

ECEN3002 - Homework 12

Fall, 2020

Due Wednesday November 18, 11:59pm – 15 Points

The purpose of this homework is to become familiar with JTAG and the tools available in Intel Quartus Prime.

After completing this assignment, you must email the instructor, and in the email include:

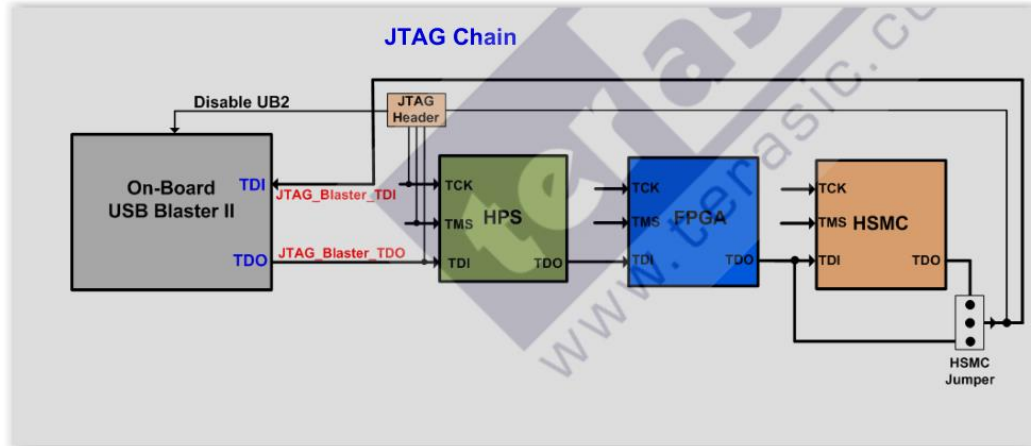
- 1. Did you complete the lab?**
- 2. Any problems you could not solve in Part 3?**

Part 1: JTAG / Cyclone V / DE10-Standard

In this class, we use the JTAG interface to program the FPGA fabric using a USB download cable. You are aware that the Cyclone V device actually contains two JTAG chain components, the Hard Processor System and the FPGA fabric.

The HPS has its own set of JTAG pins, as does the programmable portion of the FPGA. On the DE10-Standard board, the two separate sections of the device have their JTAG interfaces connected together, with the HPS first in the chain. There is no way to separate these two sections.

The DE10-Standard board also contains an HSMC (High Speed Mezzanine Connector). It is possible that a card plugged into the HSMC connector may also have a JTAG chain connection. There is a jumper on the DE10-Standard board that allows the HSMC to have the JTAG connections included in overall JTAG chain, or to have the HSMC JTAG connections bypassed. Bypass is the default, although this can be changed using a jumper on the DE10-Standard board.



- 1) Open a project in Quartus, turn on your DE10-Standard board, and open the Quartus Programmer.
- 2) If necessary, perform the steps necessary to assign a programming file to the FPGA. Notice that the SOCVHPS component shows up first in the chain diagram, as the HPS was placed first in the JTAG chain.
- 3) Right click on the SCSXFC6D6F32 component, and select Edit Properties. Write down the Checksum and JTAG User Code.

Checksum: 00AF96A8

JTAG User Code: 00AF96A8

The checksum is based on the specific programming file assigned to the FPGA, and is automatically generated by the Quartus Assembler when the .sof file is created.

The JTAG interface on the FPGA allows a user configurable code to be assigned. By default, the Quartus Assembler assigns the value of the programming file checksum. This is a good idea, as it would allow remote checking of the exact programming file version that a device contains.

- 4) Go to Quartus, Assignments > Device, Device and Pin Options. In the General category, you will see the JTAG user code. By default the Auto usercode box is checked. Uncheck this box and assign an 8 digit value for this code.
- 5) In Quartus, go to Processing > Start > Start Assembler. This will regenerate the .sof

programming file with the new JTAG usercode. Open the .sof file and you should see the new usercode near the top of the file.

