TIE-50256 High-level Synthesis 2020

Hello LED exercise

The purpose of this exercise is to teach you to create a design in Catapult HLS ~~and port it to the PYNQ-Z1 board's FPGA~~. The design is just a simple button directed LED. You should do the Catapult Lab 1 before this exercise to learn the basics of using Catapult. There are many steps in this exercise but they are simple, so don't panic! You should be mindful about what you are doing instead of just following the instructions without thinking. After all, the purpose is to learn something from this. You can later use this exercise as a reference on the synthesis steps with other tasks.

Synthesizing RTL with Catapult

1. Choose a folder for the exercises on the P-drive and create a new subfolder "Hello_LED" for this exercise. BTW, you should always avoid white spaces in path names when using EDA software. Otherwise they may not work properly.
2. Create file "led.cpp" and copy the code below as its contents. The code reads a one-bit input and returns it. The input will be mapped to a button on the PYNQ board, and the output to a LED

```
#include <ac_int.h>
ac_int<1,false> led(ac_int<1,false> input)
{
      return input;
}
```

3. Navigate to the folder in the virtual machine terminal
4. Open Catapult HLS
5. Applying what you learned from the Catapult Lab 1:
    a. Set "led.cpp" as the input file
    b. Set the function led() as the top-level function
    c. Choose "Vivado" as RTL Synthesis Tool, "Xilinx" as Vendor, and "Artix-7" as Technology.
        i. You do not need to change the default speed or part.
        ii. They do not correspond to the actual FPGA chip on the PYNQ board, but for this simple design it does not matter.
    d. Set clock frequency to 100MHz

6. Click “Architecture” in Task Bar
    a. Select the “main” loop and check the Pipeline box to pipeline the design. Click “Apply”
        i. Pipelining will be discussed later in the course.
7. Click “RTL” in Task Bar to perform the RTL synthesis
8. After RTL synthesis finishes, you should have the verilog and VHDL files of the design in your project’s Catapult/led.v1 subfolder (concat_rtl.v for verilog and concat_rtl.vhdl for VHDL). If you wish, you can check what kind of RTL code Catapult generated for this simple design
9. You can now exit Catapult

To return: The concat_rtl.vhdl file. Add in your Moodle return comments about what kind of structures you found in the vhdl that correspond to your design.

That’s all unfortunately for this year, as we don’t have proper access to FPGA lab ☹. So you can disregard what’s below. If you have a Pynq board at home, you can do the rest but you don’t have to return anything.

Synthesizing the design on the FPGA

1. Start Vivado in Windows
2. Click “Create Project” in Quick Start
3. Name the project “Hello_LED” and select a suitable location for it on your P-drive. Click “Next”
4. Select “RTL Project” as project type. Click “Next”
5. Select the “concat_rtl.vhdl” file as a source file in your project. Click “Next”
6. Do not add any constraints. Click “Next”
7. Select “xc7z020clg400-1” as the part. This is the FPGA on the PYNQ board. Click “Next”
8. Click “Finish”
9. Click “Create Block Design” in the Flow Navigator on the left
10. Give it the name “Hello_LED” and click “OK”
11. Select the Sources tab and drag the led component to the block diagram.
    a. If dragging does not work, you can instead righ-click the component and select “Add module to block design”
12. For the ports “clk”, “rst”, “input_rsc_dat”, and “return_rsc_dat”, right-click them and select “Make external”.
    a. The other two ports can be ignored. They are handshake signals added by Catapult that are not needed in this design
13. Right-click the “Hello_LED.bd” in the Sources tab and select “Create HDL wrapper”
14. Right-click the created “Hello_LED_wrapper” in the Sources tab and select “Set as Top”
15. Click “Run Synthesis” in the Flow Navigator
16. After synthesis completes, click “Open Synthesized Design” in the Flow Navigator.

17. Select “I/O Planning” from the drop-down menu at the upper right corner, and “I/O Ports” tab at the bottom of the screen
18. Expand all the ports and select “LVCMOS18” as the I/O standard for each port.
    a. Do this even if they have been selected as default because Vivado may report an error otherwise when generating bitstream
19. Open the PYNQ-Z1 Board Reference Manual from Moodle and go to page 14. The clock source pin is indicated there as H16. Set it as the Package Pin for the clk_0 port
20. Basic I/O pins are listed on the next page. Connect:
    i. Switch 0 to drive the reset
    ii. Button 0 to input
    iii. Led 0 to output
21. Click “Run Implementation” in the Flow Navigator. This is the place & route step. When asked, save XDC file as hello_LED.xdc.
22. Click “Generate Bitstream” in the Flow Manager
23. Power on the PYNQ board
24. Once bitstream has been generated, expand “Open Hardware Manager” in the Flow Manager and click “Open Target” and Auto Connect
25. Click Program Device, select the FPGA part and the generated bitstream file. Click “Program”
26. Play with the amazing design to your heart’s content! Button 0 should make the corresponding LED to blink. If nothing happens, toggle the switch 0 that was connected to the reset

~~*Return:* Demonstrate your implementation to an assistant in the lab, and as a Moodle return, just write the date of the demonstration.~~