCC has reached 3.0 V[4]. Figure 2 on page 8 shows a power on reset condition and a reset applied during operation. A power on reset is defined as the time a reset is asserted when power is being applied to the circuit. A powered reset is defined to be when the FX1 has been previously powered on and operating and the RESET# pin is asserted.

Cypress provides an application note which describes and recommends power on reset implementation and is found on the Cypress web site. While the application note discusses the FX2, the information provided applies also to the FX1. For more information on reset implementation for the FX2 family of products visit http://www.cypress.com.

Figure 2. Reset Timing Plots

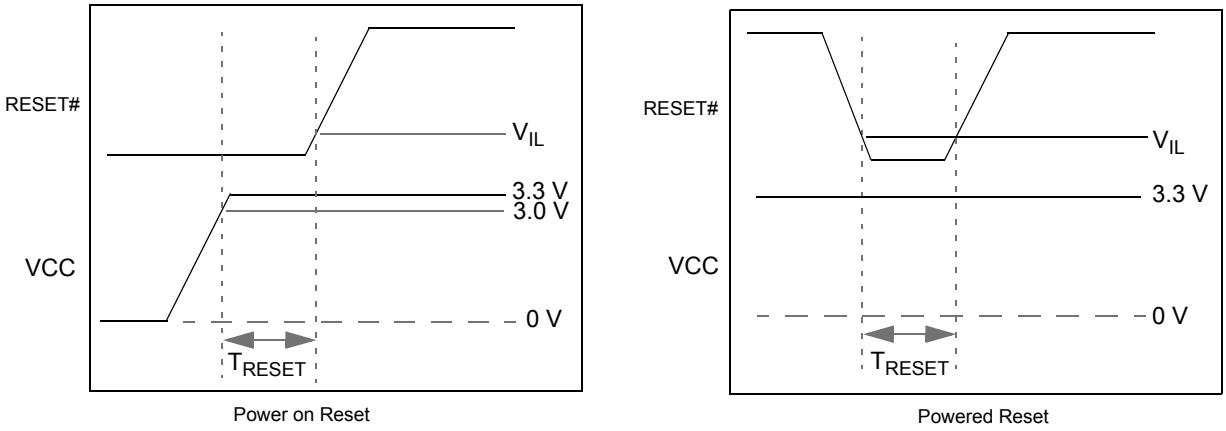


Table 5. Reset Timing Values

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

Wakeup Pins

The 8051 puts itself and the rest of the chip into a power down mode by setting PCON.0 = 1. This stops the oscillator and PLL. When WAKEUP is asserted by external logic, the oscillator restarts, after the PLL stabilizes, and then the 8051 receives a wakeup interrupt. This applies irrespective of whether the FX1 is connected to the USB or not.

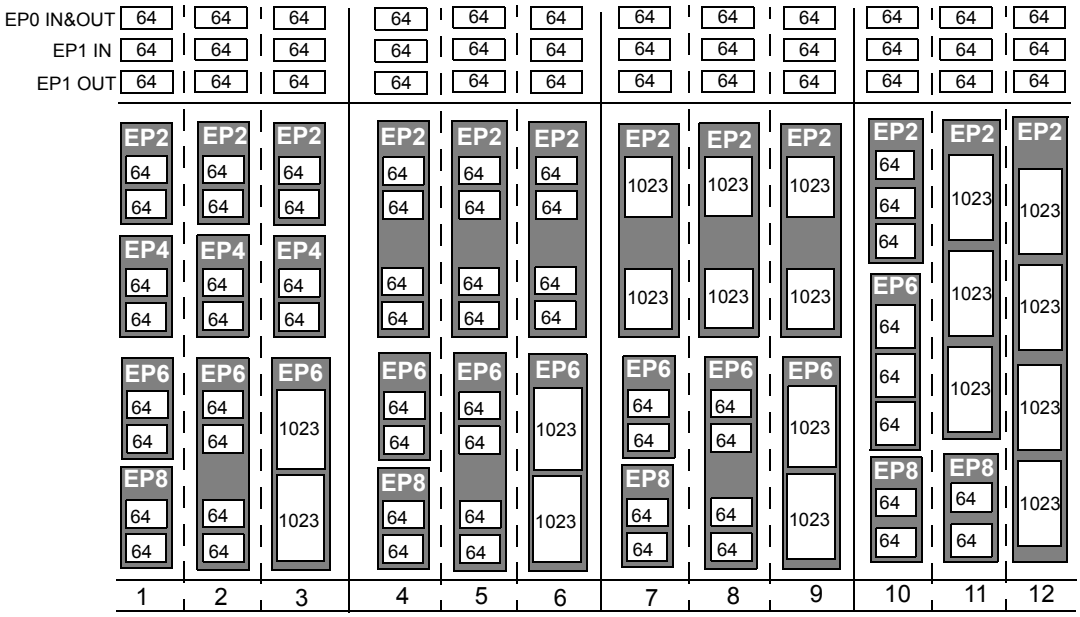
The FX1 exits the power down (USB suspend) state using one of the following methods:

- USB bus activity (if D+/D– lines are left floating, noise on these lines may indicate activity to the FX1 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.

Note
4. If the external clock is powered at the same time as the CY7C64713 and has a stabilization wait period. It must be added to the 200 µs.

Figure 6. Endpoint Configuration



Master/Slave Control Signals

The FX1 endpoint FIFOS are implemented as eight physically distinct 256 × 16 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 instantaneously, giving essentially zero transfer time between “USB FIFOS” and “Slave FIFOS”. While they are physically the same memory, no bytes are actually transferred between buffers.

