Laboratory 4

(Due date: **011**: October 31st, **005**: November 1st, **007**: November 2nd)

Objectives

- ✓ Use the Concurrent Description and the Structural Description in VHDL.
- ✓ Implement Combinational circuits on an FPGA.

VHDL Coding

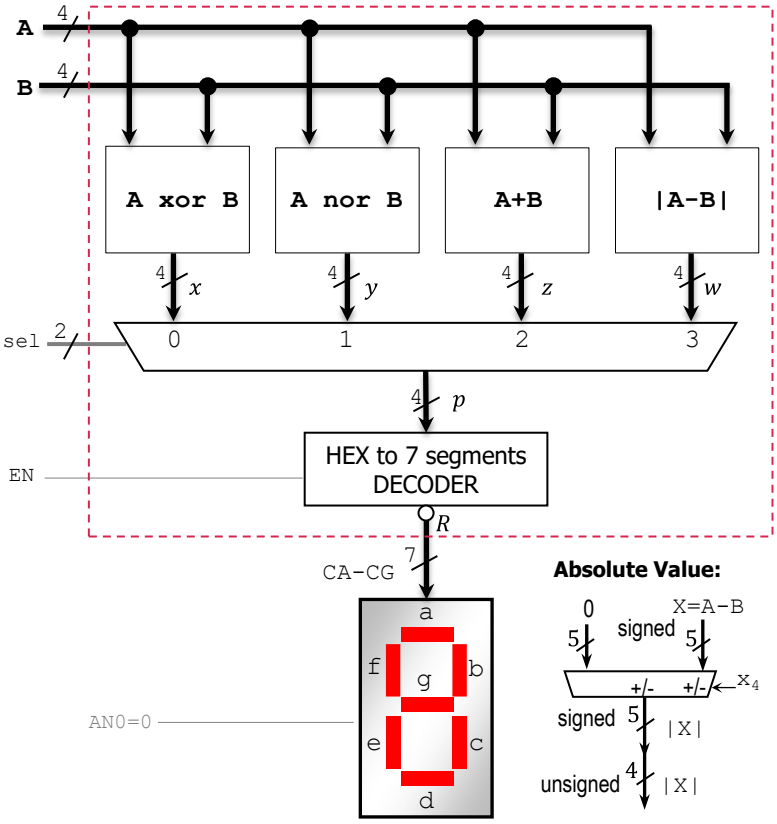
- ✓ Refer to the Tutorial: VHDL for FPGAs for a list of examples.

Activities

First Activity: Design and Simulation (70/100)

DESIGN PROBLEM

- **Simple 4-bit Arithmetic Logic Unit (ALU):** This circuit selects between arithmetic (absolute value of difference, addition) and logical (XOR, NAND) operations. Only one result (hexadecimal value) can be shown on the 7-segment display. This is selected by the input `sel(1..0)`.
- Arithmetic operations: The 4-bit inputs `A` and `B` are treated as unsigned numbers.
  - ✓ `A+B`: If there is a carry out, ignore it.
  - ✓ `|A-B|`: 4-bit result, since `|A-B|` ∈ [0,15]. Tip: zero-extend the inputs to 5 bits and implement `A-B` (5-bit signed result). Then, implement `|A-B|`, where the 5-bit signed result is always positive. Finally, use the magnitude (4 LSBs) as the unsigned output.
- Logic Operations (e.g.: `A xor B`, `A nand B`): These are bit-wise operations.
- HEX to 7 segments decoder:
  - ✓ If `EN=1` → Decoder is enabled. Result appears on the 7-segment display.
  - ✓ If `EN=0` → Decoder is disabled. All LEDs in the 7-segment display need to be off.
  - ✓ If you use a vector signal for the output (e.g. `R[6..0]`), make sure to correctly map each bit to the outputs `CA-CG`.
- In all listed boards (Nexys A7-50T/A7-100T, Basys 3, Nexys 4/DDR), each 7-segment display has active-low inputs (`CA-CG`) and an active-low anode enable `AN`. Make sure that only one 7-segment display is activated. Example:
  - ✓ Nexys A7-50T/A7-100T, Nexys 4/DDR: To use only the right-most 7-segment display, set `AN0=0`, `AN1-AN7=1`
  - ✓ Basys 3: To use only the right-most 7-segment display, set `AN0=0`, `AN1-AN3=1`.



