USB 3.0 Connectivity using the Cypress EZ-USB FX3 Controller

PLC2 FPGA Days
June 20, 2012
Stuttgart

Martin Heimlicher
Enclustra GmbH
FPGA Solution Center
Content

- Enclustra Company Profile
- USB 3.0 Overview
  - What is new?
  - Why USB 3.0?
- EZ-USB FX3 Hardware and Software
  - Hardware
  - Software Framework
  - Memory Resources
  - GPIF 2 Designer Software
  - DMA Configurations
  - Debugging the FX3
  - Boot Options
- Slave Fifo Interface
  - Configuration
  - Threads
  - FPGA Implementation
  - Timings
  - Difficulties
- Enclustra USB 3.0 Solutions
  - FPGA Manager FX3
  - Mars PM3 Base Board
  - Performance Demo
Content

- Enclustra Company Profile
  - USB 3.0 Overview
    - What is new?
    - Why USB 3.0?
  - EZ-USB FX3 Hardware and Software
    - Hardware
    - Software Framework
    - Memory Resources
    - GPIF 2 Designer Software
    - DMA Configurations
    - Debugging the FX3
    - Boot Options

- Slave Fifo Interface
  - Configuration
  - Threads
  - FPGA Implementation
  - Timings
  - Difficulties

- Enclustra USB 3.0 Solutions
  - FPGA Manager FX3
  - Mars PM3 Base Board
  - Performance Demo
Quick Facts

- Founded in 2004
- Located in the Technopark of Zurich
- 8 FPGA Engineers, 1 Technician, 1 Secretary
- Vendor-Independent

FPGA Design Center

- Provider of FPGA Design Services
- HDL Firmware (VHDL, Verilog, C++)
- Hardware (High-Speed, Analog, RF)
- Embedded Software (for FPGA processors)
Mars Family
- SO-DIMM 67.6 x 30 mm
- 108 User I/Os (Digital, LVDS, MGT, Analog)
- 1-2 Ethernet Ports, 0-1 USB Port
- 3.3V Single Supply Voltage

Mercury Family
- 56 x 54 mm
- 146-168 User I/Os (Digital, LVDS)
- 1 Ethernet Port, 1 USB Port
- 5-15V Single Supply Voltage

Base Boards & FMC Cards
- Custom „Dream” Modules
Enclustra FPGA Solution Center
IP Solutions

- IP Cores
  - Universal Drive Controller (DC/BLDC/SM)
  - 2D-Accelerated Display Controller (LVCMOS/LVDS/DVI/HDMI)
  - Camera Input Controllers (Camera Link/MIPI/LVDS)
  - UDP Streaming Ethernet Controller (10/100/1G/10G)
  - Resource-saving Memory Controllers (DDRx/QDRx/Flash)
  - Available with AMBA AXI4 and Avalon-compliant interfaces

- IP Solutions
  - FPGA Manager USB 3.0 Solution (Cypress EZ-USB FX3)
  - Easy Licensing: Evaluation, Academic, Project, Site, Source
  - Maintenance & Support
  - Custom „Dream” IP
Content

- Enclustra Company Profile
- **USB 3.0 Overview**
  - What is new?
  - Why USB 3.0?
- EZ-USB FX3 Hardware and Software
  - Hardware
  - Software Framework
  - Memory Resources
  - GPIF 2 Designer Software
  - DMA Configurations
  - Debugging the FX3
  - Boot Options
- **Slave Fifo Interface**
  - Configuration
  - Threads
  - FPGA Implementation
  - Timings
  - Difficulties
- Enclustra USB 3.0 Solutions
  - FPGA Manager FX3
  - Mars PM3 Base Board
  - Performance Demo
### Overview - What is new?

