trenz electronic
Dünner Kirchweg 77
32257 Bünde
Germany
www.trenz-electronic.de

Full-Speed USB 1.1 Function Controller

2000-November-7

Evaluation Kit/ Application Note

Introduction

Application of IP-Cores requires in-depth knowledge of the core's behavior. Building up this know-how is greatly simplified by comprehensive reference applications. The following application note was developed to give the engineer a quick hands-on experience and a good starting point for own developments. To do so, a commercial off-the shelf FPGA board was chosen as the basis.

General Overview

Application of the Full-Speed USB Function Controller requires the following components:

- *The USB Function Controller.* This component implements the complete USB protocol layer. It is completely hardwired for speed and needs no firmware intervention. The functions of this entity include: frame recognition/ generation, parallel/ serial conversion, bit-stuffing/ de-stuffing, CRC checking/ generation, PID verification/generation, address recognition and handshake evaluation/ generation.
- *Microcontroller & firmware.* The microcontroller in conjunction with the firmware implements the USB Device Framework (Chapter 9 Commands). Due to the hardwired protocol layer of the USB Function Controller all interaction between the Function Controller and the microcontroller happens with very relaxed timing requirements, which even the smallest controller can handle easily.
- *Glue Logic.* To interface the USB Function Controller with the microcontroller, a minimum amount of glue logic is required, implementing a simple memory-mapped interface.
- *Additional Endpoints.* In case the USB Function requires additional Endpoints to connect the Functional Block with USB, the control logic for these Endpoints needs to be implemented.
- *Functional Block.* This component implements the unique device functionality, e.g. a Human Interface Device, a Modem or a Monitor.

Architectural Description

XESS XSP-010 Board

The XSP-010 board is a versatile FPGA development board providing a 10k gate FPGA, an 8051 compatible microcontroller, a 7-segment LED display and 32kBytes of SRAM. *Figure 1* shows the block diagram of this board. Most of the logic required to interface the microcontroller with its external components, namely the address latch and SRAM interface is implemented inside the FPGA. This application clocks the board with 48MHz, to simplify glue logic design. Please note, that due to the narrow DPLL lock range clocking the board with 50MHz from the on-board programmable oscillator won't work.

The VHDL entity *XSP010* implements the complete board, allowing total system simulation. This is especially helpful for firmware development and debugging, as all registers of the 8051 are visible in the simulator.

The entity *te51* is a behavioral model of the industry-standard 8051 microcontroller. This model is not included in the Evaluation Kit, instead any third party model may be used. The model is required to support MOVX instructions, while interrupts and built-in peripherals are optional. In addition to this the model should be cycle-compatible with the original MCS 51 timing, as successful simulation depends on the exact timing of firmware intervention. Trenz Electronic's *te51* meets all these requirements and is therefore recommended.

The *SRAM* entity models the 32kByte RAM. RAM contents can be pre-loaded by passing a generic with the path n7ame of an Intel Hex file. This allows to create a push button flow from firmware assembly to simulation. The size of RAM is currently limited to 4kByte, to reduce simulator memory usage and increase simulation speed.

Xilinx XCS010 PC84 FPGA

The XCS010 FPGA is a low-cost FPGA with 10k system gates, which should be plenty for small microcontroller projects. All logic contained in the FPGA is implemented in entity *xspFPGA*, *Figure 2* shows the block diagram of the entity.

The glue logic required to de-multiplex the AD bus (Port 0) and interface the microcontroller with the SRAM is implemented in entity *xspUc*. The logic consists of 8bit address register and an 8bit three-state bus driver. In contrast to typical 8051 applications the address register is implemented with edge-triggered flip-flops instead of transparent latches.

FPGA Core Logic

The FPGA core logic implemented in entity *xspCORE*, contains all application-specific logic, mainly the USB Function Controller and the Functional Block. To keep this application note as comprehensive as possible, only the absolute minimum of logic is implemented here. See *Figure 3* for a block diagram of entity *xspCORE*.

