USB FIFO - Fast Parallel Data Transfer IC

FEATURES

- **Single Chip Fast Data Transfer Solution**
- **Send / Receive Data over USB at up to 1 M Bytes / sec**
- **384 byte FIFO Transmit buffer / 128 byte FIFO receive buffer for high data throughput**
- **Simple interface to CPU or MCU bus**
- **No in-depth knowledge of USB required as all USB Protocol is handled automatically within the I.C.**
- **FTDI's Virtual COM port drivers eliminate the need for USB driver development in most cases.**
- **Compact 32 pin (7mm x 7mm) MQFP package**
- **Integrated 6MHz - 48MHz Clock Multiplier aids FCC and CE compliance**
- **Integrated 3.3v Regulator – No External Regulator Required**
- **4.4v .. 5.25v Single Supply Operation**
- **UHCI / OHCI Compliant**
- **USB 1.1 Specification Compliant**
- **USB VID, PID, Serial Number and Product Description Strings in external E2PROM.**

Virtual COM Port Drivers for –

Windows 98 and Windows 98 SE
Windows 2000
Windows Millennium **
Apple iMAC **
Linux **

- **Application Areas**

USB ISDN and ADSL Modems
High Speed USB ⇔ PDA Communications
USB I/F for Digital Cameras
USB I/F for MP3 players
High Speed USB Instrumentation
USB ⇔ USB data transfer cables
USB ⇔ USB null-modem cables

GENERAL DESCRIPTION

The FT8U245AM provides an easy cost-effective method of transferring data to / from a peripheral and a host P.C. at up to 8 Million bits (1 Megabyte) per second. It's simple FIFO-like design makes it easy to interface to any CPU (MCU) either by mapping the device into the Memory / IO map of the CPU, using DMA or controlling the device via IO ports.

To send data from the peripheral to the host P.C. simply write the byte wide data into the device when the transmitter empty status bit is not active. If the (384 byte) transmit buffer fills up, the device de-asserts transmit empty in order to stop further data being written to the device until some of the FIFO data has been transferred over USB.

When the host P.C. sends data to the peripheral over USB, the device will assert the receiver full status bit to let the peripheral know that data is available. The peripheral then reads the data until the receiver full status bit goes inactive, indicating no more data is available to read.

By using FTDI's virtual COM Port drivers, the peripheral looks like a standard COM Port to the application software. Commands to set the baud rate are ignored – the device always transfers data at it's fastest rate regardless of the application's baud rate setting.