<table>
<thead>
<tr>
<th>Name</th>
<th>Data rate</th>
<th>Symbol rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low Speed</td>
<td>1.5 Mbit/s</td>
<td>1.875 Mbit/s</td>
</tr>
<tr>
<td>Full Speed</td>
<td>12 Mbit/s</td>
<td>15 Mbit/s</td>
</tr>
<tr>
<td>Hi-Speed</td>
<td>480 Mbit/s</td>
<td>600 Mbit/s</td>
</tr>
<tr>
<td>SuperSpeed</td>
<td>4 Gbit/s (500 MB/s)</td>
<td>5 Gbit/s</td>
</tr>
</tbody>
</table>

- USB 3.0 is using high-speed serial signaling similar to PCIe and relies on proper cabling and PCB design!
Overview - What is new? (2)

- With USB 3.0 two additional differential pairs are introduced that support unidirectional data transfer at 5 Gbit/s
- New cables are therefore needed for USB 3.0
- USB 2.0 cables and most plugs are compatible but force USB 3.0 devices and hosts to work in USB 2.0 mode

<table>
<thead>
<tr>
<th>Pins Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBUS</td>
<td>Power.</td>
</tr>
<tr>
<td>D-</td>
<td>USB 2.0 Differential Pair.</td>
</tr>
<tr>
<td>D+</td>
<td></td>
</tr>
<tr>
<td>GND</td>
<td>Ground for Power Return.</td>
</tr>
<tr>
<td>SSRX-</td>
<td>SuperSpeed Receiver Differential Pair.</td>
</tr>
<tr>
<td>SSRX+</td>
<td></td>
</tr>
<tr>
<td>SSTX-</td>
<td>SuperSpeed Transmit Differential Pair.</td>
</tr>
<tr>
<td>SSTX+</td>
<td></td>
</tr>
<tr>
<td>GND_DRAIN</td>
<td>Ground for Signal Return.</td>
</tr>
</tbody>
</table>

[1]
Overview - Why USB 3.0?

- 8 MP ○
  (12-bit color, 30 fps)

- 5 MP ○
  (24-bit color, 30 fps)

- 3 MP ○
  (24-bit color, 30 fps)

- 1080p ○
  (24-bit color, 30 fps)

- 720p ○
  (24-bit color, 60 fps)

- XGA ○
  (24-bit color, 60 fps)

- VGA ○
  (24-bit color, 60 fps)

- FireWire

- GigE

- USB SuperSpeed

Resolution vs. Bandwidth

[2]
Overview - Why USB 3.0?

**USB 2.0, NO COMPRESSION**
- Limited video quality, low frame rate

**USB 2.0, MJPEG COMPRESSION**
- Low cost implementation
- Good enough image quality but lossy

**USB 2.0, H.264 COMPRESSION**
- Higher BOM cost and more board space
- Better image quality for most applications

**USB 3.0, NO COMPRESSION**
- Lower BOM and smaller board
- Highest image quality

[2]
Overview – Why USB 3.0?

INDUSTRIAL CAMERAS  GAMING  CONSUMER CAMERAS  MEDICAL IMAGING

SURVEILLANCE  VIDEO CONFERENCING  SCANNERS  BIOMETRICS

HD VIDEO CAPTURE  DATA ACQUISITION  AUDIO MIXER
• Enclustra Company Profile
• USB 3.0 Overview
  • What is new?
  • Why USB 3.0?
• EZ-USB FX3 Hardware and Software
  • Hardware
  • Software Framework
  • Memory Resources
  • GPIF 2 Designer Software
  • DMA Configurations
  • Debugging the FX3
  • Boot Options
• Slave Fifo Interface
  • Configuration
  • Threads
  • FPGA Implementation
  • Timings
  • Difficulties
• Enclustra USB 3.0 Solutions
  • FPGA Manager FX3
  • Mars PM3 Base Board
  • Performance Demo
• General Programmable Interface II (GPIF II) allows maximum flexibility for connecting data sources / sinks
• Automatic data transfer between USB and GPIF II port thanks to DMA engine
• ARM9 core can pre-process the data before sending to the host
• UART, SPI, I2C and I2S interfaces are used to load the FX3 firmware and/or configure external peripherals
• USB Charger and accessory detection (EZ-Detect)
Hardware - CPU

