

Exercise 1: The FPGA and the first logic circuits

This first exercise introduces the Vivado design flow and your first adder design. The following tasks will guide you through the process. Additional information can be found in the corresponding slide set available at StudIP.

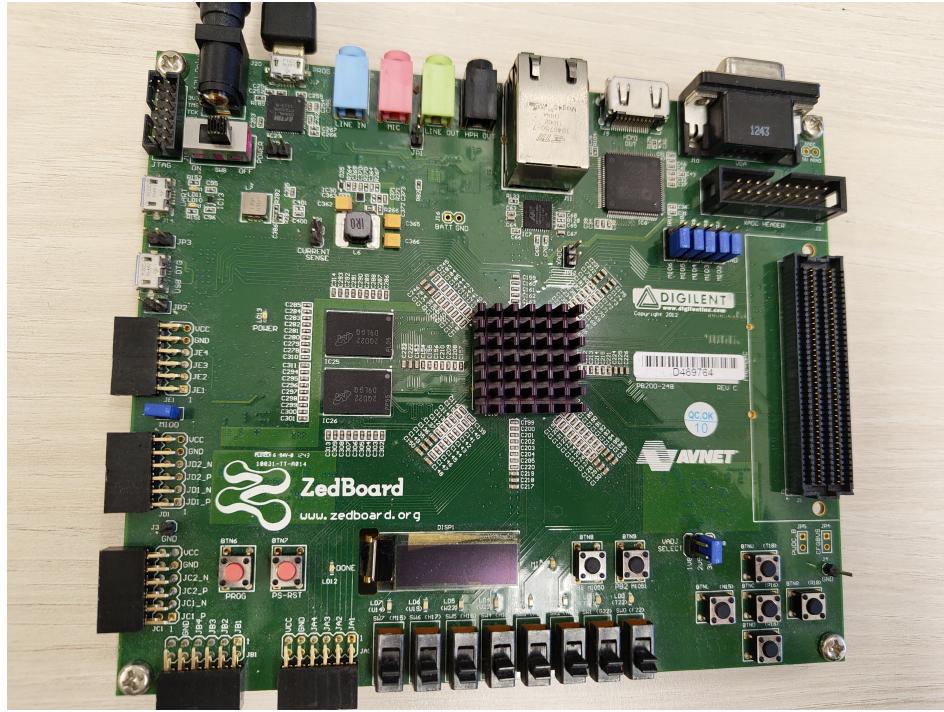


Figure 1: ZEDboard

Task 1. The goal of this task is to make you familiar with the Vivado Design flow. Attached, you find the file `and3.vhd` with the AND3 design you know from the lecture. The Following steps will guide you through the design flow.

- Create a new project and set it up for the AND3 design. The project should be an *RTL Project*, do not add any files at this point in time and choose the *ZEDboard*.
- Add the file `and3.vhd` as a *Design Source* to the project.
- Open the project settings and navigate to *Synthesis, More Options*. Enter `-mode out_of_context`. This allows you to synthesize the design without interfacing with the FPGA pins.
- Familiarize with the AND3 implementation by analyzing the VHDL.
- Elaborate the design. Compare the RTL layout to the VHDL implementation.
- Open the synthesized design and compare the Schematic to the previously explored RTL layout. What happened? How can you verify your observation?

Task 2. Continue getting familiar with the Vivado design flow. Using the further files attached, implement the AND3 architecture on the FPGA board. The Following steps will guide you through the process.

- Add the file `and3_wrapper.vhd` as a *Design Source* to the project.

- b) Add the file `constraints_zedboard.vhd` (or `_nexys_a7` or `_nexys4` depending on your FPGA) as a *Constraint* to the project.
- c) Remove the `-mode out_of_context` option from the project settings.
- d) Implement the design and generate the bitstream.
- e) Setup the ZEDboard for USB-JTAG programming.
- f) Program the FPGA and verify the design by demonstration.

The switches 0, 1 and 2 of the FPGA are connected to the inputs of your design. If their state corresponds to a logical 1, the corresponding LED will light up. The leftmost LED displays the output of your implementation in a similar fashion.

The boot mode of the FPGA is controlled using jumpers JP7 to JP11. Table 1 gives an overview over the different boot modes available on the ZEDboard.

Table 1: ZEDboard Boot Modes

Boot Mode	JP11	JP10	JP9	JP8	JP7
JTAG	0	0	0	0	0
Quad-SPI	0	1	0	0	0
SD Card	0	1	1	0	0

After the first task introduced us to the Vivado design flow, we will now explore simulation of the same AND3 design.

Task 3. Add `and3_tb.vhd` as a *Simulation Source* to your project and explore its content. Ensure that the `and3_tb` entity the top module for simulation is. Simulate the AND3 design using the provided testbench. Familiarize yourself with the waveform window. Can you verify that the design is working correctly?

Now you are familiar with the Vivado environment and you are ready for your first own design.

Task 4. Create a new project and implement the logic circuit shown below. It consists of several logic gates. There is a template for your implementation: `logic.vhd`. The template provides the general VHDL construct. Further, there is a testbench `logic_tb.vhd` and you will also need the `constraint_zedboard.xdc` (or `_nexys_a7` or `_nexys4` depending on your FPGA) file from before. Follow these steps:

- a) Implement the entity for the logic circuit. There are 4 inputs and 1 output. Name them according to the schematic.
- b) Implement the architecture for the functional behavior of the logic circuit.
- c) Compare the RTL Schematic to the given schematic.
- d) Check the functional correctness of your implementation via synthesis and the testbench provided. Ensure that the correct top module for simulation is selected. The signals `e_x` and `e_y` show you the expected values of `x` and `y` respectively. While `correct` signals if everything is fine (constant 1).
- e) Using the provided `logic_wrapper.vhd`, synthesize and implement the design. Use the resulting bitstream to program an FPGA.

Using the switches 0 - 7 on the board, you can change the status of the LEDs. Can you figure out the combination of switches that makes all LEDs (0-7) light up at the same time?