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

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 RDY pins (two in the 56-pin package, six in the 100-pin and 128-pin packages) are used as flag inputs from an external FIFO or other logic if desired. The GPIF is run from either an internally derived clock or an 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 FX1 accepts either an internally derived clock or an externally supplied clock (IFCLK with a maximum frequency of 48 MHz) and SLCS#, SLRD, SLWR, SLOE, PKTEND signals from external logic. When using an external IFCLK, the external clock must be present before switching to the external clock with the IFCLKSRC bit. Each endpoint can individually be selected for byte or word operation by an internal configuration bit, and a Slave FIFO Output Enable signal SLOE enables data of the selected width. External logic must ensure that the output enable signal is inactive when writing data to a slave FIFO. The slave interface can also operate asynchronously, where the SLRD and SLWR signals act directly as strobes, rather than a clock qualifier as in the synchronous mode. The signals SLRD, SLWR, SLOE, and PKTEND are gated by the signal SLCS#.

GPIF and FIFO Clock Rates

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

GPIF

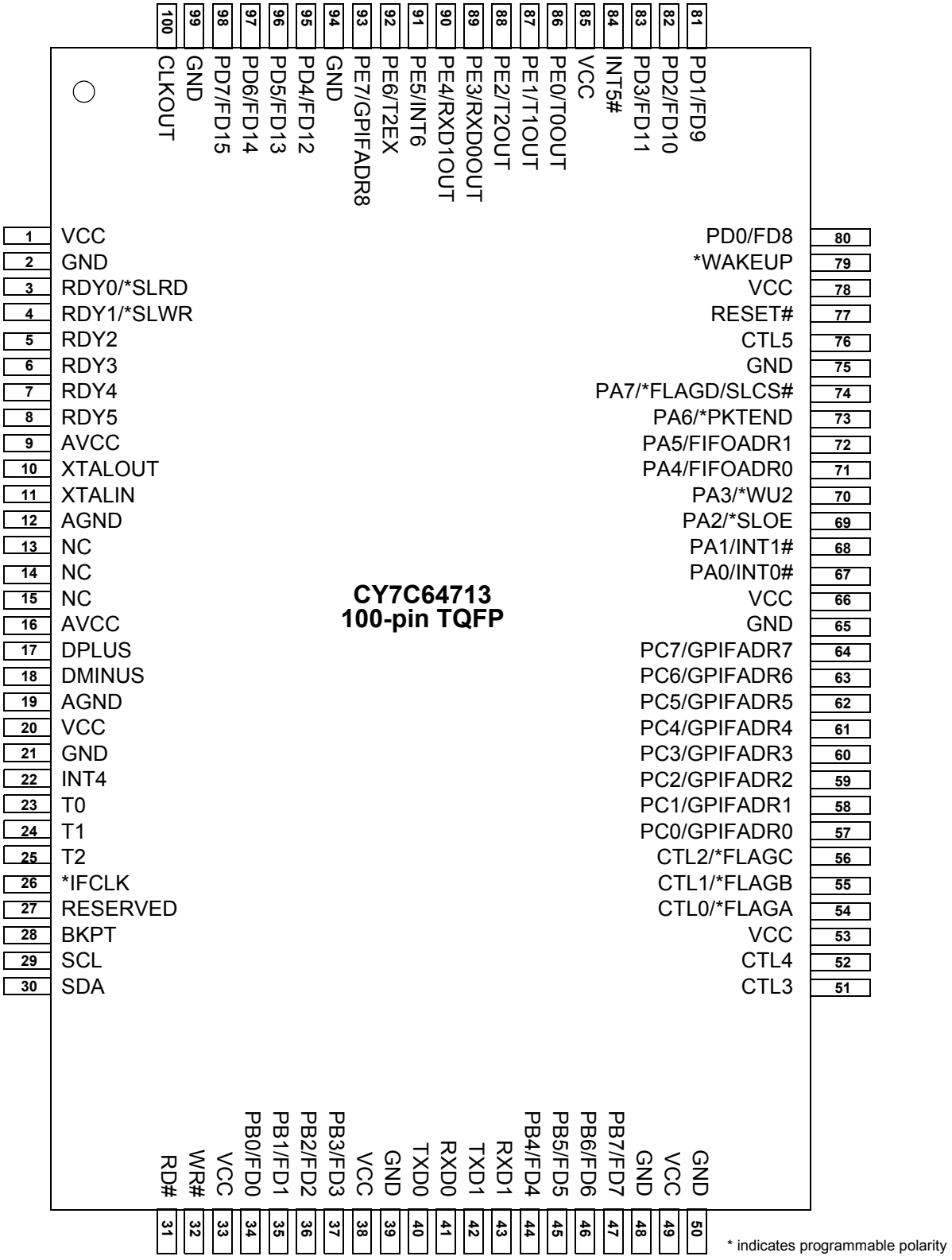
The GPIF is a flexible 8 or 16-bit parallel interface driven by a user programmable finite state machine. It allows the CY7C64713 to perform local bus mastering, and can implement a wide variety of protocols such as ATA interface, printer parallel port, and Utopia.

The GPIF has six programmable control outputs (CTL), nine address outputs (GPIFADRx), and six general purpose Ready inputs (RDY). The data bus width is 8 or 16 bits. Each GPIF vector defines the state of the control outputs, and determines what state a Ready input (or multiple inputs) must be before proceeding. The GPIF vector is programmed to advance a FIFO to the next data value, advance an address, and so on. A sequence of the GPIF vectors create a single waveform that executes to perform the data move between the FX1 and the external device.

Six Control OUT Signals

The 100-pin and 128-pin packages bring out all six Control Output pins (CTL0–CTL5). The 8051 programs the GPIF unit to define the CTL waveforms. The 56-pin package brings out three

Figure 9. CY7C64713 100-pin TQFP Pin Assignment



* indicates programmable polarity

Register Summary

FX1 register bit definitions are described in the EZ-USB TRM in greater detail.