Entity *usbEP0* implements the USB Function Controller. A 48MHz oscillator drives *clk48*, an asynchronous, active-high reset on *rst* is required during power-on. The 12MHz clock is generated by the internal digital PLL, therefore *clk12* is driven by *clk12o*. The generics *epin_mask*, *epout_mask*, *epsetup_mask* and *episo_mask* are set up to define Endpoint Zero as the one and only endpoint, with isochronous transfers being not supported. All signals not shown in the diagram are left open. Special care has to be taken, that the data bus *txd* is set to high impedance.

The entity *xspUSB* implements glue logic to connect the USB Function Controller with the 8051 microcontroller. Datapathes to access the FIFO, the Control/Status Word and the Device Address are required. These items are memory-mapped for easy implementation, reducing logic to simple address decoding and three-state drivers.

The entity *xspPDIUSBP11A* interfaces the USB Function Controller with Philips Semiconductors's USB Transceiver PDIUSBP11A. This is achieved by simple combinational logic.

The entity *xspCLK* implements a simple clock divider, creating a 24MHz clock used by the 8051 microcontroller.

The entity *xspLED* implements glue logic to connect the 7-segment LED display with the microcontroller. A simple memory mapped write-only register is used here.

Simulation

In order to simulate this application note successfully, the following items should be considered:

- *Evaluation model does not support generics.* The evaluation model's generics are fixed to the values required by this application note. Changes to the values have no effect. Refer to the data sheet for further details.
- *Simulation time.* Due to the complexity of the system including 8051 and SRAM, simulation time is worth consideration. The total length of the complete testbench is about 4ms, which results in approximately 3minutes of simulation time on a Pentium-III 700MHz System (tested with Active-HDL).

Distributables

This application note comes with several distributables. Each source file contains only a single VHDL entity, to ease separation of the design into its building blocks.

Table 1 lists all files, which are specific to the application note's hardware or the XSP010 simulation environment.

Table 1: Application Note Hardware.

file	contents
xspUSB.vhd	entity xspUSB(BHV)
xspUC.vhd	entity xspUC(BHV)
xspLED.vhd	entity xspLED(BHV)
xspCORE.vhd	entity xspCORE(BHV)
xspFPGA.vhd	entity xspFPGA(BHV)
xspPHY.vhd	entity xspPHY(BHV)
xspCLK.vhd	entity xspCLK(BHV)
SRAM.vhd	SRAM(SIM)
xsp010.vhd	entity xsp010(SIM)

Table 2 lists all files which belong to the application note's firmware.

Table 2: Application Note Firmware.

file	contents
firmware.a51	firmware, 8051 assembler
firmware.hex	firmware, Intel hex file

Table 3 lists all files which belong to the application note's testbench.

Table 3: Application Note Testbench.

file	contents
TBxsp010pak.vhd	package TBxsp010pak
TBxsp010.vhd	entity TBxsp010(Test)

Table 4 lists all files which are taken from the USB Function Controller's Evaluation Kit.

Table 4: IP-Core Evaluation Package.

file	contents
usbTSTPAK.vhd	package usbTSTPAK
usbEP0eval.vhd	entity usbEP0(Eval)

References

Literature

- Full-Speed USB 1.1 Function Controller
  Product Specification
  Trenz Electronic
  http://www.trenz-electronic.de
- Universal Serial Bus Specification
  USB Implementers Forum
  http://www.usb.org
- USB Design by Example
  John Hyde
  Wiley
  http://www.usb-by-example.com

Hardware

- Xilinx Inc.
  2100 Logic Drive
  San Jose, CA 95124
  Phone: +1 408-559-7778
  Fax: +1 408-559-7114
  http://www.xilinx.com
- Philips Semiconductors
  811 East Arques Avenue
  P.O. Box 3409
  Sunnyvale, California 94088–3409
  Telephone 800-234-7381
  http://www.philips.com
- XESS Corporation
  2608 Sweetgum Drive
  Apex NC 27502
  Toll-free: 800-549-9377
  International: 919-387-0076
  FAX: 919-387-1302_
  http://www.xess.com

8051 Model

- MCS 51 Compatible
  8-bit Microcontroller Core
  http://www.trenz-electronic.de

