

Laboratory 1

(Due date : September 27th)

OBJECTIVES

HARDWARE-ONLY PROJECT

- ✓ Learn the Xilinx FPGA Design Flow with the Vivado Webpack software: Synthesis, Simulation, and Bitstream Generation.
- ✓ Learn how to assign FPGA I/O pins and download the bitstream on the ZYBO Z7-10 Board.

SOFTWARE-HARDWARE PROJECT

- ✓ Create an Embedded System using the ZYBO Z7-10 Board.
- ✓ Use the Block Based Design in Vivado to instantiate the PS and the AXI GPIO (General Purpose I/O).
- ✓ Create a software application in SDK to control the LEDs (via AXI GPIO) and to print out messages via UART.

TUTORIALS

VHDL CODING

- ✓ Refer to the [Tutorial: VHDL for FPGAs](#) for a tutorial and a list of examples.

EMBEDDED SYSTEM DESIGN

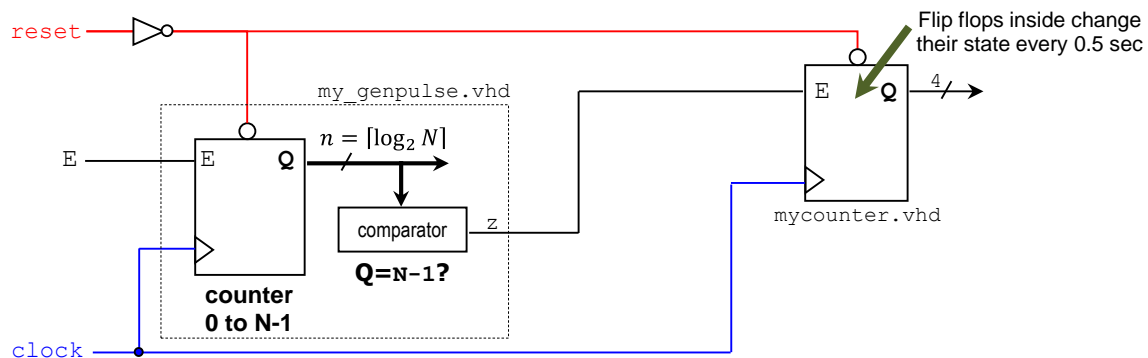
- ✓ Refer to the [Tutorial: Embedded System Design for Zynq SoC](#) for tutorials specific to the ZYBO Board.
- ✓ Refer to the Zynq Book Tutorials (www.zynqbook.com) for step-by-step instructions.

ZYBO Z7-10 BOARD SETUP

- The ZYBO Z7-10 Board can receive power from the shared UART/JTAG USB port (J12). Connect your Board to a computer via the USB cable. If it does not turn on, connect the power supply of the Board.
- ZYBO Z7-10 documentation: Available in [class website](#).

FIRST ACTIVITY (50/100)

- Design a 4-bit counter with the following count: 14, 12, 10, 8, 6, 4, 2, 0, 1, 3, 5, 7, 9, 11, 13, 15
- **Inputs:** enable, reset, clock. **Outputs:** 4-bit count (connected to LEDs).
- The count should increase every 0.5 seconds: You can use a pulse generator (VHDL code is provided) that generates a pulse where we can customize the interval of time between pulses. This pulse is fed to the enable input of the counter.
- Note that the frequency of the input clock is 125 MHz. You should set up the parameter N of the pulse generator so that it generates a pulse (of duration $1/125$ us) every 0.5 s.



▪ XILINX ZYNQ SOC DESIGN FLOW:

- ✓ Create a new Vivado Project. Select the **ZYNQ XC7Z010-1CLG400C** device.
- ✓ Write the VHDL code for the counter with enable (`mycounter.vhd`). You can use the state machine method.
- ✓ Using the structural coding approach in VHDL, instantiate the counter and the pulse generator into a top file. Synthesize your circuit (Run Synthesis).
- ✓ Write the VHDL testbench to properly test the circuit. Since N is a large number, use $N=10$ just for simulation purposes. When implementing the circuit, use the proper N value.
- ✓ Perform Functional Simulation (Run Simulation → Run Behavioral Simulation). **Demonstrate this to your instructor.**
- ✓ I/O Assignment: Create the XDC file. On the ZYBO Z7-10 Board, use LD3 to LD0 (pins D18, G14, M15, M14) for the outputs, SW0 (pin G15) for enable, BTN0 (pin K18) for reset, and CLK125 (pin K17) for the input clock.
- ✓ Implement your design (Run Implementation).

- ✓ Generate the bitstream file (Generate Bitstream).
- ✓ Download the bitstream on the ZYNQ SoC (Open Hardware Manager→ Program Device) and test. **Demonstrate this to your instructor.**
- Submit (as a .zip file) the generated files: VHDL code, VHDL testbench, and XDC file to Moodle (an assignment will be created). DO NOT submit the whole Vivado Project.

Instructor signature: _____

Date: _____

SECOND ACTIVITY (50/100)

- Create a new project in Vivado. Select the **ZYNQ XC7Z010-1CLG400** device.
- Block Design: Instantiate the Zynq PS and the AXI GPIO peripherals.
- SDK Software application:
 - ✓ Make the 4 LEDs flash as follows: 1000, 1100, 1110, 1111, 0111, 0011, 0001, 0000 (and then repeat the sequence). '1': LED on, '0': LED off.
 - ✓ Play with the software delays until you make the LED flash approximately every 0.5 s.
 - ✓ Each time a full sequence is completed, print the following message on the terminal (via UART):
'ECE4900 - LEDs ok!'.
- Download the hardware bitstream on the ZYNQ SoC.
- Launch your software application on the Zynq PS. The LEDs should be flashing properly and the messages should be appearing on the Terminal. **Demonstrate this to your instructor.**
- Submit the software routine (.c file) you created to Moodle (an assignment will be created).

Instructor signature: _____

Date: _____