Fairchild Dual USB PHY Daughter Card Users Guide

Author: Steve Fielding
sfielding@base2designs.com

Rev. 1.0
November 17, 2008

Revision History

Rev.	Date	Author	Description
1.0	11/16/08	Sfielding	Created

Contents

1 Introduction

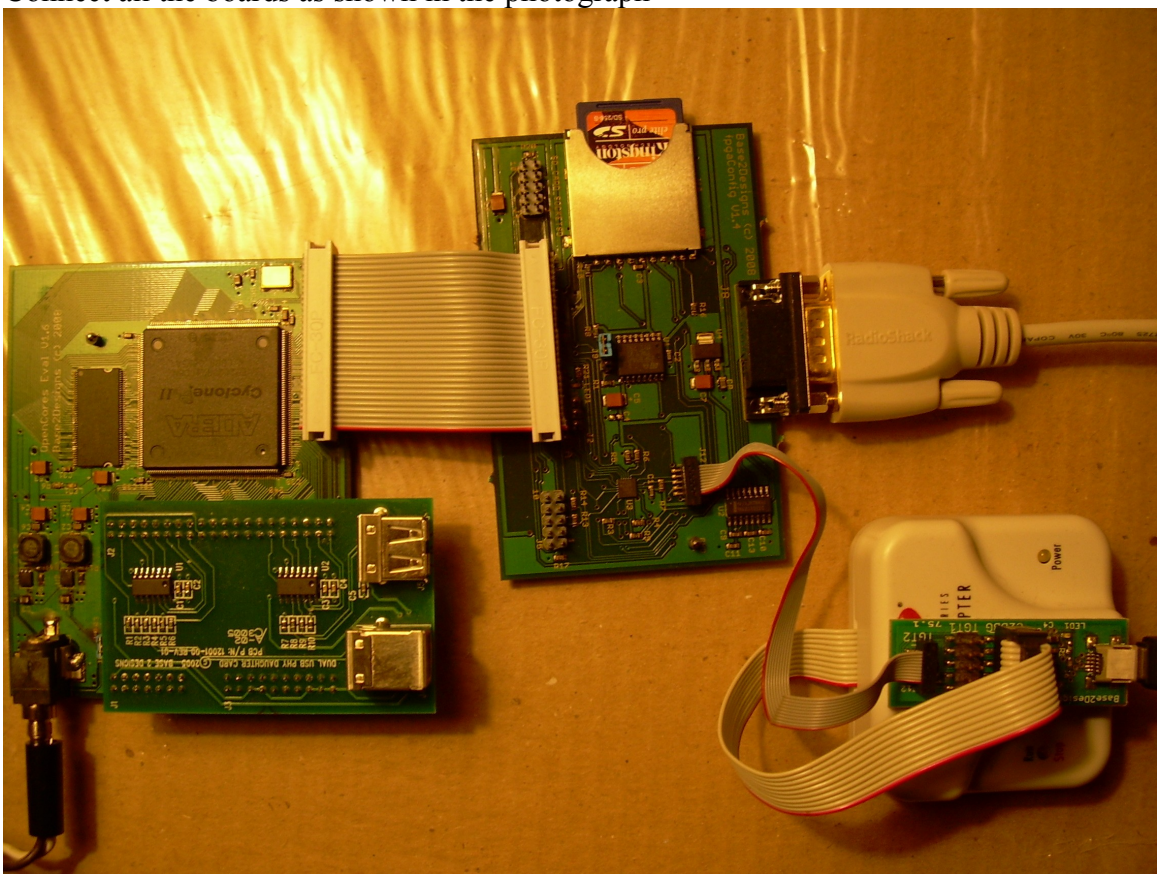
Typically FPGA development boards have sufficient logic resources to support the development of USB hardware, but they do not have the capability to support the USB electrical specification. DUSB-PHY provides two USB transceivers, each capable of supporting USB full and low speed. One transceiver is connected to a USB type A (Host) connector, and the other transceiver is connected to a USB type B (function or slave) connector. The daughter card connects to FPGA development boards using the Santa Cruz Connector format (three 0.1" pitch connectors), which is supported by Altera, and other development board vendors such as Microtronix.

