Parallel Cable III Emulator for the XSV Board

June 1, 2002 (Version 1.1) Application Note by D. Vanden Bout

Summary

This application note shows how to configure the XC95108 CPLD on the XSV Board so its parallel port interface emulates the functions of the Xilinx Parallel Cable III. This lets you use the Xilinx Foundation downloading and JTAG tools with the XSV Board through its simple 25-wire downloading cable.

Why Emulate the Parallel Cable III?

Xilinx Foundation software contains tools for downloading and testing Virtex FPGAs through their dedicated programming pins and 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 Virtex FPGA on the XSV Board is accessed from the PC parallel port through a simple 25-wire cable that connects to an XC95108 CPLD on the XSV Board. The connections of the CPLD to the configuration and JTAG pins of the Virtex device are shown in Figure 2. The XSV Board is supplied with a default CPLD configuration that lets you download bitstreams to the Virtex using the GXSLOAD utility provided by XESS.

This application note describes an alternate circuit that allows the XSV Board CPLD to emulate the functions of the PCBLIII. By loading this circuit into the CPLD, you can use all the Xilinx downloading and testing tools with the XSV 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 XC95108 CPLD on the XSV Board. This interface provides two functions:

- It transfers configuration bitstreams from the PC to the Virtex FPGA using the slave-serial mode.
- After the Virtex FPGA is configured and its DONE pin goes high, the JTAG interface becomes active and can be used to readback and/or test the FPGA.

How the VHDL implements these functions is described below.

Line 38 disables the Ethernet PHY and video decoder chips on XSV Board so they cannot interfere with the configuration of the Virtex device. The reset is released once the Virtex FPGA configuration process is done.

Line 39 disables the Flash RAM even after the FPGA is configured because the data outputs of the Flash (D0–D3) can interfere with Virtex JTAG signals (TCK, TMS, TDI, and TDO). Note that pin 170 of the Virtex also connects to the Flash chip-enable input, so this FPGA pin cannot be used as an output due to contention.

Line 42 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 47 handles the loop from D6 back to S5. Status pin S5 is used by Foundation to detect the presence of a cable that supports downloading of configuration data to the dedicated programming pins of the FPGA.

Line 48 loops the signal from D6 to status pin S7 to indicate that the cable also supports a JTAG interface. But the CPLD also sends its TDO signal through S7 when the CPLD is being reprogrammed through its own JTAG interface, so this loop is broken

when the Virtex is not configured. If this were not done, then the contention between the two CPLD pins would make it impossible to reprogram the CPLD with a new bitstream. This also means that the JTAG interface to the Virtex FPGA is not available until the FPGA is configured with a bitstream. This works with the programmer software in Xilinx Foundation 2.1i and 3.1i.

Line 48 causes problems when the emulator is used with the newer iMPACT programmer software (i.e., the software will be unable to detect the cable). You need to modify line 48 of the VHDL so it reads as follows:

```
pps(7) <= ppd(6);
```

(Note that if you do this and then try to change the CPLD interface using the XSTOOLS 4.0 software, it will complain that it cannot read the ID code from the CPLD on the XSV Board. That's because the TDO pin of the CPLD is blocked from using pps(7) because it is already being driven by ppd(6). Just go ahead and tell the XSTOOLS software to continue even if the ID code is wrong. This allows you to re-write the CPLD with a new SVF file.)

Line 54 drives the mode pins of the Virtex FPGA to set it in the slave-serial configuration mode. Line 55 connects data pin D2 of the parallel port to the /PROGRAM pin that initiates the configuration of the Virtex FPGA. A low logic level on data pin D3 enables the drivers for all the configuration and JTAG pins as shown in Figure 1. Control of the /PROGRAM pin is passed to pushbutton SW4 on the XSV Board once the FPGA DONE signal goes high. Pushing SW4 places a low logic level on the /PROGRAM pin that erases its configuration so it can be reprogrammed. /PROGRAM cannot be left under the control of data pin D2 because D2 also controls the TMS pin of the FPGA (line 60) so a low level on TMS would erase the configuration in the FPGA. Note that pin 185 of the FPGA is also connected to SW4 so it cannot be used as an output without causing possible contention. As an alternative to using the pushbutton, power could be removed from the XSV Board to erase the FPGA and reconnect D2 to the /PROGRAM pin since DONE will go low.

Lines 56 and 57 connect data pins D1 and D0 to the CCLK and DIN programming pins of the FPGA. By default, D0 and D1 pass through inverters on the XSV Board before reaching the CPLD so they must be inverted again before being passed to the FPGA.

Lines 60–62 connect parallel port data pins D2, D1, and D0 to the Virtex TMS, TCK, and TDI JTAG pins once D3 is low and the FPGA is configured (DONE is high). The TCK input is driven low when the JTAG interface is not active to prevent any accidental activation. The TMS and TDI inputs are allowed to float. Note that FPGA pins 156 and 167 cannot be used as outputs without interfering with the TMS and TDI signals from the CPLD.