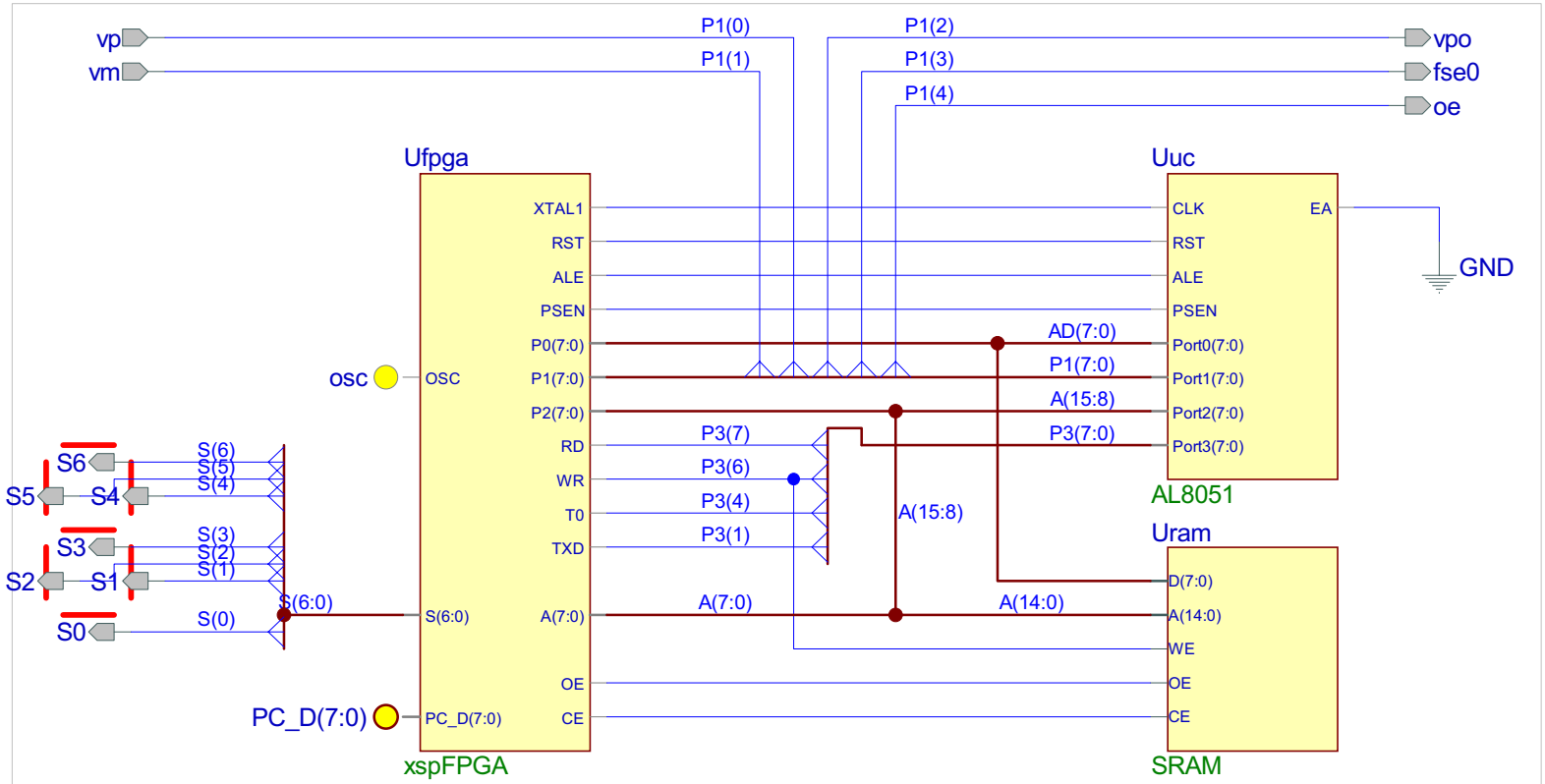
Revisions History

Table 5: Revisions History

Version	Date	Who	Description
1.0	00aug26	FB	Initial version
1.1	00nov01	FB	te51 added

Figure 1: XESS XSP-010 Board

```
constant fname: STRING:= "src\Testbenches\XSP-010\firmware.hex" ;
```