Table 9. FX1 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											
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[8]	Slave FIFO FLAGA and FLAGB Pin Configuration	FLAGB3	FLAGB2	FLAGB1	FLAGB0	FLAGA3	FLAGA2	FLAGA1	FLAGA0	00000000	RW
E603	1	PINFLAGSCD[8]	Slave FIFO FLAGC and FLAGD Pin Configuration	FLAGD3	FLAGD2	FLAGD1	FLAGD0	FLAGC3	FLAGC2	FLAGC1	FLAGC0	00000000	RW
E604	1	FIFORESET[8]	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[8]	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[8]	Chip Revision Control	0	0	0	0	0	0	dyn_out	enh_pkt	00000000	rrrrrrbb

Note

8. Read and writes to these register may require synchronization delay, see the section “Synchronization Delay” in the EZ-USB TRM.

Table 9. FX1 Register Summary *(continued)*

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
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	I²C Bus Control & Status	START	STOP	LASTRD	ID1	ID0	BERR	ACK	DONE	000xx000	bbbrrrrr
E679	1	I2DAT	I²C Bus Data	d7	d6	d5	d4	d3	d2	d1	d0	xxxxxxxx	RW
E67A	1	I2CTL	I²C Bus Control	0	0	0	0	0	0	STOPIE	400KHZ	00000000	RW
E67B	1	XAUTODAT1	Autoptr1 MOVX access, when APTREN = 1	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
E67C	1	XAUTODAT2	Autoptr2 MOVX access, when APTREN = 1	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
		UDMA CRC											
E67D	1	UDMACRCH[17]	UDMA CRC MSB	CRC15	CRC14	CRC13	CRC12	CRC11	CRC10	CRC9	CRC8	01001010	RW
E67E	1	UDMACRCL[17]	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	0	0	0	0	DISCON	NOSYNSOF	RENUM	SIGRSUME	x0000000	rrrrbbbb
E681	1	SUSPEND	Put chip into suspend	x	x	x	x	x	x	x	x	xxxxxxxx	W
E682	1	WAKEUPCS	Wakeup Control & Status	WU2	WU	WU2POL	WUPOL	0	DPEN	WU2EN	WUEN	xx000101	bbbbrbbb
E683	1	TOGCTL	Toggle Control	Q	S	R	I/O	EP3	EP2	EP1	EP0	x0000000	rrrbbbbb
E684	1	USBFRAMEH	USB Frame count H	0	0	0	0	0	FC10	FC9	FC8	00000xxx	R

Note
17. Read and writes to these register may require synchronization delay, see the section “Synchronization Delay” in the EZ-USB TRM.

Table 9. FX1 Register Summary *(continued)*

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
E6AE	1	EP4FIFOBCL	Endpoint 4 slave FIFO total byte count L	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	00000000	R
E6AF	1	EP6FIFOBCH	Endpoint 6 slave FIFO total byte count H	0	0	0	0	BC11	BC10	BC9	BC8	00000000	R
E6B0	1	EP6FIFOBCL	Endpoint 6 slave FIFO total byte count L	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	00000000	R
E6B1	1	EP8FIFOBCH	Endpoint 8 slave FIFO total byte count H	0	0	0	0	0	BC10	BC9	BC8	00000000	R
E6B2	1	EP8FIFOBCL	Endpoint 8 slave FIFO total byte count L	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0	00000000	R
E6B3	1	SUDPTRH	Setup Data Pointer high address byte	A15	A14	A13	A12	A11	A10	A9	A8	xxxxxxxx	RW
E6B4	1	SUDPTRL	Setup Data Pointer low address byte	A7	A6	A5	A4	A3	A2	A1	0	xxxxxxx0	bbbbbbbr
E6B5	1	SUDPTRCTL	Setup Data Pointer Auto Mode	0	0	0	0	0	0	0	SDPAUTO	00000001	RW
	2	reserved											
E6B8	8	SETUPDAT	8 bytes of setup data	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	R
			SETUPDAT[0] = bmRequestType										
			SETUPDAT[1] = bmRequest										
			SETUPDAT[2:3] = wValue										
			SETUPDAT[4:5] = wIndex										
			SETUPDAT[6:7] = wLength										
		GPIF											
E6C0	1	GPIFWFSELECT	Waveform Selector	SINGLEWR1	SINGLEWR0	SINGLERD1	SINGLERD0	FIFOWR1	FIFOWR0	FIFORD1	FIFORD0	11100100	RW

Table 9. FX1 Register Summary *(continued)*

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
E6D0	1	GPIFTCB1[20]	GPIF Transaction Count Byte 1	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	00000000	RW
E6D1	1	GPIFTCB0[20]	GPIF Transaction Count Byte 0	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0	00000001	RW
	2	reserved										00000000	RW
		reserved											
		reserved											
E6D2	1	EP2GPIFFLGSEL[20]	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[20]	Endpoint 2 GPIF Trigger	x	x	x	x	x	x	x	x	xxxxxxxx	W
	3	reserved											
		reserved											
		reserved											
E6DA	1	EP4GPIFFLGSEL[20]	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[20]	Endpoint 4 GPIF Trigger	x	x	x	x	x	x	x	x	xxxxxxxx	W
	3	reserved											
		reserved											
		reserved											
E6E2	1	EP6GPIFFLGSEL[20]	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[20]	Endpoint 6 GPIF Trigger	x	x	x	x	x	x	x	x	xxxxxxxx	W

Note
20. Read and writes to these register may require synchronization delay, see the section “Synchronization Delay” in the EZ-USB TRM.

Table 9. FX1 Register Summary *(continued)*

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
	3	reserved											
		reserved											
		reserved											
E6EA	1	EP8GPIFFLGSEL[21]	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[21]	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	XGPIFSGLDATLNOX	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

Note
21. Read and writes to these register may require synchronization delay, see the section “Synchronization Delay” in the EZ-USB TRM.