Line 67 passes the logical-AND of the FPGA configuration status and the JTAG TDO signal back to the Foundation software through status pin S4. When the FPGA is not configured, the TDO pin floats to a high level and the DONE signal appears on S4. Once the FPGA is configured and DONE is high, then the TDO signal passes through S4. S4 is driven low when data pin D4 is low as shown in Figure 1.

Line 70 causes the FPGA DONE signal to be displayed on the uppermost segment of the bargraph LED. The bar segment is illuminated when the DONE signal is high.

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

Using the Parallel Cable III Emulator

First, connect the XSV Board to the parallel port of a PC through the simple 25-wire cable provided by XESS. Then download the piii-xsv.svf file into the XSV XC95108 CPLD using the GXSLOAD tool from XESS. At this point, the bitstream downloading portion of the PCBLIII emulator is active.

Next, click-on the Programming tools icon in Foundation and select the Hardware Debugger tool. Select the Cable➔Communications... menu item in the Hardware Debugger window and set the Cable Type to Parallel. Then set the Port Name to LPT1 (or whatever parallel port the XSV Board connects to).

Once the cable is set up, select a .BIT file in the Hardware Debugger window and download it to the XSV Board. After the download is complete, the status information at the bottom of the window should show the FPGA is configured. Once this happens, the JTAG portion of the PCBLIII emulator is activated.

Now click-on the Programming tools icon but this time select the JTAG Programmer tool. Select the Output➔Cable Setup... menu item in the JTAG Programmer window and set the Communication Mode to Parallel and the Port to lpt1. Then select the Output➔Cable Reset command to initialize the JTAG

state machine in the Virtex FPGA. Now you can execute JTAG operations on the Virtex FPGA.

If you want to reconfigure the Virtex FPGA, first press pushbutton SW4 to erase the current configuration. This re-activates the bitstream downloading portion of the PCBLIII emulator and disables the JTAG interface. Then you can use the Hardware Debugger to download another bitstream.

