Laboratory 5

(Due date: **002/003**: November 11th, **004**: November 12th, **005**: November 13th)

OBJECTIVES

- ✓ Describe synchronous circuits in VHDL.
- ✓ Learn Testbench generation for synchronous circuits.

VHDL CODING

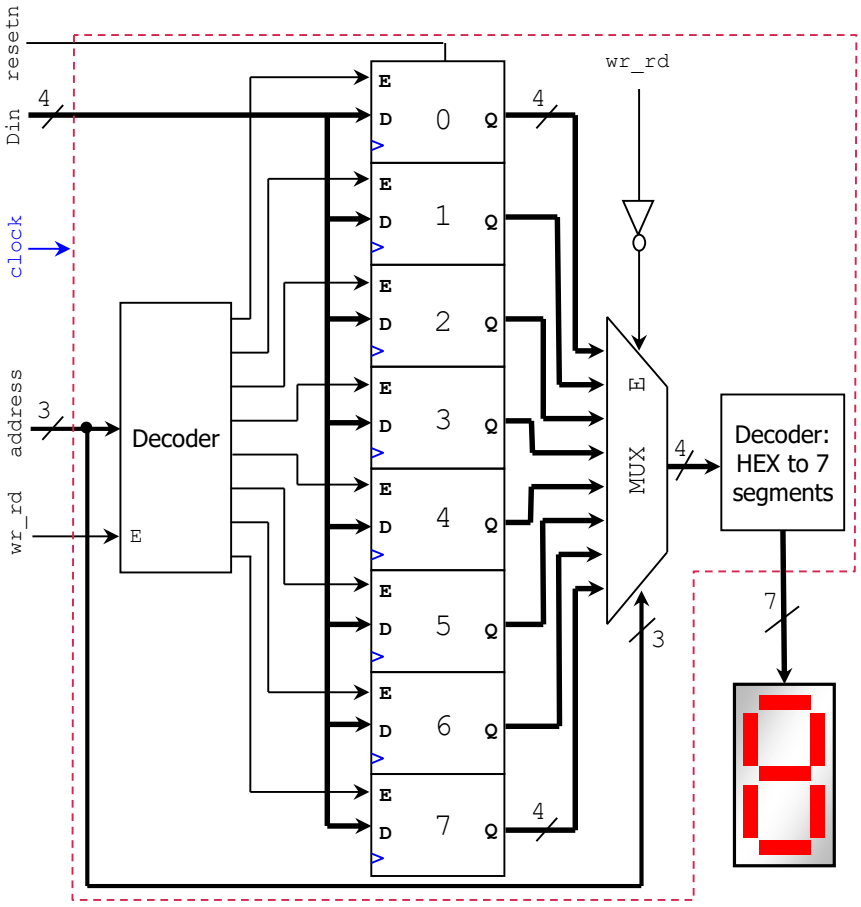
- ✓ Refer to the Tutorial: VHDL for FPGAs for a list of examples.

FIRST ACTIVITY (100/100)

- **RANDOM MEMORY ACCESS (RAM) EMULATOR:** The following circuit is a memory with 8 addresses, each address holding a 4-bit data. The memory positions are implemented by 4-bit registers. The `resetn` (active low) and `clock` signals are shared by all the registers. Data is written onto (or read from) one of the registers.

- **Memory Write** ($wr\_rd = 1$): The 4-bit input `Din` is written into one register. The `address[2..0]` signal selects the register to be written. Here, the 7-segment display must show `0`. Example: if `address= "101"`, then `Din` is written into register 5.

- **Memory Read** ($wr\_rd = 0$): The MUX output appears on the 7-segment display (hex. value). The `address[2..0]` signal selects the register from which data is read. For example, if `address= "010"`, then data in register 2 appears on the 7-segment display. If data in register 2 is `"1010"`, then the symbol '`A`' appears on the 7-segment display.



- ✓ **NEXYS A7-50T**: Create a new Vivado Project. Select the **XC7A50T-1CSG324 Artix-7 FPGA** device.
- ✓ Write the VHDL code for the given circuit. Create a separate file for i) Register with enable, ii) MUX with enable, iii) decoder with enable, iv) HEX-to-7 segments decoder, and v) top file.
- ✓ Write the VHDL testbench to test at least 8 writes (each on a different memory address), and then 8 reads (each from a different memory address). You must generate a 100 MHz input clock for your simulations.
- ✓ Perform Functional Simulation and Timing Simulation of your design. **Demonstrate this to your TA**.
- ✓ I/O Assignment: Create the XDC file. Nexys A7-50T: Use `SW0` to `SW7` for the input (`Din`, `address`, wr_rd), CLK100MHZ for the input `clock`, `BTN_RES` (CPU Reset) push-button for `resetn`, `CA-CG` (7-segment display signals) for the 7-bit output, and `AN7-AN0` (anode enable for the eight 7-segment displays; enable only one 7-segment display).
  * Note: If you are using the **Basys3 Trainer Board**, use `SW8` for $resetn$.
- ✓ Generate and download the bitstream on the FPGA and test. **Demonstrate this to your TA**.

- Submit (as a .zip file) all the generated files: VHDL code files, VHDL testbench, and XDC file to Moodle (an assignment will be created). DO NOT submit the whole Vivado Project.

TA signature: _____ **Date:** _____

Instructor: Daniel Llamocca
TAs: Ekhlass Alkharabsheh, Prakruthi Jagadeesh, Aishwarya Shitole