- 32-bit, 200 MHz ARM926EJ-S core
- The core has direct access to 16 kB of Instruction Tightly Coupled Memory (TCM) and 8 kB of Data TCM
- JTAG interface for firmware download and debugging
- 512 kB of embedded SRAM for code and data
- 8 kB of instruction and data cache
- DMA connectivity between the various peripherals (i.e. USB, GPIF II, I2S, SPI, UART)
- Industry-standard development tools for ARM926EJ-S can be used
AN76405 - EZ-USB® FX3 Boot Options
AN70193 - EZ-USB® FX3 SPI Boot Option
AN68914 - EZ-USB® FX3 I2C Boot Option
AN70707 - EZ-USB® FX3 Hardware Design Guidelines and Schematic Checklist
AN77960 - Introduction to EZ-USB® FX3’s High-Speed USB Host Controller
AN76348 - Migrating from EZ-USB® FX2LP™ Based Design to EZ-USB FX3 Based Design
AN75432 - USB 3.0 EZ-USB® FX3™ Orientation
AN75705 - Getting Started with FX3
AN68829 - Slave FIFO Interface for EZ-USB® FX3™: 5-Bit Address Mode
AN65974 - Designing with the EZ-USB® FX3 Slave FIFO Interface
AN73304 - Booting EZ-USB® FX3 over Synchronous ADMux Interfaces
AN73150 - Booting EZ-USB® FX3 over High-Speed USB
- Application framework for FX3 provided by Cypress
- Host USB library with simple read and write function exists
- ThreadX RTOS running on FX3
- GPIF II interface can be configured using GPIF II Designer software
- All necessary tools except compiler for host system are part of the SDK
- For on-chip debugging, an ARM9 compatible JTAG probe with GDB server is required
- Application can be downloaded to RAM and I2C / SPI attached memory over USB using the Control Center application
Software Framework – Host USB Library

- C++ and C# library available
- Simple read and write functions to transfer data between host and device
- Cypress demo applications can be used as a starting point for the application development
• The USB device is opened by instantiating the CCyUSBDevice class
• Based on the endpoint mode defined in the USB descriptor the appropriate derived CCyUSBEndpoint class is used
• ThreadX RTOS enables multi-threaded applications
• FX3 API provides a complete framework to configure and operate the FX3
• No direct access to FX3 registers and RAM necessary
Software Framework – FX3 API

- ThreadX RTOS supports:
  - Threads
  - Message Queues
  - Semaphores
  - Mutex
  - Memory Allocation
  - Events
  - Timer

- Cypress application examples can be used as a starting point for the development

[1]
• CPU and memory initialization is done before entering the main() function
• GPIOs and peripherals can be configured in the main() function using the FX3 API
• User application is running in its own thread
• DMA channels are set up in the user application
512 kB RAM partitioned into:

- 256 kB DMA buffer
- 32 kB User Data
- 180 kB Code
- 8 kB RTOS heap
- 8 kB Stack
- 4 kB Interrupt and exception vectors

➢ Cypress delivered application framework and RTOS results already in 145 kB code size
**FX3 – GPIF II Designer Software**

**STEP 1: CHOOSE YOUR PINS**
- Functions as master or slave
- Provides 256 firmware programmable states
- Supports 8 Bit, 16 Bit and 32 Bit parallel data bus
- Enables interface frequencies up to 100 MHz

**STEP 2: DESIGN YOUR INTERFACE**
- Supports 14 configurable control pins when 32 Bit data bus is used. All control pins can be either input/output or bidirectional.
- Supports 16 configurable control pins when 16 or 8 Bit data bus is used. All control pins can be either input/output or bidirectional.
DMA – Auto Channel

- No CPU intervention
- Data cannot be modified
- Maximum performance
- Optional call-back functionality
- CPU can modify every data packet
- Interrupt driven
- Performance severely reduced
DMA – Manual IN Channel

- Useful if data is read and processed on FX3
One possible application is to send debug messages over UART.
• FX3 application can be downloaded and debugged over JTAG (e.g. Segger or Signum JTAG probe) using Eclipse and GDB server
FX3 firmware can be loaded from various sources:

<table>
<thead>
<tr>
<th>PMODE[2:0][2]</th>
<th>Boot From</th>
</tr>
</thead>
<tbody>
<tr>
<td>F00</td>
<td>Sync ADMUX (16-bit)</td>
</tr>
<tr>
<td>F01</td>
<td>Async ADMUX (16-bit)</td>
</tr>
<tr>
<td>F11</td>
<td>USB boot</td>
</tr>
<tr>
<td>F0F</td>
<td>Async SRAM (16-bit)</td>
</tr>
<tr>
<td>F1F</td>
<td>I²C, On Failure, USB Boot is Enabled</td>
</tr>
<tr>
<td>1FF</td>
<td>I²C only</td>
</tr>
<tr>
<td>0F1</td>
<td>SPI, On Failure, USB Boot is Enabled</td>
</tr>
</tbody>
</table>