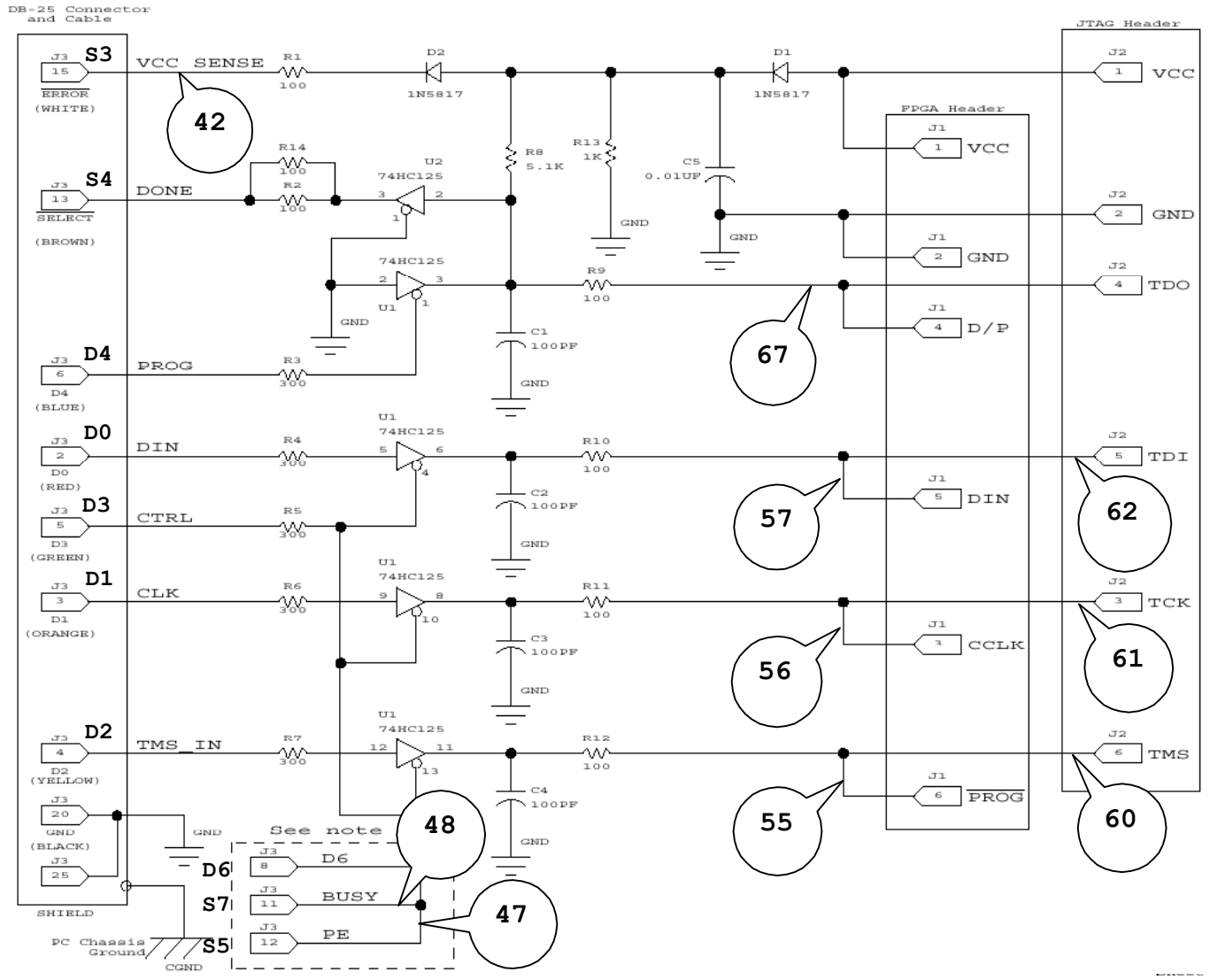

Figure 1: Xilinx Parallel Cable III schematic. The line numbers of the VHDL code in Listing 1 associated with each schematic element are shown.

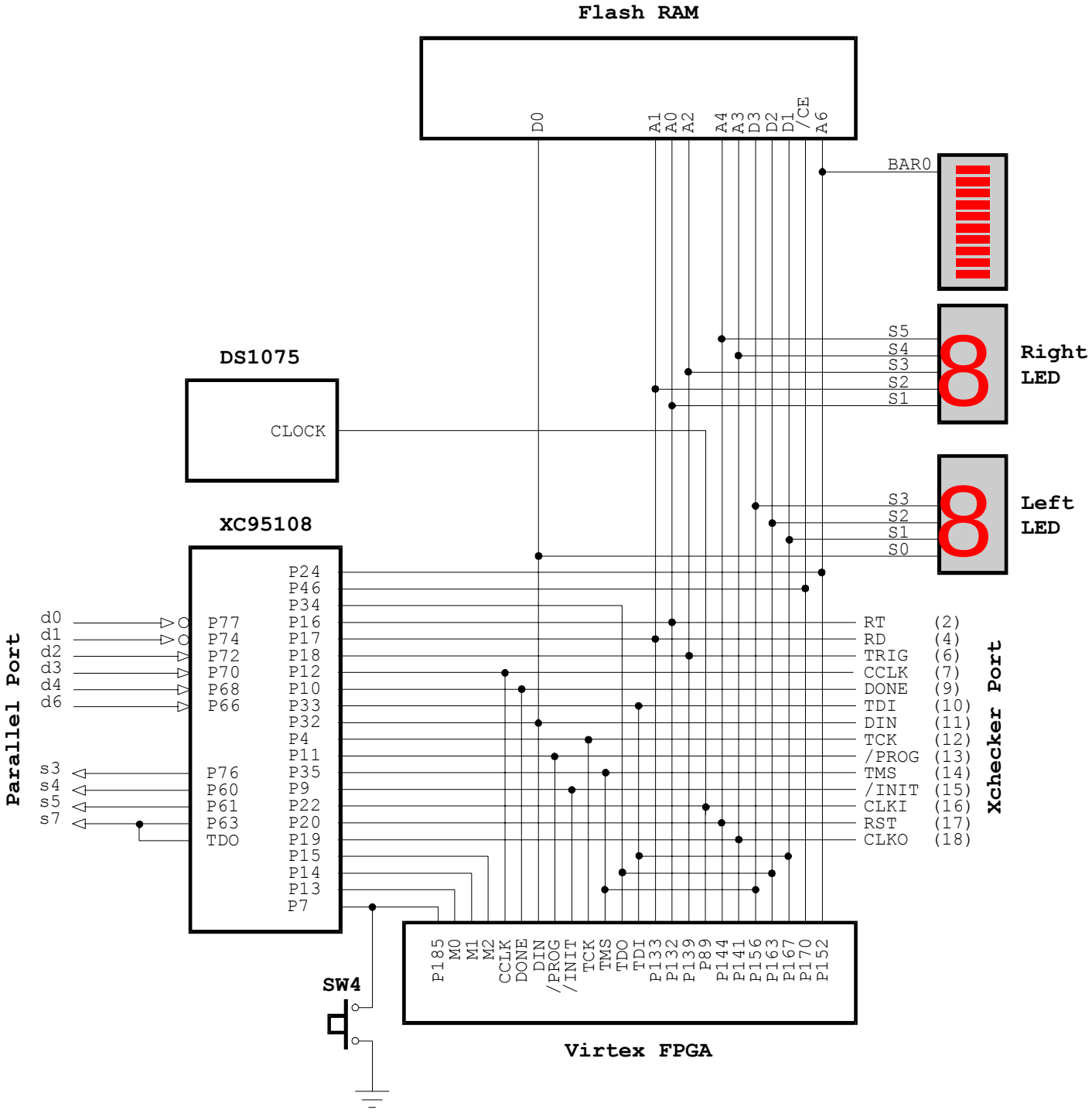

Figure 2: XSV Board CPLD connections.