Table 9. FX1 Register Summary *(continued)*

Hex	Size	Name	Description	b7	b6	b5	b4	b3	b2	b1	b0	Default	Access
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[24]	Clock Control	x	x	T2M	T1M	T0M	MD2	MD1	MD0	00000001	RW
8F	1	reserved											
90	1	IOB[24]	Port B (bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
91	1	EXIF[24]	External Interrupt Flag(s)	IE5	IE4	I²CINT	USBNT	1	0	0	0	00001000	RW
92	1	MPAGE[24]	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[24]	Autopointer 1 Address H	A15	A14	A13	A12	A11	A10	A9	A8	00000000	RW
9B	1	AUTOPTRL1[24]	Autopointer 1 Address L	A7	A6	A5	A4	A3	A2	A1	A0	00000000	RW
9C	1	reserved											
9D	1	AUTOPTRH2[24]	Autopointer 2 Address H	A15	A14	A13	A12	A11	A10	A9	A8	00000000	RW
9E	1	AUTOPTRL2[24]	Autopointer 2 Address L	A7	A6	A5	A4	A3	A2	A1	A0	00000000	RW
9F	1	reserved											
A0	1	IOC[24]	Port C (bit addressable)	D7	D6	D5	D4	D3	D2	D1	D0	xxxxxxxx	RW
A1	1	INT2CLR[24]	Interrupt 2 clear	x	x	x	x	x	x	x	x	xxxxxxxx	W
A2	1	INT4CLR[24]	Interrupt 4 clear	x	x	x	x	x	x	x	x	xxxxxxxx	W
A3	5	reserved											

Note
24. SFRs not part of the standard 8051 architecture.

Figure 13. Data Memory Read Timing Diagram

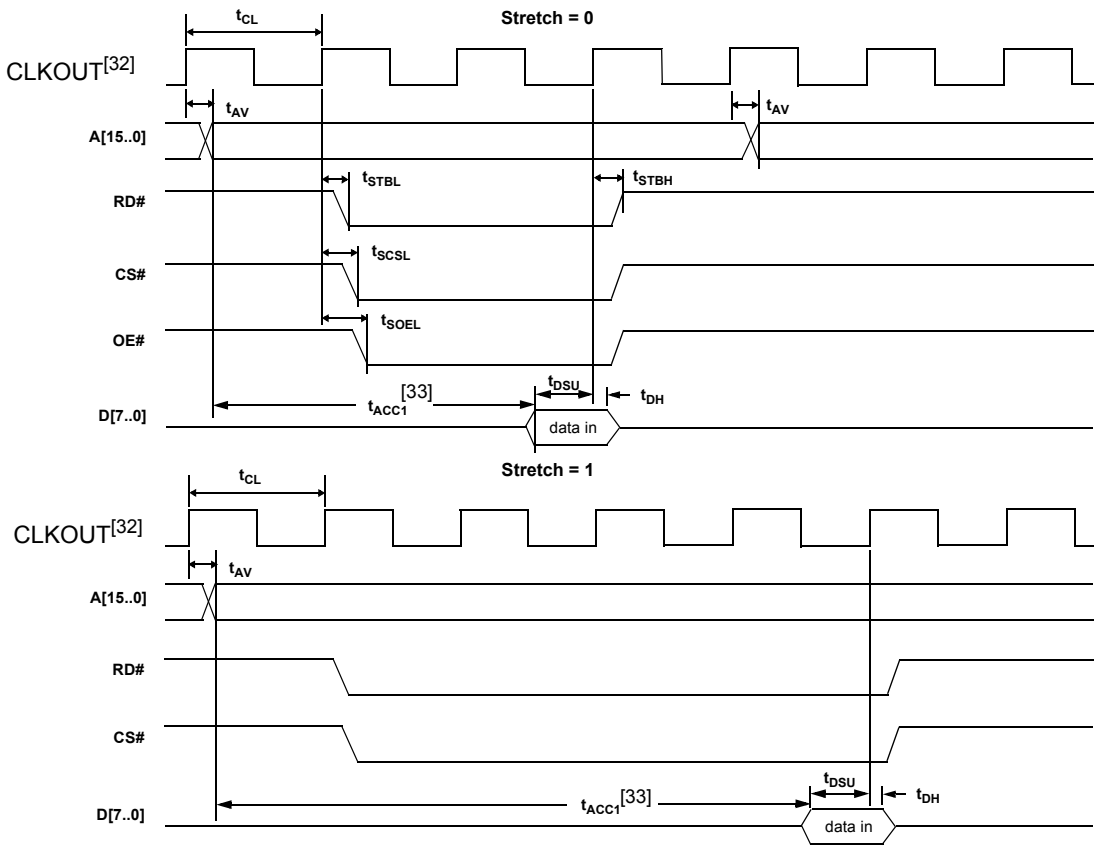


Table 11. Data Memory Read Parameters

Parameter	Description	Min	Typ	Max	Unit	Notes
t_{CL}	1/CLKOUT Frequency	–	20.83	–	ns	48 MHz
		–	41.66	–	ns	24 MHz
		–	83.2	–	ns	12 MHz
t_{AV}	Delay from Clock to Valid Address	–	–	10.7	ns	
t_{STBL}	Clock to RD LOW	–	–	11	ns	
t_{STBH}	Clock to RD HIGH	–	–	11	ns	
t_{SCSL}	Clock to CS LOW	–	–	13	ns	
t_{SOEL}	Clock to OE LOW	–	–	11.1	ns	
t_{DSU}	Data Setup to Clock	9.6	–	–	ns	
t_{DH}	Data Hold Time	0	–	–	ns	

When using the AUTPOPTR1 or AUTOPTR2 to address external memory, the address of AUTOPTR1 is active only when either RD# or WR# are active. The address of AUTOPTR2 is active throughout the cycle and meets the above address valid time for which is based on the stretch value.