DUSB-PHY can support any USB hardware IP that has a USB transceiver interface designed for simple USB transceivers such as the Fairchild USB1T11A. However, it is specifically designed to support usbhostslave, which is an Opencores USB 1.1 Host and Function IP core.

2

Getting Started

Connect all the boards as shown in the photograph



Connect the FPGA Support Board RS-232 port to a PC. You may have to use a RS-232 to USB adapter if your PC does not have a RS-232 serial port. On the PC use a terminal program such as Windows Hyperterminal, and configure the attached serial port for 11520 baud, 8 data bits, 1 stop bit, no parity, no flow control.

Connect the USB2Flash adapter to the PC using a type A to mini B USB cable. On the PC install USB2Flash console application fpgaConfig, either from the Base2Designs install disk or from:

http://www.base2designs.com/downloads/fpgaConfig_v1_1.zip

Load and run USB loop back test

Connect a Type A to type B USB cable (most common USB cable) between the type A and type B USB connectors on the DUSB-PHY board.

From:

cyc2-openrisc\progFiles\2008_11_02\bootLoader

Execute:

downloadFPGAimage.bat

This copies the binary FPGA configuration file cyc_or12_mini_top_memTest.rbf to flash memory at address 0, and forces the FPGA to be reconfigured, thus copying the new file from flash to the FPGA. You should see the following at the terminal screen:

Execute:

downloadSoftware_USBLoopBackTest.bat

The batch file copies the software image file to flash memory at address 0x90000, and then force an FPGA re-configuration. After re-configuration the OpenRISC will run the bootloader program which will then copy the new software image file from flash to DRAM, jump to DRAM, and execute the new program.

At the PC terminal window, monitor the progress of the test. The test is very extensive, and runs for a long period. If there is a problem it will stop and report the error. Try removing the loopback cable, and ensure that the test stops.

Load and run USB mouse test – with OpenRISC

Connect a Type A to type B (most common USB cable) between J4 on DUSB-PHY and a PC.

From:

cyc2-openrisc\progFiles\2008_11_02\bootLoader

Execute:

downloadSoftware_USBMouse.bat

The PC should detect the new USB mouse peripheral, and you should see the Windows mouse pointer move slowly diagonally down and to the right of the screen.

Load and run USB mouse test – no OpenRISC

Connect a Type A to type B (most common USB cable) between J4 on DUSB-PHY and a PC.

From:

usbhostslave\usbDevice\progFiles\2008_11_16

Execute:

download_USB1T11_usbMouse.bat

The PC should detect the new USB mouse peripheral, and you should see the Windows mouse pointer move slowly diagonally down and to the right of the screen.

3

Building the OpenRISC Project

First you will need to install Quartus. You can download Quartus Web Edition for free:

http://www.altera.com/products/software/quartus-ii/web-edition/qts-we-index.html

Check the project settings

First make sure all the file settings are correct. If you are using an unmodified cyc2-openrisc project then the defaults should be correct.

Using a text editor open :

cyc2-openrisc/syn/cyc_or12_mini_top.qsf

Ensure that onchip_ram_loadRAM is slected:

 # set_global_assignment -name VERILOG_FILE ../rtl/mem_if/onchip_ram_loadRAM.v

set_global_assignment -name VERILOG_FILE ../rtl/mem_if/onchip_ram_memTest.v

Next check that cyc_or12_mini_top.v is top level module:

set_global_assignment -name VERILOG_FILE ../rtl/top/cyc_or12_mini_top.v

 # set_global_assignment -name VERILOG_FILE ../rtl/top/cyc_or12_mini_top_sdCard.v

Open in a text editor:

cyc2-openrisc /rtl/top/cyc_or12_defines.v

Ensure that

`define PHY_ISP1105

is commented out

Build the project

Now open Quartus, and load the project file:

File >> Open Project

browse to cyc2-openrisc/syn/cyc_or12_mini_top.qpf

Build the project:

Processing >> Start Compilation

Once compilation is complete, create the binary programming file

File >> Convert Programming Files...

In the new window, select

Open Conversion Setup Data...

and browse to syn/cyc_or12_mini_top.cof

Now select:

Generate

This will generate cyc_or12_mini_top.rbf

Execute syn/download.bat , to download the newly created programming file to flash memory.