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(7 downto 0);
      pps:       out   std_logic_vector(7 downto 3);

      sw4:       in    std_logic;    -- pushbutton 4

      -- Virtex FPGA pins
      V_tck:     out   std_logic; -- driver to Virtex JTAG clock
      V_tms:     out   std_logic; -- driver to Virtex JTAG mode input
      V_tdi:     out   std_logic; -- driver to Virtex JTAG serial data input
      V_tdo:     in    std_logic; -- input from Virtex JTAG serial data output
      V_cclk:    out   std_logic; -- driver to Virtex config clock
      V_prog_n:  out   std_logic; -- driver to Virtex /PROGRAM pin
      V_done:    in    std_logic; -- input from Virtex DONE pin
      V_din:     out   std_logic; -- driver to Virtex config serial data input
      V_m:       out   std_logic_vector(2 downto 0);-- Virtex config mode pins

      ce_n:      out   std_logic; -- Flash chip-enable
      reset_n:   out   std_logic; -- reset for video input and Ethernet chips
      bar:       out   std_logic_vector(9 downto 0)    -- LED bargraph
   );
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(2 downto 0) := "111";
begin

   -- disable video/Ethernet and Flash chips
   reset_n    <= LO when V_done=NO else HI;   -- disable video/Ethernet until config done
   ce_n       <= HI;                          -- disable Flash

   -- the XSV power status is sent back through the parallel port status pin 3
   pps(3)     <= HI;     -- tell the PC that the VCC for the XSV 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 also used by the JTAG TDO
   -- pin of the XC9500 CPLD on the XSV Board, so free status pin 7 when
   -- the Virtex is not configured.
   pps(5)     <= ppd(6);
   pps(7)     <= ppd(6) when V_done=YES else 'Z';  -- for iMPACT use "pps(7) <= ppd(6);"

   -- drive the Virtex configuration pins from the parallel port when tristate
   -- control pin (parallel port data pin 3) is low.  Once the Virtex is
   -- configured (DONE=1), use pushbutton SW4 to pull the /PROG pin low to
   -- allow the XSV Board to be programmed again.
   V_m        <= SLAVE_SERIAL_MODE;-- set Virtex config mode pins
   V_prog_n   <=     ppd(2) when ppd(3)=LO and V_done=NO else sw4;
   V_cclk     <= not ppd(1) when ppd(3)=LO else 'Z';
   V_din      <= not ppd(0) when ppd(3)=LO else 'Z';

   -- drive the Virtex JTAG pins once the Virtex is configured.
   V_tms      <=     ppd(2) when ppd(3)=LO and V_done=YES  else 'Z';
   V_tck      <= not ppd(1) when ppd(3)=LO and V_done=YES  else LO;
   V_tdi      <= not ppd(0) when ppd(3)=LO and V_done=YES  else 'Z';
```

```
63
64      -- send the Virtex config. status back to the PC.  Once the Virtex is
65      -- configured (DONE=1), the JTAG TDO output is sent back through the
66      -- status pin.
67      pps(4)     <= V_done and V_tdo when ppd(4)=HI else LO;
68
69      -- display status of Virtex DONE pin on the bargraph LED
70      bar(0)     <= V_done;
71
72  end arch;
```

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

```
#
# pin assignments for the XC95108 CPLD chip on the XSV Board
#

net V_tck       loc=p4;
net V_tms       loc=p35;
net V_tdi       loc=p33;
net V_tdo       loc=p34;
net V_din       loc=p32;
net V_done      loc=p10;
net V_prog_n    loc=p11;
net V_cclk      loc=p12;
net V_m<0>      loc=p13;
net V_m<1>      loc=p14;
net V_m<2>      loc=p15;
net ce_n        loc=P46;
net sw4         loc=p7;
net bar<0>      loc=p24;
net reset_n     loc=p3;
net ppd<0>      loc=p77;
net ppd<1>      loc=p74;
net ppd<2>      loc=p72;
net ppd<3>      loc=p70;
net ppd<4>      loc=p68;
net ppd<6>      loc=p66;
net pps<3>      loc=p76;
net pps<4>      loc=p60;
net pps<5>      loc=p61;
net pps<7>      loc=p63;
```