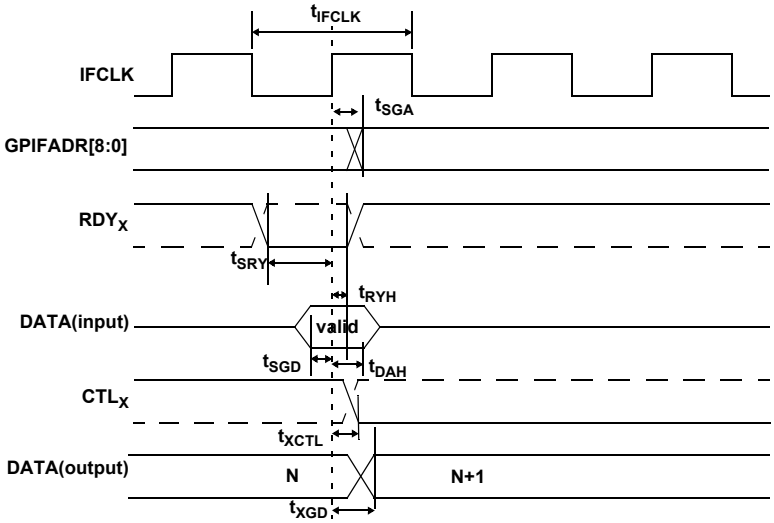
Notes
32. CLKOUT is shown with positive polarity.
33. t_{ACC2} and t_{ACC3} are computed from the parameters in Table 11 as follows:
t_{ACC2}(24 MHz) = 3 × t_{CL} – t_{AV} – t_{DSU} = 106 ns
t_{ACC2}(48 MHz) = 3 × t_{CL} – t_{AV} – t_{DSU} = 43 ns

t_{ACC3}(24 MHz) = 5 × t_{CL} – t_{AV} – t_{DSU} = 190 ns
t_{ACC3}(48 MHz) = 5 × t_{CL} – t_{AV} – t_{DSU} = 86 ns.

GPIF Synchronous Signals

In the following figure, dashed lines indicate signals with programmable polarity.

Figure 18. GPIF Synchronous Signals Timing Diagram



The following table provides the GPIF Synchronous Signals Parameters with Internally Sourced IFCLK. [34, 35]

Table 13. GPIF Synchronous Signals Parameters with Internally Sourced IFCLK

Parameter	Description	Min	Max	Unit
t_{IFCLK}	IFCLK Period	20.83	–	ns
t_{SRY}	RDY_X to Clock Setup Time	8.9	–	ns
t_{RYH}	Clock to RDY_X	0	–	ns
t_{SGD}	GPIF Data to Clock Setup 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

The following table provides the GPIF Synchronous Signals Parameters with Externally Sourced IFCLK.[35]

Table 14. GPIF Synchronous Signals Parameters with Externally Sourced IFCLK

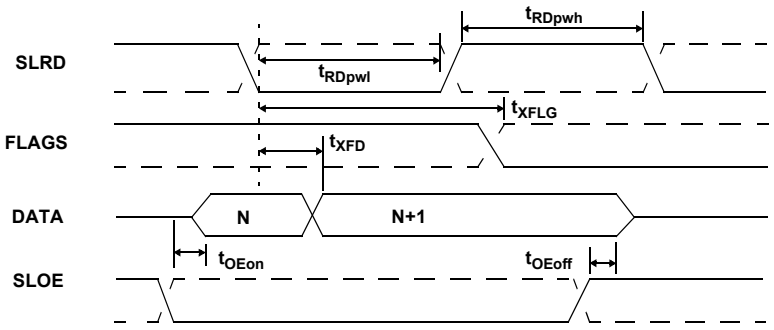
Parameter	Description	Min	Max	Unit
t_{IFCLK}	IFCLK Period	20.83	200	ns
t_{SRY}	RDY_X to Clock Setup Time	2.9	–	ns
t_{RYH}	Clock to RDY_X	3.7	–	ns
t_{SGD}	GPIF Data to Clock Setup Time	3.2	–	ns
t_{DAH}	GPIF Data Hold Time	4.5	–	ns
t_{SGA}	Clock to GPIF Address Propagation Delay	–	11.5	ns
t_{XGD}	Clock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

Notes
34. GPIF asynchronous RDY_x signals have a minimum Setup time of 50 ns when using internal 48-MHz IFCLK.
35. IFCLK must not exceed 48 MHz.

Slave FIFO Asynchronous Read

In the following figure, dashed lines indicate signals with programmable polarity.

Figure 20. Slave FIFO Asynchronous Read Timing Diagram



In the following table, the Slave FIFO asynchronous parameter values use internal IFCLK setting at 48 MHz.

Table 17. Slave FIFO Asynchronous Read Parameters

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 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.

Single and Burst Synchronous Write

In the following figure, dashed lines indicate signals with programmable polarity.