- When using USB boot mode, the firmware is downloaded by the host driver and therefore driver and firmware versions are always matching
- If the FX3 needs to work without USB connection, the firmware image can be stored using an external memory attached to the SPI or I²C interface
<table>
<thead>
<tr>
<th>Enclustra Company Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB 3.0 Overview</td>
</tr>
<tr>
<td>What is new?</td>
</tr>
<tr>
<td>Why USB 3.0?</td>
</tr>
<tr>
<td>EZ-USB FX3 Hardware and Software</td>
</tr>
<tr>
<td>Hardware</td>
</tr>
<tr>
<td>Software Framework</td>
</tr>
<tr>
<td>Memory Resources</td>
</tr>
<tr>
<td>GPIF 2 Designer Software</td>
</tr>
<tr>
<td>DMA Configurations</td>
</tr>
<tr>
<td>Debugging the FX3</td>
</tr>
<tr>
<td>Boot Options</td>
</tr>
<tr>
<td>Slave Fifo Interface</td>
</tr>
<tr>
<td>Configuration</td>
</tr>
<tr>
<td>Threads</td>
</tr>
<tr>
<td>FPGA Implementation</td>
</tr>
<tr>
<td>Timings</td>
</tr>
<tr>
<td>Difficulties</td>
</tr>
<tr>
<td>Enclustra USB 3.0 Solutions</td>
</tr>
<tr>
<td>FPGA Manager FX3</td>
</tr>
<tr>
<td>Mars PM3 Base Board</td>
</tr>
<tr>
<td>Performance Demo</td>
</tr>
</tbody>
</table>
The FX3 slave fifo interface provides four threads that are addressed using two address signals.

Each thread can be configured as read or write thread.

Flags can be assigned to a fixed thread or the currently addressed thread.

Each thread can be linked to a DMA channel and its associated buffer.
• FlagA and FlagB can be configured as full/empty and watermark flags
• Flags can be assigned to current thread or a fixed thread
• Data bus width can be set to 8, 16 or 32 Bit
• Two threads are addressed using the SlFifo_Addr signal
• One thread is used as write and one as read thread
• Flag A is configured as read thread empty
• Flag B is configured as write thread almost full
Synchronous Read Cycle Timing

PCLK

SLCS

FIFO ADDR

SLRD

SLOE

FLAGA
(dedicated thread Flag for An)
(1 = Not Empty 0 = Empty)

FLAGB
(dedicated thread Flag for Am)
(1 = Not Empty 0 = Empty)

Data Out

SLWR (HIGH)

3 cycle latency from addr to data

t_{CYC}

t_{CL}

t_{RD}

t_{RDH}

t_{OEZ}

[3]
Slave Fifo Interface – Write Timings

Synchronous Write Cycle Timing

- $t_{Cyc}$
- $t_{As}$, $t_{AM}$
- $t_{WRS}$, $t_{WRH}$
- $t_{DS}$, $t_{OH}$
- $t_{PEH}$, $t_{PEH}$

