Chapter 8. General Programmable Interface (GPIF)

8.1 What is GPIF?

The **G**eneral **P**rogrammable **I**nter**F**ace (GPIF) is an extremely flexible 8- or 16-bit parallel interface that allows designers to reduce system costs by providing a *glueless* interface between the EZ-USB FX and many different types of external peripherals.

The GPIF allows the EZ-USB FX to perform local bus mastering to external peripherals using a wide variety of protocols. For example, EIDE/ATAPI, printer parallel port (IEEE P1284), Utopia, and other interfaces can be supported using the GPIF block of the EZ-USB FX.

To support a wide range of applications, the GPIF implements an extensive feature set that can be modified to suit the design. As with other highly configurable chips, some initialization steps are required. To support a range of interface styles, the GPIF provides multiple programmable I/O pins and multiple registers to configure those pins.

This chapter provides an overview of GPIF, discusses external connections, and explains the operation of the GPIF *engine*.

Figure 8-1 presents a block diagram illustrating GPIF's place in the FX System.

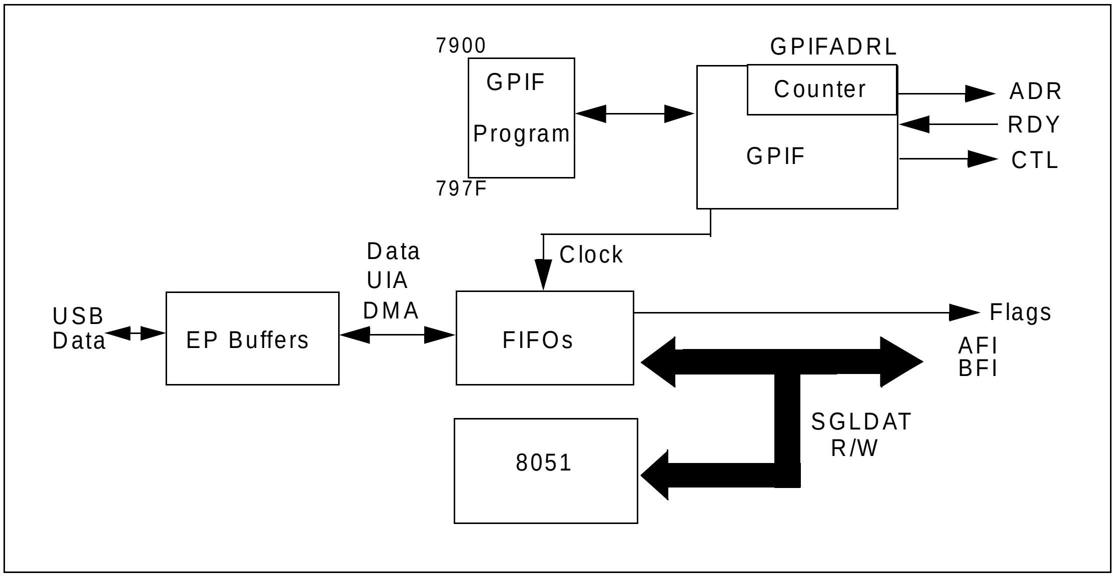

Figure 8-1. GPIF's Place in the FX System

8.2 Applicable Documents and Tools

- EZ-USB FX Data Sheet
- EPP Reference Design
- Mass Storage Reference Design
- GPIF Tool — A Windows application that assists GPIF firmware development. The GPIF Tool can be found on the EZ-USB *FX* Developer's Kit CD.

8.3 Typical GPIF Interface

The GPIF allows the EZ-USB FX connect directly to external peripherals such as ASICs, DSPs, or other digital logic that uses an 8- or 16-bit parallel interface.

The GPIF provides external pins that can operate as outputs (CTL0 to CTL5), inputs (RDY0 to RDY5), Data bus (GDA[7..0] and GDB[7..0]), and Address Lines (ADR0 to ADR5).

A Waveform Descriptor in internal RAM describes the behavior of each of the GPIF signals. The Waveform Descriptor is loaded into the GPIF registers by the 8051 firmware during initialization, and it is then used throughout the execution of the 8051 code to perform transactions over the GPIF interface.

Figure 8-2 shows a block diagram of a typical interface between the EZ-USB FX and a peripheral function.