Figure 32. Slave FIFO Synchronous Write Sequence and Timing Diagram

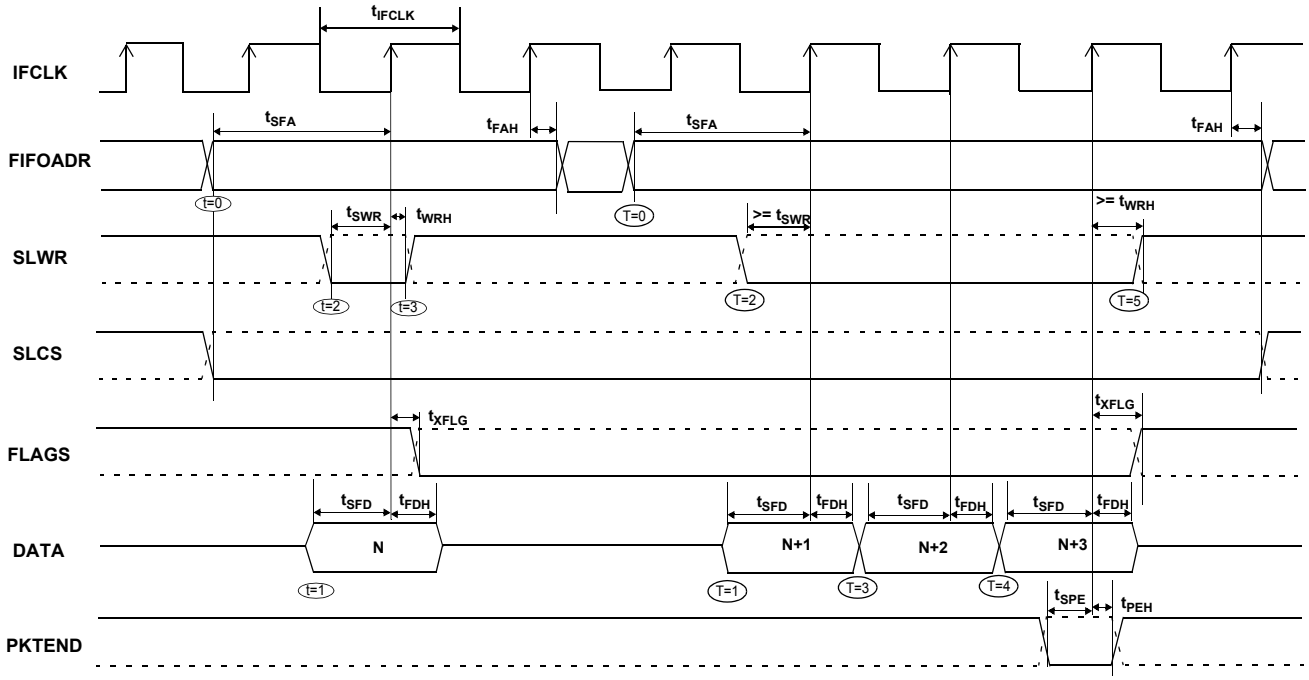


Figure 32 shows the timing relationship of the SLAVE FIFO signals during a synchronous write using IFCLK as the synchronizing clock. This diagram illustrates a single write followed by burst write of 3 bytes and committing all 4 bytes as a short packet using the PKTEND pin.

- 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 setup time is more than one IFCLK cycle.

- At t = 1, the external master or 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 setup time of t_{SWR} (time from asserting the SLWR signal to the rising edge of IFCLK) and maintain a minimum hold time of t_{WRH} (time from the IFCLK edge to the deassertion of the SLWR signal). If SLCS signal is used, it must be asserted with SLWR or before SLWR is asserted. (that is the SLCS and SLWR signals must both be asserted to start a valid write condition).
- While the SLWR is asserted, data is written to the FIFO and on the rising edge of the IFCLK, the FIFO pointer is incremented. The FIFO flag is also updated after a delay of t_{XFLG} from the rising edge of the clock.

The same sequence of events are also shown for a burst write and are marked with the time indicators of T = 0 through 5.

Note For the burst mode, SLWR and SLCS are left asserted for the entire duration of writing all the required data values. In this burst write mode, after the SLWR is asserted, the data on the FIFO data bus is written to the FIFO on every rising edge of IFCLK. The FIFO pointer is updated on each rising edge of IFCLK. In Figure 32, after the four bytes are written to the FIFO, SLWR is deasserted. The short 4-byte packet is committed to the host by asserting the PKTEND signal.

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

Although there are no specific timing requirement for asserting PKTEND, there is a specific corner case condition that needs attention while using the PKTEND to commit a one byte or word packet. Additional timing requirements exist when the FIFO is configured to operate in auto mode and it is necessary 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 or 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 or 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 Table 19 on page 55 for further details on this timing.