Enclustra GmbH
### Slave Fifo Interface – Timings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREQ</td>
<td>Interface clock frequency</td>
<td>–</td>
<td>100</td>
<td>MHz</td>
</tr>
<tr>
<td>tCYC</td>
<td>Clock period</td>
<td>10</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tCH</td>
<td>Clock high time</td>
<td>4</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tCL</td>
<td>Clock low time</td>
<td>4</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tRDS</td>
<td>SLRD# to CLK setup time</td>
<td>2</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tRDH</td>
<td>SLRD# to CLK hold time</td>
<td>0.5</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tWRS</td>
<td>SLWR# to CLK setup time</td>
<td>2</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tWRH</td>
<td>SLWR# to CLK hold time</td>
<td>0.5</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tCO</td>
<td>Clock to valid data</td>
<td>–</td>
<td>8</td>
<td>ns</td>
</tr>
<tr>
<td>tDS</td>
<td>Data input setup time</td>
<td>2</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tDH</td>
<td>CLK to data input hold</td>
<td>0.5</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tAS</td>
<td>Address to CLK setup time</td>
<td>2</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tAH</td>
<td>CLK to address hold time</td>
<td>0.5</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tOELZ</td>
<td>SLOE# to data low-Z</td>
<td>0</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tCFLG</td>
<td>CLK to flag output propagation delay</td>
<td>–</td>
<td>8</td>
<td>ns</td>
</tr>
<tr>
<td>tOEZ</td>
<td>SLOE# deassert to Data Hi Z</td>
<td>–</td>
<td>8</td>
<td>ns</td>
</tr>
<tr>
<td>tPES</td>
<td>PKTEND# to CLK setup</td>
<td>2</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tPEH</td>
<td>CLK to PKTEND# hold</td>
<td>0.5</td>
<td>–</td>
<td>ns</td>
</tr>
<tr>
<td>tCDH</td>
<td>CLK to data output hold</td>
<td>2</td>
<td>–</td>
<td>ns</td>
</tr>
</tbody>
</table>

**Note:** Three-cycle latency from ADDR to DATA/FLAGS

- Data and flags are valid only 2 ns before clock edge
- Three-cycle latency from ADDR to DATA/FLAGS need to be taken into account
Slave Fifo Interface – Difficulties

- Careful layout is required to ensure signal integrity with 32 data lines and 100 MHz interface frequency.
- Proper timing analysis needs to be done as data and flags are valid only 2 ns before the rising edge of the clock.
- Maximum data transfer is only achievable when using auto DMA mode combined with overlapped transaction on the host side.
- The 256 kB DMA memory can only buffer data for 750 us at 330 MB/s transfer rate, which is insufficient considering the non-real-time nature of the host computer. If data loss is not acceptable, additional buffering with external memory needs to be considered.
- Achievable transfer rate is still very much dependent on the employed USB host controller, driver version and operating system. This should be getting better by now as hardware and drivers are getting more mature.
- When using 32 Bit slave fifo mode, some peripherals like SPI and UART cannot be used anymore.
Content

- Enclustra Company Profile
- USB 3.0 Overview
  - What is new?
  - Why USB 3.0?
- EZ-USB FX3 Hardware and Software
  - Hardware
  - Software Framework
  - Memory Resources
  - GPIF 2 Designer Software
  - DMA Configurations
  - Debugging the FX3
  - Boot Options
- Slave Fifo Interface
  - Configuration
  - Threads
  - FPGA Implementation
  - Timings
  - Difficulties
- Enclustra USB 3.0 Solutions
  - FPGA Manager FX3
  - Mars PM3 Base Board
  - Performance Demo
FPGA manager allows simple data transfer between host computer and AXI bus.

Also enables access to I2C and SPI interfaces by high-level functions.
Enclustra – Evaluation Board (1)

- Designed for Mars family FPGA and EP modules
- 300+ MB/sec data transfer rate over USB 3.0
- Suitable for prototype and series production
- Small solution size (p-ITX, 100x72 mm)
- Available in commercial and industrial temperature grade

<table>
<thead>
<tr>
<th>Product Number</th>
<th>Supported Mars FPGA Modules</th>
<th>Temperature Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA-PM3-C</td>
<td>MA-MX1, MA-MX2, MA-AX3*, MA-ZX3*, MA-CA4*</td>
<td>0..+70°C</td>
</tr>
<tr>
<td>MA-PM3-I</td>
<td>MA-MX1, MA-MX2, MA-AX3*, MA-ZX3*, MA-CA4*</td>
<td>-25..+85°C</td>
</tr>
</tbody>
</table>
Transferring data at 320.0 MB/s
Next events:

- Sindex Bern
  
  ![Sindex Bern Logo](image)

- Embedded World
  Messezentrum Nürnberg

Martin Heimlicher
Enclustra GmbH
heimlicher@enclustra.com
Tel. +41 43 343 39 40

Newsletter: subscribe@enclustra.com

Slides in PDF format:
References