Laboratory 1

(Due date: May 21st)

OBJECTIVES

HARDWARE-ONLY PROJECT (FIRST ACTIVITY)
- Describe a Digital System using VHDL.
- Learn the Xilinx FPGA Design Flow with Vivado: Synthesis, Simulation, and Bitstream Generation.
- Learn how to assign FPGA I/O pins and download the bitstream on the ZYBO Z7-10 Board (or ZYBO).

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

REFERENCE MATERIAL

VHDL CODING
- Refer to the Tutorial: VHDL for FPGAs for a tutorial and a list of examples.

EMBEDDED SYSTEM DESIGN FOR PSOCS
- Refer to the Tutorial: Embedded System Design for Zynq PSoC for tutorials specific to the ZYBO Z7-10 Board (or ZYBO)

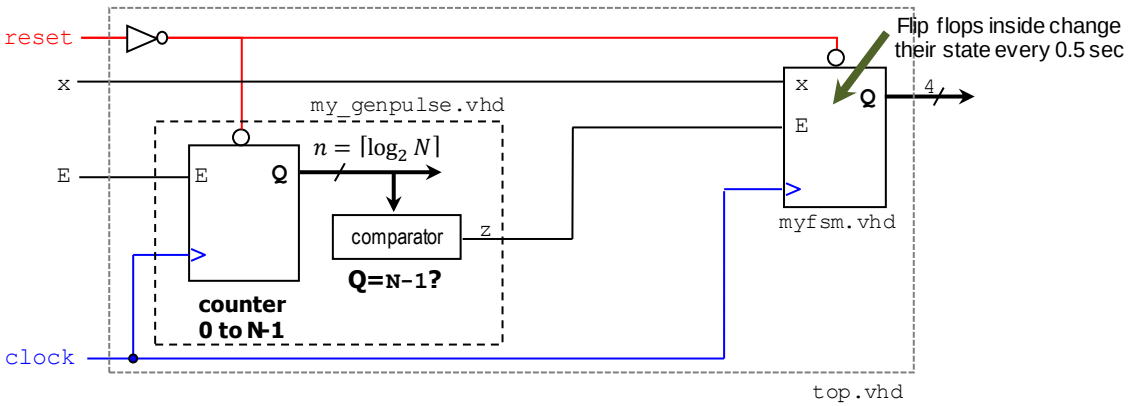
ZYBO Z7-10 BOARD SETUP
- The ZYBO Z7-10 Board can receive power from the shared UART/JTAG USB port (J12). Connect your Board to a computer via the USB cable. If it does not turn on, connect the power supply of the Board.
- ZYBO Z7-10 documentation: Available in class website.

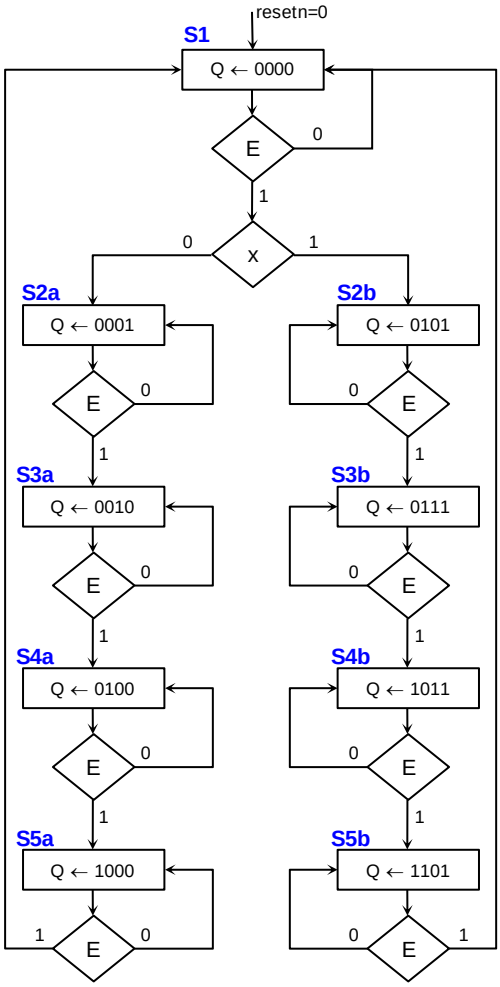
FIRST ACTIVITY (50/100)

- Refer to the Embedded System Design for Zynq PSoc Tutorial → Unit 1 for a similar activity (except for the FSM).

- **Digital System specifications**:
  - ✓ Design a circuit that generates a certain count (that changes every 0.5 seconds) depending on the value of the input x.
    - x = 0: 0,1,2, 4, 8,0,1,…
    - x = 1: 0,5,7,11,13,0,5,…
  - ✓ **Inputs**: reset (active high), E (enable), x (control), clock. **Outputs**: 4-bit count (connected to LEDs).
  - ✓ Input x: If x changes, the count sequence is altered only when the current sequence is completed.
  - ✓ Enable input (E): If E=0, the count pauses. If E=1, the count works as usual.

- **Digital System**: The circuit (FSM + Datapath) is depicted below.
  - ✓ FSM: The `myfsm.vhd` circuit generates the 4-bit count based on x. This FSM is depicted (next page).
  - ✓ Datapath: For the count to update every 0.5 s, we use a counter modulo-N (`my_genpulse.vhd`) that generates a 1-cycle pulse every N cycles (the interval of time between pulses is customizable). This pulse is fed to the input E of the FSM.
  - ✓ The top file (`top.vhd`) integrates all the components: FSM + Datapath. To support functions in the `my_genpulse.vhd` component declaration, make sure to include the library declarations: 'use ieee.math_real.log2;', 'use ieee.math_real.ceil;'.
  - ✓ ZYBO Z7-10 (or ZYBO): 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 (if you have installed the board definition files, you can also select the ZYBO or ZYBO Z7-10 board).
  - ✓ Write the VHDL code for the FSM (`myfsm.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 (`top.vhd`). Synthesize your circuit (Run Synthesis).
  - ✓ Write the VHDL testbench (`tb_top.vhd`) 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 (or ZYBO), use `LD3` to `LD0` for the outputs, `SW0` for E, `SW1` for x, `BTN0` for reset, and `CLK125` 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, as well as a screenshot of your simulation to Moodle (an assignment will be created). DO NOT submit the whole Vivado Project.

Instructor signature: _____ **Date:** _____

SECOND ACTIVITY (50/100)

- Refer to the Embedded System Design for Zynq PSoc Tutorial → Unit 2 for a similar activity.

- **Hardware/software design in Vivado**: Control 4 LEDs connected to the PL via software.

- 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):
    `ECE5736 – LEDs ok!.`

- Download the hardware bitstream on the ZYNQ PSoC.
- 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) and the XDC file you created to Moodle (an assignment will be created).

Instructor signature: _____ **Date:** _____