PROCEDURE

- **Vivado: Complete the following steps:**
  - ✓ Create a new Vivado Project. Select the corresponding Artix-7 FPGA device (e.g.: the XC7A50T-1CSG324 FPGA device for the Nexys A7-50T).
  - ✓ Write the VHDL code for the given circuit. Synthesize your circuit to clear syntax errors. Pay attention to the warnings.
    - **IMPORTANT**: For `A+B` and `|A-B|` circuits, you must use full adders and logic gates (as in Lab 2).
    - To implement the Bus MUX and decoder, it is strongly advised that you use the VHDL concurrent statements.
    - Use the Structural Description: Create a separate `.vhd` file for the Arithmetic and Logic circuits, the 4-to-1 Bus MUX, the Hex to 7-segment decoder (with active low outputs), and the top file.

Instructor: Daniel Llamocca
TAs: Abd Alrahman Al Nounou, Benjamin Kidwell, Yashwanth Naidu

- ✓ Write the VHDL testbench to simulate the circuit for the following cases:

A	B	sel	EN
1001	1101	00	1
		01	
		10	
		11	
0111	1010	00	1
		01	
		10	
		11	
0111	1010	11	0

A	B	sel	EN
0101	1011	00	1
		01	
		10	
		11	
1110	0111	00	1
		01	
		10	
		11	

  - We use 4 sets of `A` and `B` values. At each set, we make `sel` vary from 0 to 3.
  - We also include one case with `EN=0` to verify that the 7-segment display is OFF (`CA-CG = 1111111`).

- ✓ Perform Functional Simulation of your design. **Demonstrate this to your TA**.
  - Add internal signals (x, y, z, w, p) to the waveform view. Go to: SCOPE window: testbench → UUT. Then go to Objects Window → Signal(s) → Add to Wave Window. Finally, re-run the simulation.

    - This step is extremely useful when debugging your circuit. Your circuit might be cleared of syntax errors, but there might still be errors that can be difficult to spot. By tracing the internal signals, we can determine where the error is located in the circuit.

    - For example: In this circuit, for a given set of input values, we can compute the expected output values and internal signal values. Then, we compare those values with those provided by the simulation:
      If the output `CA-CG` is incorrect (simulation results do not match our calculated values), we then look at the value of the signal p:
      - If the value of p is correct (i.e., simulation results match our calculated values), then the error is in the HEX to 7-segments Decoder.
      - If the value of p is incorrect, then look at the values of x, y, z, w. If all of them are correct, then the error is in the Bus MUX. If any of the values of x, y, z, w are incorrect, then we can determine which arithmetic or logic unit is generating incorrect output values.

  - For the following set of inputs, complete the expected values of the listed internal signal and outputs. Then, run the simulation and compare the values in the simulation waveform with the ones you computed. This will help you figure out where the errors (if any) are located at.

A	B	x	y	z	w	sel	EN	p	CA-CG
1001	1101					00	1		
						01			
						10			
						11			
0111	1010					00	1		
						01			
						10			
						11			

SECOND ACTIVITY: TESTING (30/100)

- **Vivado: complete the following steps:**
  - ✓ I/O Assignment: Generate the XDC file associated with your board.
    - Suggestion

Board pin names	SW15 SW14	SW13	SW7-SW4	SW3-SW0	CA-CG	AN7-AN0
Signal names in code	sel_1 sel_0	EN	A_3-A_0	B_3-B_0	CA-CG	AN_7-AN_0

    - The board pin names are used by all the listed boards (Nexys A7-50T/A7-100T, Basys 3, Nexys 4/DDR). I/Os: Note that `CA-CG` and `AN7-AN0` are active low, whereas the switches are active high.
    - **Basys 3**: There are only four 7-segment displays, hence you only have signals `AN3-AN0`.

  - ✓ Implement your design and run Timing Simulation.
  - ✓ Generate and download the bitstream on the FPGA, then perform testing (use a sample of representative cases from your testbench). **Demonstrate this to your TA**.

Instructor: Daniel Llamocca
TAs: Abd Alrahman Al Nounou, Benjamin Kidwell, Yashwanth Naidu

SUBMISSION

- Submit to Moodle (an assignment will be created):
  - ✓ This lab sheet (as a .pdf) completed signed off by the TA (or instructor)
  - ✓ (As a .zip file) all the generated files: VHDL code files, VHDL testbench, and XDC file. **DO NOT submit the whole Vivado Project**.
    - Your .zip file should only include one folder. Do not include subdirectories.
    - It is strongly recommended that all your design files, testbench, and constraints file be located in a single directory. This will allow for a smooth experience with Vivado.
    - You should only submit your source files AFTER you have demoed your work. Submission of work files without demoing will be assigned NO CREDIT.

TA signature: _____ **Date:** _____