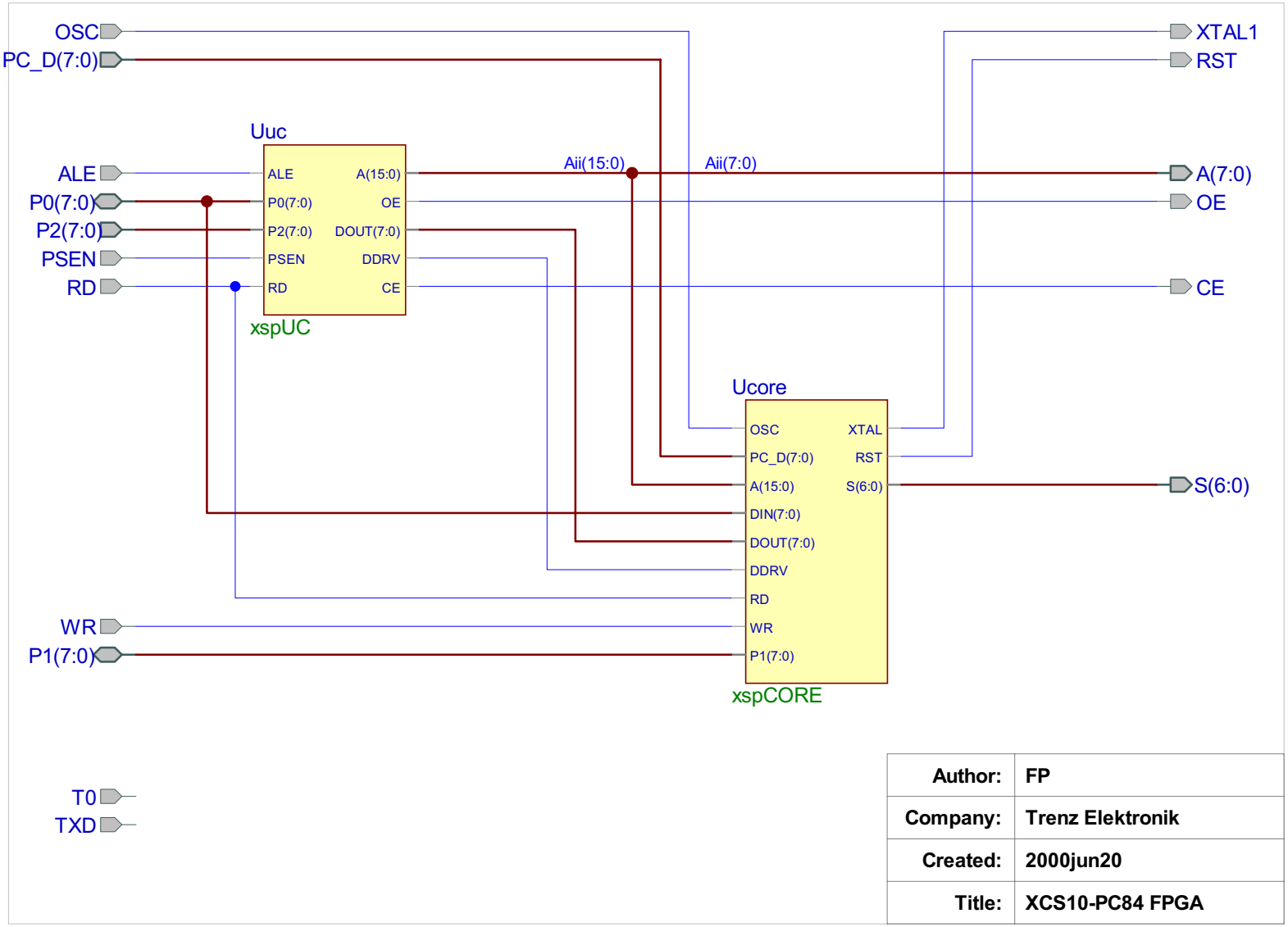



```
Statement0
process
begin
  osc<= '0', '1' after 10ns;
  wait for 21ns;
end process;

PC_D(0)<= '1', '0' after 500ns;
```

