Parallel Cable III Emulator for the XSA Board

March 17, 2004 (Version 1.1) Application Note by D. Vanden Bout

Summary

This application note shows how to configure the XC9572XL CPLD on the XSA Board so its parallel port interface emulates the functions of the Xilinx Parallel Cable III. This lets you use the Xilinx WebPACK JTAG tools with the XSA Board through its simple 25-wire downloading cable.

Why Emulate the Parallel Cable III?

Xilinx WebPACK software contains tools for downloading and testing SpartanII FPGAs through their their JTAG interface. One of the ways these tools access the FPGA is through a Parallel Cable III connected from the FPGA to the parallel port of a PC. A schematic for the Parallel Cable III (henceforth referred to as PCBLIII) is shown in Figure 1.

The SpartanII FPGA on the XSA Board is accessed from the PC parallel port through a simple 25-wire cable that connects to an XC9572XL CPLD on the XSA Board. The CPLD can be programmed to pass signals from the parallel port to and from the JTAG pins of the SpartanII device. The XSA Board is supplied with a default CPLD configuration that lets you download bitstreams to the SpartanII using the GXSLOAD utility provided by XESS.

This application note describes an alternate circuit that allows the XSA Board CPLD to emulate the JTAG functions of the PCBLIII. By loading this circuit into the CPLD, you can use all the Xilinx downloading and testing tools with the XSA Board through the simple downloading cable provided by XESS.

VHDL for the Parallel Cable III Emulator

Listing 1 shows the VHDL code for the PCBLIII emulator that is programmed into the XC9572XL CPLD on the XSA Board. This interface provides two functions:

- It transfers configuration bitstreams from the PC to the SpartanII FPGA using the JTAG interface.
- After the SpartanII FPGA is configured and its DONE pin goes high, the JTAG interface is used to readback and/or test the FPGA.

How the VHDL implements these functions is described below.

Line 39 disables the Flash RAM until after the FPGA is configured. The clock signal from the programmable oscillator is passed to a dedicated clock pin of the SpartanII on line 42.

Line 45 outputs a high logic level to status pin S3 of the parallel port. The Xilinx software checks for a high level on this status pin which indicates that power is being supplied to the PCBLIII.

The Xilinx software also checks for the presence of the PCBLIII by looping a signal from parallel port data pin D6 back through two of the status pins S5 and S7. Line 50 handles the loop from D6 back to S5. The CPLD cannot pass D6 to S7, however, because its TDO pin is already attached to S7. Therefore, the shunt on jumper J9 has to be moved to the **xi** position to manually connect D6 and S7.

Line 52 drives the mode pins of the SpartanII FPGA to set it in the slave-serial configuration mode. This doesn't really do anything since the SpartanII will be programmed through its JTAG interface. Line 53 uses a pullup on the CPLD pin to hold the /PROGRAM pin of the SpartanII at a high logic level, thus preventing accidental erasure of the FPGA configuration.

Lines 57–59 connect parallel port data pins D2, D1, and D0 to the SpartanII TMS, TCK, and TDI JTAG pins if D3 is low. The TMS, TCK and TDI inputs are allowed to float when D3 is high.

Line 62 passes the FPGA JTAG TDO signal back to the PC through status pin S4 of the paralle port. S4 is driven low when data pin D4 is low.

The PCBLIII emulator I/O pin assignments for the CPLD on the XSA Board are shown in Listing 2.

Using the Parallel Cable III Emulator

First, connect the XSA Board to the parallel port of a PC through the simple 25-wire cable provided by XESS. Make sure the shunt on jumper J9 is in the **xs** position. Then download the piiijtag.svf file into the XC9572XL CPLD using the GXSLOAD tool from XESS. Now move the shunt on J9 to the **xi** position. At this point, the bitstream downloading portion of the PCBLIII emulator is active.

Next, double-click the Configure Device (iMPACT) icon in the Process pane of the WebPACK **Project Navigator** window. (This discussion assumes you already have a SpartanII design synthesized and implemented in WebPACK.) The **iMPACT** window will appear. Within a few seconds the XSA Board will be probed and the JTAG chain consisting of a single XC2S100 FPGA will be detected.

In the lower pane of the **iMPACT** window, click on the xc2s100 object to select this device as the target for configuration. Then select the Operations➔Program... menu item and the **Program Options** window will appear. Click on the OK button and the bitstream for your design will download into the SpartanII FPGA on the XSA Board.