Build the software

First you will need to load Cygwin. You will need several of the development tools that are available with Cygwin, and also the OpenRisc tool chain.

There are two options for installing Cygwin. One is to install from the internet, and the other is to install from a local directory. If you install from the internet, get Setup from;

http://www.cygwin.com/setup.exe

When you run setup you will be presented with options on the modules that you can install. You will need to install several of the Development tools. Although a large install, the easiest option is simply to install all the Development tools.

Alternatively you can run setup from the Base2Designs installation disk, and select the "install from directory" option.

Next you need to install the Openrisc tool chain:

or32-uclinux-2003-04-13.cygwin.tar.gz

Unzip the files to your Cygwin install directory, c:/cygwin by default.

Add the install path for the new tools to your .bash_profile. C:/cygwin/home/myName by default. Try to use a plain text editor such as vim, as Wordpad can introduce unwanted formating that will confuse Cygwin.

Set PATH to or32 tools

PATH=$PATH:/opt/or32-uclinux/bin

Start cygwin, and change directory to the install path for cyc2-openrisc/sw. You will need to cd to cygdrive/c to get at your c: drive root. Eg

cd /cygdrive/c/myProjects/cyc2-openrisc/sw/usbLoopBackTest

Now rebuild the project

make clean all

and download to flash memory

./download.bat

Now try re-building the USB mouse project, and downloading to flash.

cd ../usbMouse

make clean all

./download.bat

4

Building the usbDevice Project

Using a text editor, open:

usbhostslave/usbDevice/RTL/usbDeviceAlteraTop.v

check to ensure that the following line is commented out:

`define PHY_ISP1105

Open Quartus, and load the project file:

File >> Open Project

browse to:

usbhostslave/usbDevice/syn/altera/usbDeviceAlteraTop.qpf

Build the project:

Processing >> Start Compilation

Once compilation is complete, create the binary programming file

File >> Convert Programming Files...

In the new window, select

Open Conversion Setup Data...

and browse to

usbhostslave/usbDevice/syn/altera/syn/usbDeviceAlteraTop.cof

Now select:

Generate

This will generate usbDeviceAlteraTop.rbf

Connect a Type A to type B (most common USB cable) between J4 on DUSB-PHY and a PC.

Execute download.bat , to download the newly created programming file to flash memory.

You should see the PC mouse pointer moving slowly across the screen.

5 Simulation

Install simulator and waveform viewer

First install Icarus Verilog and GTKwave. You can get the Icarus install file from either Base2Designs install disk, or from:

ftp://icarus.com/pub/eda/verilog/v0.8/Windows

or

http://www.base2designs.com/downloads/iverilog-0.8-setup.exe

You can get GTKWave from the Base2Designs intsall disk, or from:

http://www.base2designs.com/downloads/GTKwave.msi

Build and run the usbhostslave simulation

From the sim directory, build the project. There are two different builds which select different top level modules:

build_icarus_comboHostSlave.bat

or

build_icarus_sepHostSlave.bat

and run the simulation

run_icarus.bat

At the console window you should see the following:

VCD info: dumpfile wave.vcd opened for output.

Host Version number = 0x20

Slave Version number = 0x20

Testing host register read/write --- PASSED

Testing slave register read/write --- PASSED

Testing register reset --- PASSED

Configure host and slave mode. Connect full speed --- PASSED

Cancel interrupts --- PASSED

Disconnect --- PASSED

Connect full speed --- PASSED

Host forcing reset --- PASSED

Connect full speed --- PASSED

Trans test: Device address = 0x00, 2 byte SETUP transaction to Endpoint 0. Checking receive data --- PASSED

Trans test: Device address = 0x5a, 20 byte OUT DATA0 transaction to Endpoint 1. Checking receive data --- PASSED

Trans test: Device address = 0x01, 2 byte IN transaction to Endpoint 2. Checking receive data --- PASSED

Finished all tests

View the waveforms:

viewWave.bat

From the GTKWave application;

Search >> Signal Search Tree

and browse design hierarchy and select the signals you wish to view.