Sequence Diagram of a Single and Burst Asynchronous Read

Figure 33. Slave FIFO Asynchronous Read Sequence and Timing Diagram

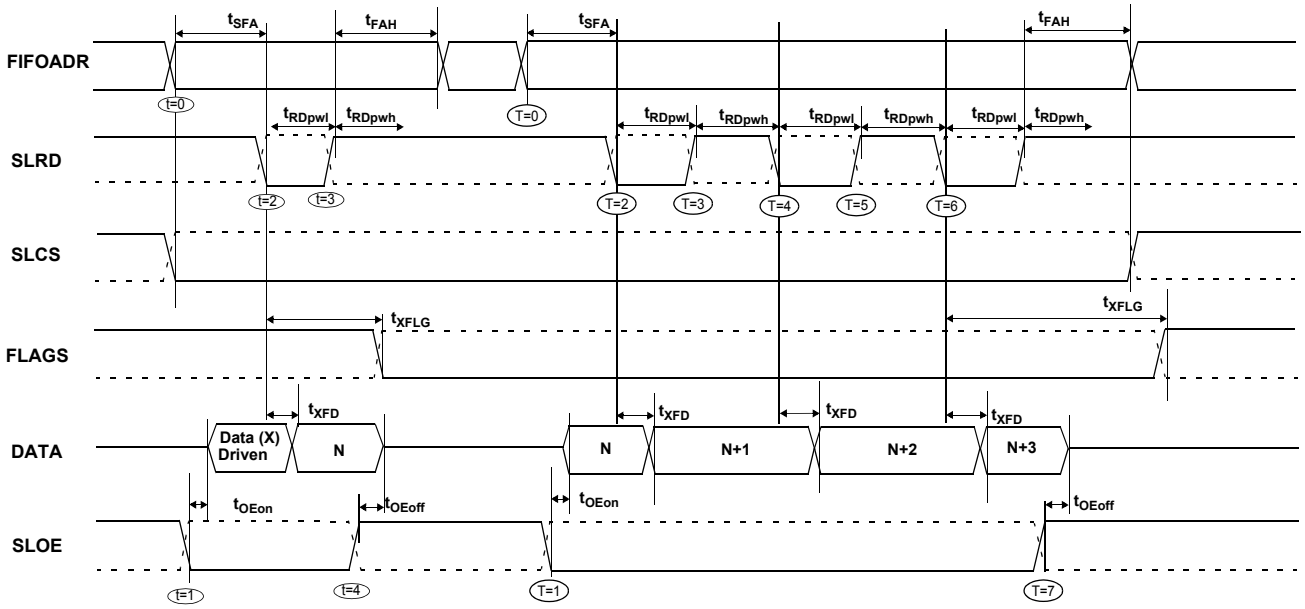


Figure 34. Slave FIFO Asynchronous Read Sequence of Events Diagram

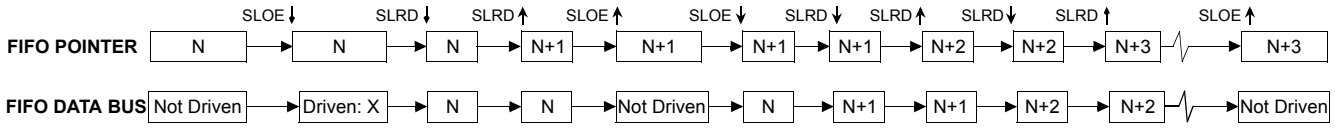


Figure 33 shows the timing relationship of the SLAVE FIFO signals during an asynchronous FIFO read. It shows a single read followed by a burst read.

- At t = 0 the FIFO address is stable and the SLCS signal is asserted.
- At t = 1, SLOE is asserted. This results in the data bus being driven. The data that is driven on to the bus is 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 (that is, the SLCS and SLRD signals must both be asserted to start a valid read condition).
- The data that drives after asserting SLRD, is the updated data from the FIFO. This data is valid after a propagation delay of t_{XFD} from the activating edge of SLRD. In Figure 33, data N is the first valid data read from the FIFO. For data to appear on the data bus during the read cycle (that is, SLRD is asserted), SLOE MUST be in an asserted state. SLRD and SLOE can also be tied together.

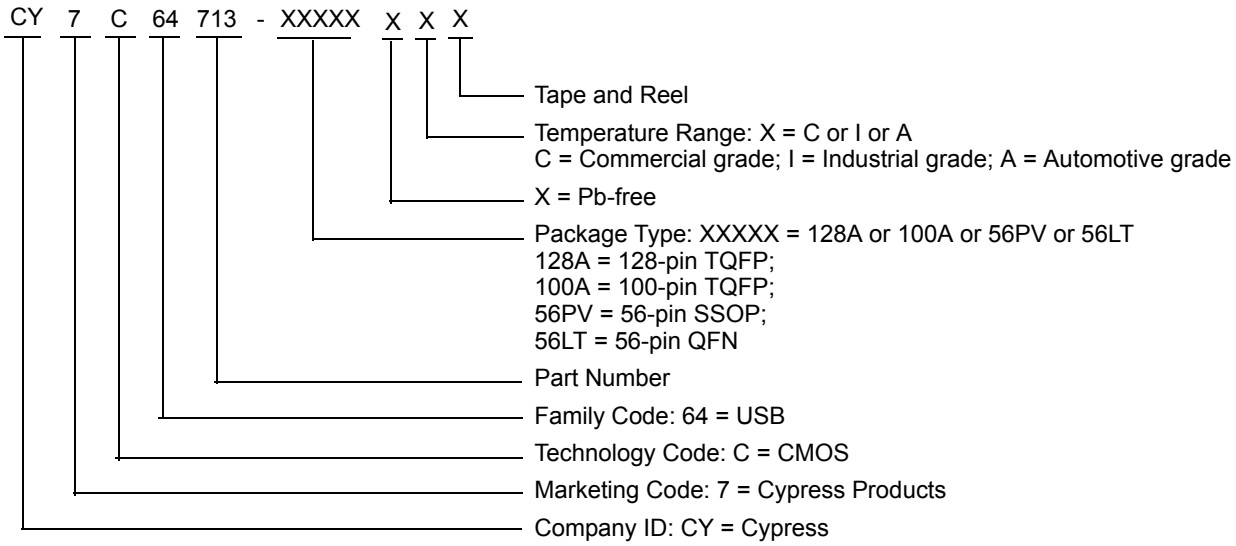
The same sequence of events is also shown for a burst read marked with T = 0 through 5.

Note In burst read mode, during SLOE is assertion, the data bus is in a driven state and outputs the previous data. After the 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.

Ordering Information

Ordering Code	Package Type	RAM Size	# Prog I/Os	8051 Address/Data Busses
CY7C64713-128AXC	128-pin TQFP - Pb-free	16K	40	16/8 bit
CY7C64713-100AXC	100-pin TQFP - Pb-free	16K	40	–
CY7C64713-56PVXC	56-pin SSOP - Pb-free	16K	24	–
CY7C64713-56LTXC	56-pin QFN - Pb-free	16K	24	–
CY3674	EZ-USB FX1 Development Kit			