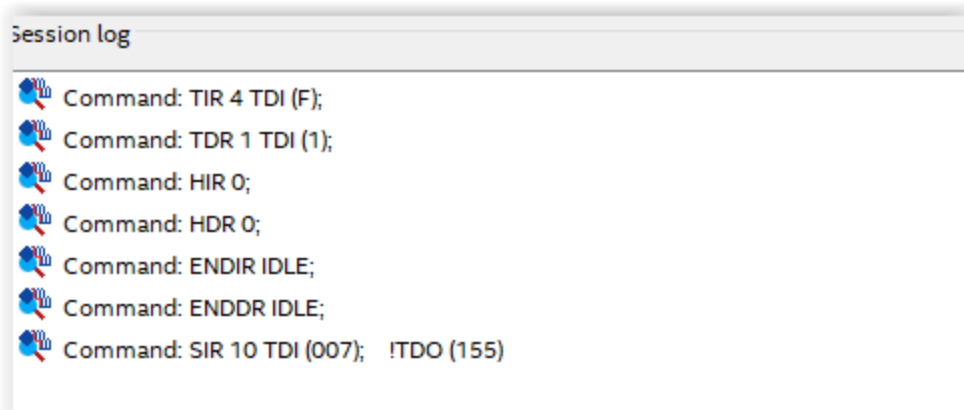
6) Program your board, and using the steps from #3 above, verify the new usercode. Didn't work? I couldn't get it to work, and eventually ran out of time trying to figure out why. I will let you know if I find out something.

Part 2: Break the JTAG Chain

- 1) In Quartus, go to Tools > JTAG Chain Debugger.
- 2) In the JTAG Chain Integrity tab, click Test JTAG Chain and confirm all is well. At this point, note that the USERCODE shown for Device2 is the same value as you saw in #3 is Part 1.
- 3) On your DE10-Standard board, find the jumper that is at the top right of the board, almost but not quite under the plastic cover. The jumper is labeled JTAG Chain is very tiny letter. Remove the jumper, noting first the current location of the jumper.
- 4) Run Test JTAG Chain again. Congratulations, you broke the chain (TDO is no longer driven). Replace the jumper and confirm all is well.

Part 3: Manually read back JTAG Information

- 1) Switch to the JTAG Chain Debugging tab. In the Device Chain pane, right click on the the rightmost device and select Activate Selected Device. Keep clicking around if the option is grayed out, it will eventually come to life. Notice that the two chips in the diagram contain the values 4 and 10, respectively. These values indicate the length of the JTAG instruction registers.
- 2) You can send specific commands to the FPGA in this mode. The specific instructions we will try are IDCODE (code = 0x006) and USERCODE (code = 0x007). Along the way you will again see that the GUI designers for FPGA tools never actually test anything.
- 3) With only the rightmost device selected, change Command: to Scan Instruction Register, set the Clocks: value to 10, and enter the value 0x007 in the TDI field. Press Run. Your session log should look like this:



If you got an error, it is because the GUI is misbehaving. In the TDI field, type 0x and notice that a zero is automatically appended to the right. The command must read 0x007 to run properly, not 0x0007 which is likely what happened. Correct the mistake and rerun.

4) change Command: to Scan Data Register, set the Clocks: value to 32, and enter the value 0xFFFFFFFF in the TDI field. Press Run. One additional line should be written to the session log, indicating a TDO value that matches the usercode.

5) Repeat these steps but use the command 0x006. You should see the device ID value of 02D020DD returned after the final step.

6) Down in the device pane, right click on the either device and select Activate All Devices. Rerun #5 above. I see a TDO value of FFFFFFFC. Now, repeat again with only the rightmost device activated to confirm everything is working.

Why do you think the test fails when both devices are activated?

Since the HPS is now in the JTAG chain (and not in bypass mode), it is necessary to add additional clocks when loading the instruction register, as the JTAG chain now sees the instruction length as 4 + 10.

7) Rerun again with the number of clocks for the Scan Instruction Register command set to 14, and the TDI value now set to 0x0006 (one extra zero). The data setting can be the same. Now you should get the correct value.

JTAG is both incredibly powerful and incredibly frustrating, especially when you are trying to do things manually. The key takeaway here is that FPGA devices rely on JTAG for many things, including configuration. You have full access to the JTAG features in the device, although unless you find yourself in a test engineering role, you may never have to deal with this level of

detail again. But, you may find yourself responsible for designing a circuit containing an FPGA, so you need to understand how connections are made.