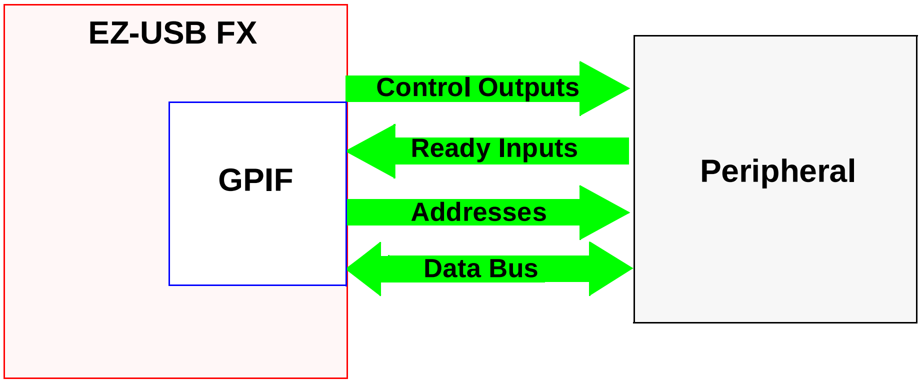

Figure 8-2. EZ-USB FX Interfacing to a Peripheral

The following sections detail the features available and steps needed to create an efficient GPIF design. This includes definition of the external GPIF connections and the internal register settings, along with 8051 firmware needed to execute data transactions over the interface.

8.4 External GPIF Connections

8.4.1 The External GPIF Interface

The GPIF provides many general input and output signals with which to interface your external peripherals *gluelessly* to the EZ-USB FX.

The GPIF interface signals are shown in **Table 8-1**.

Table 8-1. GPIF Pin Descriptions

PIN	IN/OUT	Description
ADR[5:0]	O	Address outputs
GDA[7:0]	I/O	Bidirectional A-FIFO data bus
GDB[7..0]	I/O	Bidirectional B-FIFO data bus
CTL[5:0]	O	Programmable control outputs
RDY[5:0]	I	Sampleable ready inputs

Refer to the figure *EZ-USB FX 128-pin Package* on p. 13 of the *CY7C64603/613 Data Sheet*.

The Control Outputs (CTL0 to CTL5) are intended to be strobes, read/write lines, and other non-bused outputs.

The Ready Inputs (RDY0 to RDY5) sample a signal to allow a transaction to wait (inserting wait states), continue, or repeat until the signal is at the appropriate level.

The GPIF Data Bus is a collection of the GDA[7..0] and GDB[7..0] pins.

- A GPIF interface 8 bits wide uses pins GDA[7..0].
- A GPIF interface 16 bits wide uses pins GDA[7..0] and GDB[7..0].

The GPIF Address lines (ADR0 to ADR5) can generate an automatically incrementing address during a burst transaction. For non-burst transactions, these address lines remain static. For higher-order address lines that may be needed, other non-GPIF I/O signals should be used.

The GPIF Clock can be either an internal 48MHz clock, or an externally-supplied clock from the XCLK pin. If the XCLK_SEL pin is tied high, the GPIF clock is the XCLK pin. Otherwise, the GPIF clock is the 48 MHz internal clock.

8.4.2 Connecting GPIF Signal Pins to Hardware

The first step in creating the interface between the EZ-USB FX GPIF and your peripheral is to define the hardware interconnects. This physical connection of GPIF signals to your interface signals determines the configurations that are necessary by your 8051 firmware.

1. **Determine the proper GPIF Data Bus size.** If your interface's data bus is 8 bits wide, use the GDA[7..0] pins. If your interface's data bus is 16 bits wide, use GDA[7..0] and GDB[7..0].

2. **Assign the CTL signals to your interface**. Make a list of all interface signals to be driven by your peripheral (inputs to the GPIF), and assign them to the RDY0 to RDY5 Inputs. If there are more input signals than available RDY inputs, you need to use other, non-GPIF I/O signals and sample them manually using 8051 firmware. In this case, you should choose the RDY inputs only for signals that must be sampled in the middle of a data transaction.

3. **Assign the RDY signals to your interface.** Make a list of all interface signals to be driven by your peripheral (inputs to the GPIF), and assign them to the RDY0 through RDY5 Inputs. If there are more input signals than available RDY inputs, you will need to use other non-GPIF I/O signals and sample them manually using 8051 firmware. In this case, you should choose to use the RDY inputs only for signals that must be sampled in the middle of a data transaction.

4. **Determine the proper GPIF Address connections**. If your interface uses an Address Bus, use the ADR0 through ADR5 signals for the least significant bits, and other non-GPIF I/O signals for the most significant bits. You may leave these signals unconnected if you do not use address signals, as with a FIFO.

8.4.3 Example GPIF Hardware Interconnect

The following example illustrates the hardware connections that can be made for a standard interface: a 27C256 EPROM.

Table 8-2. Example GPIF Hardware Interconnect

Step	Result	Connection Made
1. Determine the proper GPIF Data Bus size.	8 bits.	Connect GDA[7..0] to D0..D7 of the EPROM.
2. Assign the CTL signals to your interface.	CS# and OE# are inputs to the EPROM.	Connect CTL0 to CS# and CTL1 to OE#.
3. Assign the RDY signals to your interface.	There are no output ready/wait signals from a 27C256 EPROM.	No connection.
4. Determine the proper GPIF Address connections.	16 bits of address.	Connect ADR0..ADR5 to A0..A5 and other I/O ports to A6..A15.

The process is the same for larger and more complicated interfaces.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.