Ordering Code Definitions

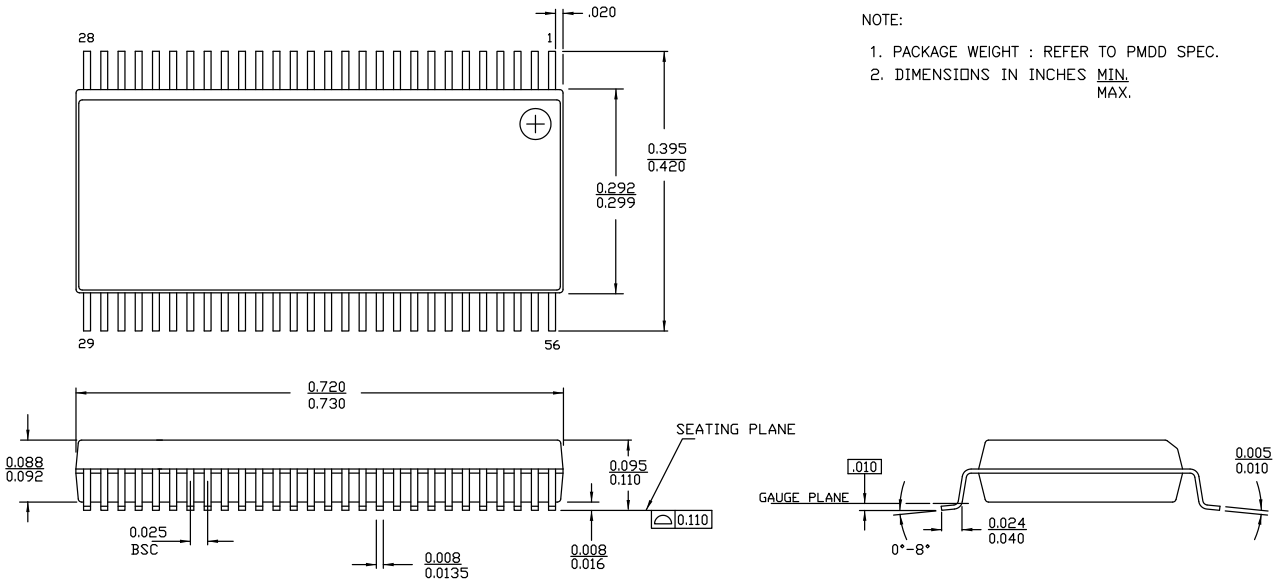


Package Diagrams

The FX1 is available in four packages:

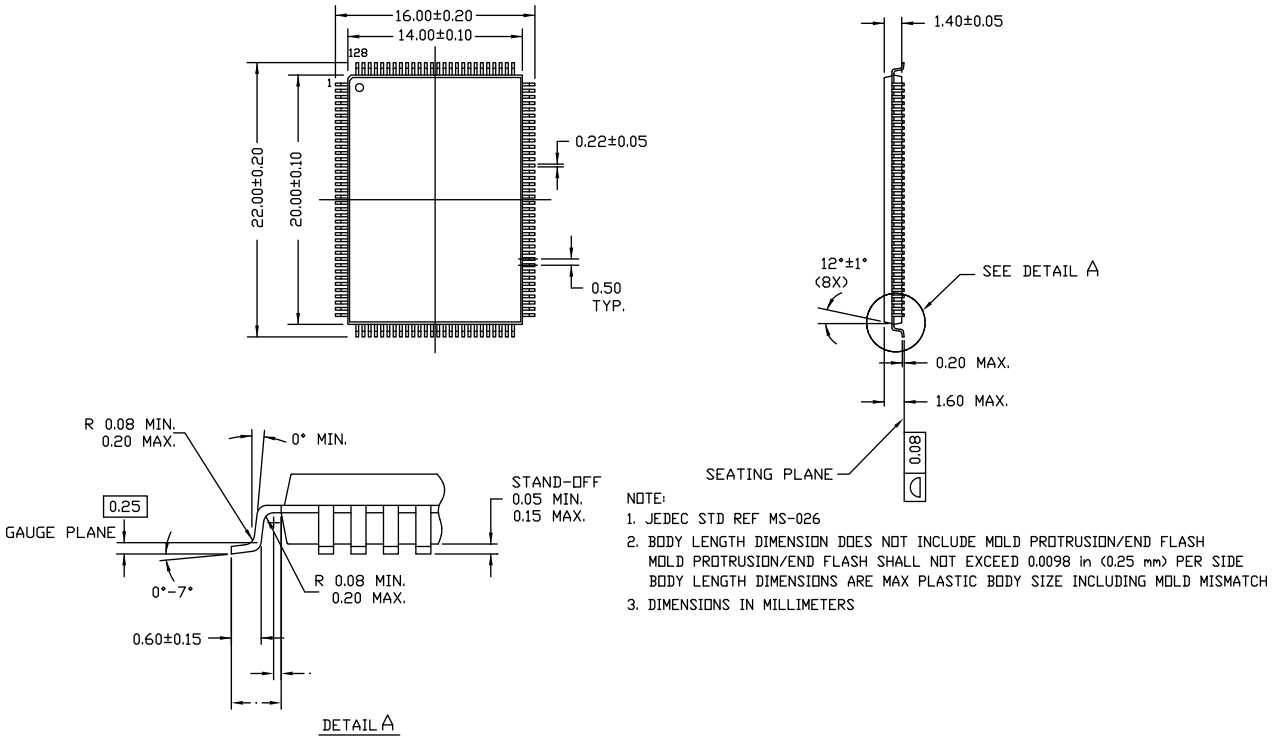
- 56-pin SSOP
- 56-pin QFN
- 100-pin TQFP
- 128-pin TQFP

Figure 36. 56-pin SSOP 300 Mils O563



51-85062 *F

Figure 39. 128-pin TQFP (14 × 20 × 1.4 mm) A128RA



51-85101 *F

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. As a result, 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 FX1 through the device's metal paddle on the bottom side of the package. Heat from here, is conducted to the PCB at the thermal pad. It is then conducted from the thermal pad to the PCB inner ground plane by a 5 × 5 array of via. A via is a plated through hole in the PCB with a finished diameter of 13 mil. The QFN's metal die paddle must be soldered to the PCB's thermal pad. Solder mask is placed on the board top side over each via to resist solder flow into the via. The mask on the top side also minimizes outgassing during the solder reflow process.

For further information on this package design please refer to 'Application Notes for Surface Mount Assembly of Amkor's MicroLeadFrame (MLF) Packages'. This can be found on Amkor's website http://www.amkor.com.

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

Figure 40 on page 69 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 must 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 41 on page 69 is a plot of the solder mask pattern and Figure 42 on page 69 displays an X-Ray image of the assembly (darker areas indicate solder).