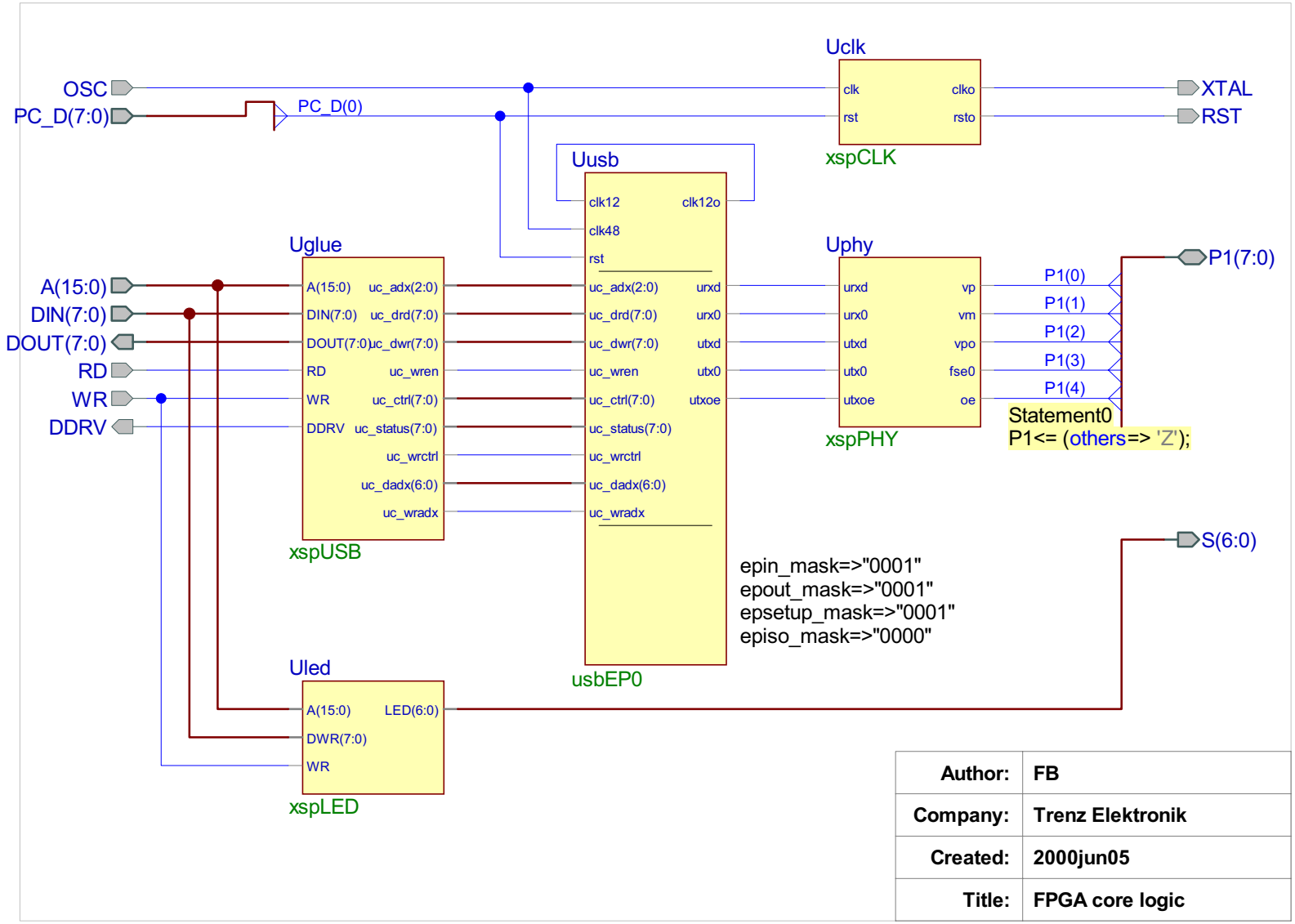
Author:	**FB**
Company:	**Trenz Elektronik**
Created:	**2000jun20**
Title:	**XESS XSP-010 Board**

Figure 2: Xilinx XCS10 FPGA



Author:	**FP**
Company:	**Trenz Elektronik**
Created:	**2000jun20**
Title:	**XCS10-PC84 FPGA**

Figure 3: FPGA Core Logic



Author:	**FB**
Company:	**Trenz Elektronik**
Created:	**2000jun05**
Title:	**FPGA core logic**