Module 1, Assignment 1

Contents

1 Goal of the Assignment

In this assignment, you will carry out the full CAD tool flow for AMD/Xilinx FPGAs (Vivado software) and investigate the output products after each step.

You will start with describing a simple logic function in VHDL, and verify its correctness with the schematics Vivado provides and a behavioral simulation. In the next step, you will perform the I/O planning, synthesize and implement the design on the Nexys A7 board. Vivado will be able to find a valid implementation, which fits on the board and meets the timing requirements. You will then assign a custom placement and routing, which does not meet the timing requirements.

2 Task Description

2.1 Create a new Vivado Project

Start Vivado (2023.2) via these terminal commands:

```
xilinx2023.2
vivado
```

Create a new Vivado RTL project. Select VHDL as target language, choose the Nexys A7-100T board (part *xc7a100tcsg324-1*) for your project, and add the constraints file provided in PANDA.

2.2 Create a new Design

Create a new VHDL design source and implement the logic function shown in Equation 1 in a sequential logic module. The input values are captured synchronous to the rising ege of the clock and the output signal is also synchronous to the rising edge of the clock, i.e., use registers at the inputs and output of your module.

$$f(a, b) = a \wedge b \tag{1}$$

Create a new block design, add your module to it, and make the ports of your module external. For the clock, use the 100MHz clock from the board, which is available in the block design via the CLK100MHZ pin.

Hint: You can add a design source to your block design by right-clicking on your VHDL file and selecting "Add Module to Block Design".

Hint: Do not forget to create the HDL wrapper for your block design.

Draw the schematic you expect for the elaborated design:

Show your solution to your tutor.

Generate the block design, open the schematic of the elaborated design, and compare your solution to the generated schematic.

Hint: Make sure you select the synthesis option "Global" when generating the block design.

2.3 Simulate your Design

Create a test bench for your design and run a behavioral simulation.

Hint: Make sure you set the testbench source as top.

Show your solution to your tutor.

2.4 Assign the Inputs and the Output to the Package Pins

The clock pin has already been assigned to the correct pin of the FPGA package. For the elaborated design, open the window *Package Pins* and the window *I/O Ports*, and assign the following properties to your pins:

- All of your pins use the IO standard *LVCMOS33*.
- Assign a to pin *J15*.
- Assign b to pin *L16*.
- Assign the output to pin *H17*.
- Leave all other parameters at their default value.

Save the assignments in your constraints file by pressing *Ctrl + s*.

2.5 Synthesize the Design

Draw the schematic you expect for the synthesized design:

Show your solution to your tutor.

Synthesize the design, open it, and compare your schematic to the synthesized solution. Also compare the schematic after synthesis to the elaborated design. What are the differences?

After synthesis, Vivado generates several reports. Open the *Utilization report* and extract the following information:

How many LUTs (total) are available on the FPGA? _____

How many LUTs did you use as logic? _____

How many LUTs did you use as memory? _____

Is the resource utilization after synthesis accurate? _____

How many DSPs are available on the FPGA? _____

Let Vivado create the *Timing Summary* (you can use the default parameters).

What is the *worst negative slack* in your design? _____

What is the *worst hold slack* in your design? _____

2.6 Implement the Design

Implement the design.

What is the *worst negative slack* in your design? _____

What is the *worst hold slack* in your design? _____

Compare the slack values to the results from synthesis. Are the values equal to the results from synthesis? If yes, why? If no, why?

_____

_____

_____

2.7 Custom Placement and Routing

In the device view, you can see the placed and routed design. In the previous step, you saw that Vivado was able to find a valid implementation, i.e., all logic blocks have been placed on the FPGA, a congestion free routing has been found, and timing constraints are met. Modify the placement and/or routing to cause the timing constraints to fail.

Since no input or output delays have been specified: What is the delay you can focus on to destroy the timing?

Modifying the placement:

In the *Device* window, you can drag&drop the placed cells (highlighted in cyan) to new locations. Save the modified placement.

What are you effectively modifying when you modify the placement?

Modifying the routing:

Switch to the *Routing Resource* view in the *Device* window, select an appropriate routed net (highlighted in green), enter the *Assign Routing Mode*, and modify the routing of the net by adding or removing nodes. If you have added a net gap, you can let Vivado route this gap by selecting *Auto-Route* (right click on the net gap). Once you finished your routing, *Assign Routing* and make sure that you checked the *Fix Routing* checkbox. This will ensure that Vivado will not rip up the net when you re-run the implementation.

Your modifications - when saved - can be found in the constraints file.

Re-run the implementation.

What is the *worst negative slack* in your design? _____

What is the *worst hold slack* in your design? _____

2.8 Generate the Bitstream and Program the FPGA

Generate the bitstream, load it onto the FPGA, and verify the functionality on the real hardware. The two inputs of the AND gate are connected to the switches 0 and 1 on the Nexys board. The output of the gate is connected to LED 0.