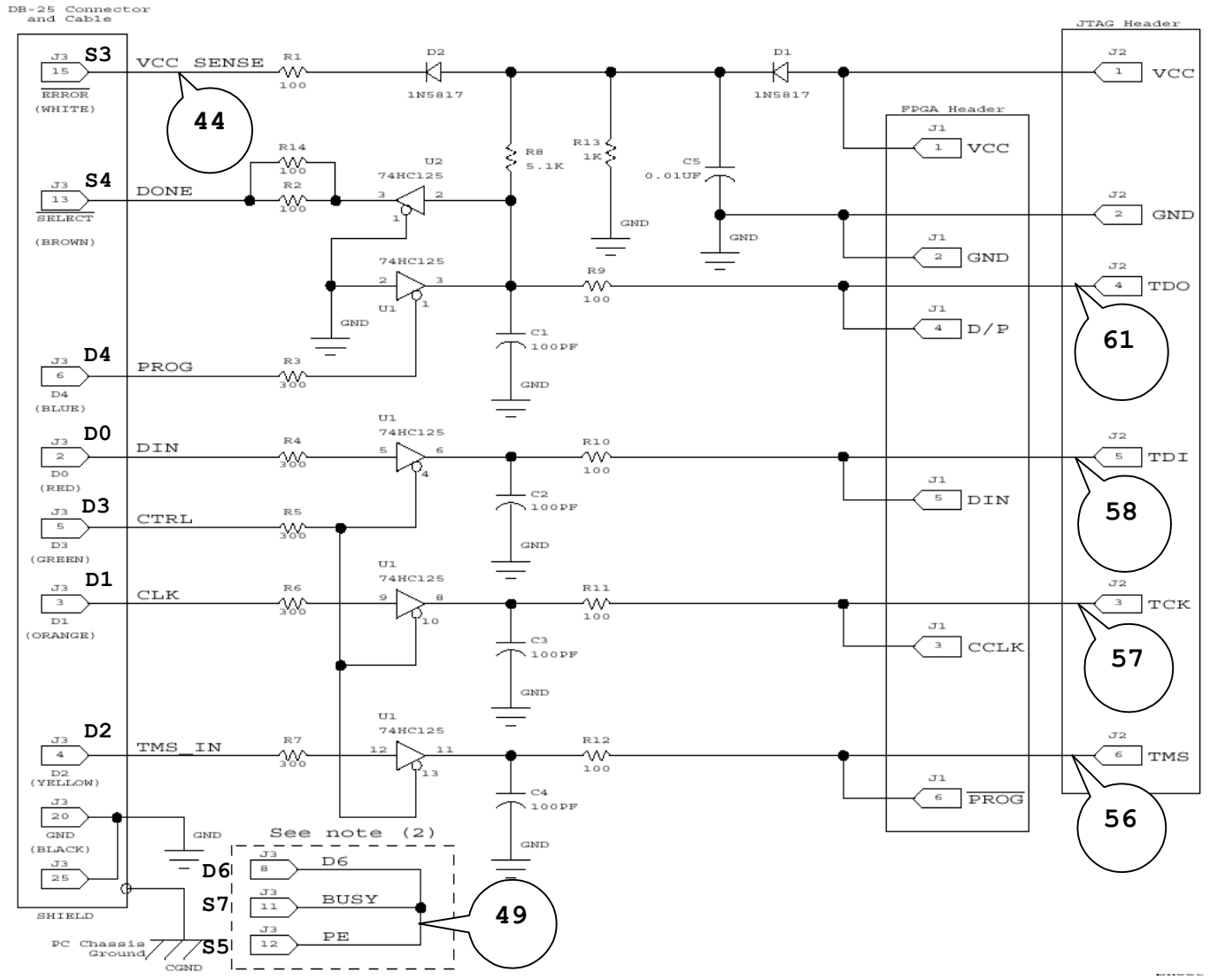

Figure 1: Xilinx Parallel Cable III schematic. The line numbers of the VHDL code in Listing 1 associated with each schematic element are shown.