Build and run the usbDevice simulation

From the usbDevice/sim directory, build the project:

build_icarus.bat

and run the simulation:

run_icarus.bat

View the waveforms:

viewWave.bat

From the GTKWave application;

Search >> Signal Search Tree

and browse design hierarchy and select the signals you wish to view.

6 Connectors

Table 1: USB Connectors

Signal	Pin No.	Dir	Description
J5 – USB Type A			
+5V	1	output	+5V supply
D-	2	inout	Data minus
D+	3	inout	Data plus
GND	4		Ground
J4 – USB Type B			
No connect	1	output	USB +5V input. Not used
D-	2	inout	Data minus
D+	3	inout	Data plus
GND	4		Ground

Table 2: Santa Cruz Connector

Signal	Santa Cruz Signal	Pin No.	Dir	Description
	SC_P_CLK	J3-11		
	SC_RST_N	J2-1		
	SC_CS_N	J2-38		
FUNCTION_SPD	SC_P0	J2-3	input	Fuction speed. Controls PHY edge transition rate. High = full speed, low = low speed.
	SC_P1	J2-4		
FUNCTION_RX_VM	SC_P2	J2-5	output	Function RX data minus. Gated version of USB D-
	SC_P3	J2-6		
FUNCTION_RX_VP	SC_P4	J2-7	output	Function RX data plus. Gated version of USB D+
	SC_P5	J2-8		
FUNCTION_OE_N	SC_P6	J2-9	input	Function TX data output enable. Low = enable.

Signal	Santa Cruz Signal	Pin No.	Dir	Description
	SC_P7	J2-10		
FUNCTION_TX_VMO	SC_P8	J2-11	input	Function TX data minus. When FUNCTION_OE_N is low, this signal drives USB D-
	SC_P9	J2-12		
FUNCTION_TX_VPO	SC_P10	J2-13	input	Function TX data plus. When FUNCTION_OE_N is low, this signal drives USB D+
	SC_P11	J2-14		
DPLUS_DEF_CTRL	SC_P12	J2-15	input	Enable Function D+ 1.5K pull-up resistor. High = enable. D+ pull-up active, and D- pull-up inactive puts USB bus in Full speed mode.
	SC_P13	J2-16		
DMINUS_DEF_CTRL	SC_P14	J2-17	input	Enable Function D- 1.5K pull-up resistor. High = enable. D+ pull-up inactive, and D- pull-up inactive puts USB bus in Full speed mode.
	SC_P15	J2-18		
	SC_P16	J2-21		
	SC_P17	J2-23		
	SC_P18	J2-25		
	SC_P19	J2-27		
	SC_P20	J2-28		
HOST_SPD	SC_P21	J2-29	input	Host speed. Controls PHY edge transition rate. High = full speed, low = low speed.
HOST_RX_VM	SC_P22	J2-31	output	Host RX data minus. Gated version of USB D-
	SC_P23	J2-32		
HOST_RX_VP	SC_P24	J2-33	output	Host RX data plus. Gated version of USB D+
HOST_OE_N	SC_P25	J2-35	input	Host TX data output enable. Low = enable.
	SC_P26	J2-36		
HOST_TX_VMO	SC_P27	J2-37	input	Host TX data minus. When FUNCTION_OE_N is low, this signal drives USB D-.
HOST_TX_VPO	SC_P28	J2-39	input	Host TX data plus. When FUNCTION_OE_N is low, this signal drives USB D+
	SC_P29	J1-4		
	SC_P30	J1-5		
	SC_P31	J1-6		
	SC_P32	J1-7		

Signal	Santa Cruz Signal	Pin No.	Dir	Description
	SC_P33	J1-8		
	SC_P34	J1-9		
	SC_P35	J1-10		
	SC_P36	J1-11		
	SC_P37	J1-12		
	SC_P38	J1-13		
	SC_P39	J1-14		
+5V	+5V	J1-2	input	+5V supply. Only used for USB +5V supply.
+3.3V	+3.3V	J3-5, 7, 15, 17, 19	input	+3.3V supply. Main supply for USB PHYs.
GND	GND	J1-1 J2-2, 19, 22, 24, 26, 30, 40 J3-2, 4, 6, 8, 10, 12, 14, 16, 18, 20		Ground