Listing 1: VHDL code for the Parallel Cable III emulator.

```
library ieee;
use ieee.std_logic_1164.all;

entity piii is
   port(
      -- parallel port data and status pins
      ppd:         in std_logic_vector(6 downto 0);
      pps:         out  std_logic_vector(5 downto 3);

      -- programmable oscillator
      clk: in std_logic;

      -- Spartan2 FPGA pins
      S2_tck:      out  std_logic; -- driver to Spartan2 JTAG clock
      S2_tms:      out  std_logic; -- driver to Spartan2 JTAG mode input
      S2_tdi:      out  std_logic; -- driver to Spartan2 JTAG serial data input
      S2_tdo:      in std_logic; -- input from Spartan2 JTAG serial data output
      S2_cclk:   out  std_logic; -- driver to Spartan2 config clock
      S2_prog_n: out  std_logic; -- driver to Spartan2 /PROGRAM pin
      S2_done:   in std_logic; -- input from Spartan2 DONE pin
      S2_din:      out  std_logic; -- driver to Spartan2 config serial data input
      S2_m:        out  std_logic_vector(0 downto 0); -- Spartan2 config mode pins
      S2_clk:      out std_logic;  -- clock output to Spartan2

      fce_n:       out  std_logic    -- Flash chip-enable
   );
end piii;

architecture arch of piii is
   constant NO: std_logic := '0';
   constant YES: std_logic := '1';
   constant LO: std_logic := '0';
   constant HI: std_logic := '1';
   constant SLAVE_SERIAL_MODE: std_logic_vector(0 downto 0) := "1";
   component pullup port(O: out std_logic); end component;
begin

   -- disable Flash chip when Spartan2 is not configured
   fce_n     <= HI when S2_done=NO else 'Z';

   -- send the programmable oscillator clock to the Spartan2
   S2_clk    <= clk;

   -- the XSA power status is sent back through the parallel port status pin 3
   pps(3)    <= HI;     -- tell the PC that the VCC for the XSA board is OK
   -- the cable is detected by sending data through data pin 6 and returning
   -- it on status pins 5 and 7.  Status pin 7 is used by the JTAG TDO
   -- pin of the XC9500 CPLD on the XSA Board, so place a shunt at position "xi"
   -- of jumper J9 to make this connection.
   pps(5)    <= ppd(6);

   S2_m      <= SLAVE_SERIAL_MODE; -- set Spartan2 config mode pins
   u0: pullup port map(O=>S2_prog_n); -- place a pullup on the Spartan2 PROGRAM pin

   -- drive the Spartan2 JTAG pins from the parallel port when tristate
   -- control pin (parallel port data pin 3) is low.
   S2_tms    <= ppd(2) when ppd(3)=LO else 'Z';
   S2_tck    <= ppd(1) when ppd(3)=LO else 'Z';
   S2_tdi    <= ppd(0) when ppd(3)=LO else 'Z';

   -- the JTAG TDO output is sent back through the status pin
   pps(4)    <= S2_tdo when ppd(4)=HI else LO;
```

```
63  end arch;
```

Listing 2: User-constraint file for CPLD pin assignments.

```
#
# pin assignments for the XC9572XL CPLD chip on the XSA Board
#

# Spartan2 FPGA connections to CPLD
net S2_clk    loc=p42;
net S2_tck    loc=p13;
# net S2_dout   loc=p18;
net S2_tms    loc=p18;
# net S2_din    loc=p2;
# net S2_wr_n   loc=p19;
net S2_tdo    loc=p19;
# net S2_cs_n   loc=p15;
net S2_tdi    loc=p15;
# net S2_init_n    loc=p38;
net S2_done     loc=p40;
net S2_prog_n   loc=p39;
# net S2_cclk   loc=p16;
net S2_m<0>     loc=p36;
# net S2_d<0>   loc=p2;
# net S2_d<1>   loc=p4;
# net S2_d<2>   loc=p5;
# net S2_d<3>   loc=p6;
# net S2_d<4>   loc=p7;
# net S2_d<5>   loc=p8;
# net S2_d<6>   loc=p9;
# net S2_d<7>   loc=p10;

# Flash RAM
# net fd<0>     loc=p2;
# net fd<1>     loc=p4;
# net fd<2>     loc=p5;
# net fd<3>     loc=p6;
# net fd<4>     loc=p7;
# net fd<5>     loc=p8;
# net fd<6>     loc=p9;
# net fd<7>     loc=p10;
# net fa<0>     loc=p1;
# net fa<1>     loc=p64;
# net fa<2>     loc=p63;
# net fa<3>     loc=p62;
# net fa<4>     loc=p61;
# net fa<5>     loc=p60;
# net fa<6>     loc=p59;
# net fa<7>     loc=p58;
# net fa<8>     loc=p45;
# net fa<9>     loc=p44;
# net fa<10>    loc=p57;
# net fa<11>    loc=p43;
# net fa<12>    loc=p56;
# net fa<13>    loc=p46;
# net fa<14>    loc=p47;
# net fa<15>    loc=p52;
# net fa<16>    loc=p51;
# net fa<17>    loc=p48;
# net frst_n    loc=p50;# Flash reset
# net foe_n     loc=p12;# Flash output-enable
# net fwe_n     loc=p49;# Flash write-enable
net fce_n    loc=p11;# Flash chip-enable

# DIP and pushbutton switches
# net dipsw<1>     loc=p47;
```

```
63  # net dipsw<2>     loc=p52;
64  # net dipsw<3>     loc=p51;
65  # net dipsw<4>     loc=p48;
66
67  # 7-segment LEDs
68  # net s<0>    loc=p10;
69  # net s<1>    loc=p2;
70  # net s<2>    loc=p9;
71  # net s<3>    loc=p8;
72  # net s<4>    loc=p5;
73  # net s<5>    loc=p7;
74  # net s<6>    loc=p6;
75  # net dp   loc=p4;
76
77  # programmable oscillator
78  net clk       loc=p17;
79
80  # parallel port
81  net ppd<0>       loc=p33;
82  net ppd<1>       loc=p32;
83  net ppd<2>    loc=p31;
84  net ppd<3>    loc=p27;
85  net ppd<4>    loc=p25;
86  # net ppd<5>    loc=p24;
87  net ppd<6>    loc=p23;
88  # net ppd<7>    loc=p22;
89  net pps<3>    loc=p34;
90  net pps<4>    loc=p20;
91  net pps<5>    loc=p35;
92
```