

Laboratory 2

(Due date: **002/003**: Feb. 7th, **004**: Feb. 8th, **006**: Feb. 9th)

OBJECTIVES

- ✓ Use the Structural Description on VHDL.
- ✓ Test arithmetic circuits on an FPGA.

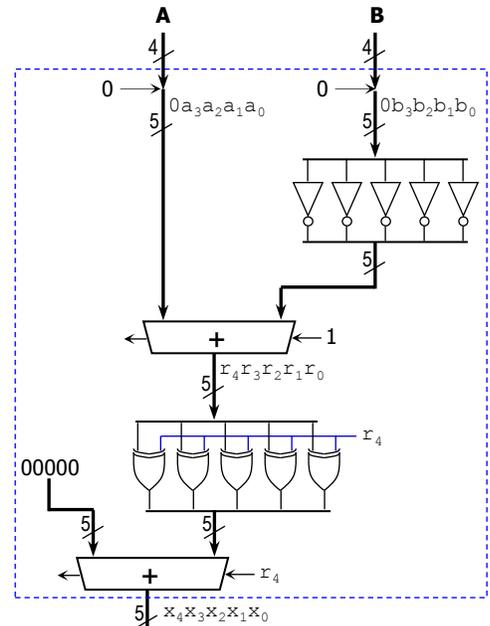
VHDL CODING

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

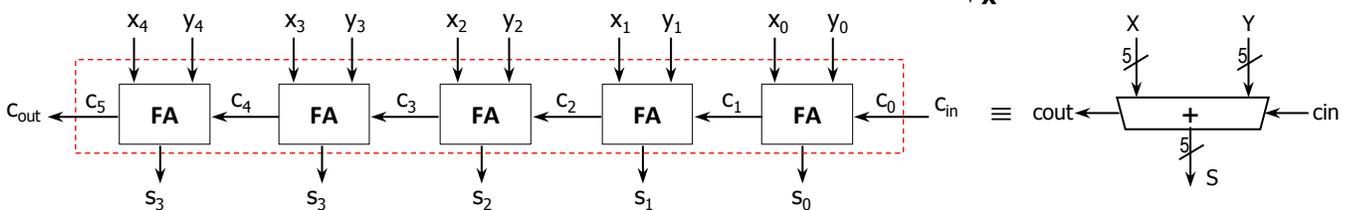
FIRST ACTIVITY (100/100)

DESIGN PROBLEM

- Absolute value of the difference. The operands (A and B) are 4-bit unsigned numbers. The operation is $X = |A - B|$.
- This circuit can be built out of two 5-bit adders, 5 NOT gates, and 5 XOR gates as depicted in the figure.
 - ✓ The output of the top 5-bit adder is $R = A - B$, where $R = r_4r_3r_2r_1r_0$.
 - For this adder: $c_{in}=1$, c_{out} is unused.
 - ✓ The output of the bottom 5-bit adder is $X = |R| = |A - B|$.
 - For this adder: $c_{in}=1$, c_{out} is unused.
 - ✓ Given the result $X = x_4x_3x_2x_1x_0$, it can be demonstrated that $x_4 = 0$. Thus, it is very common to only use $x_3x_2x_1x_0$ as the 4-bit unsigned result of $X = |A - B|$.

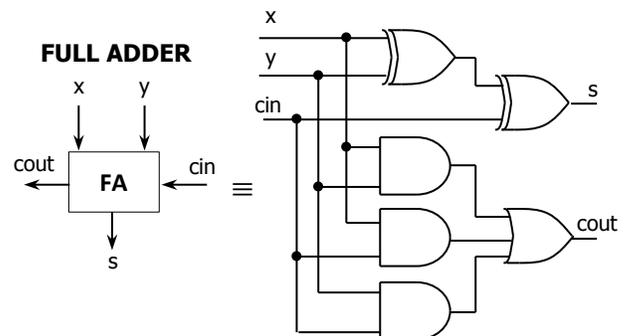


- The figure below depicts the internal architecture of the 4-bit adder and the 5-bit adder. The full adder (FA) circuit is also shown.



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 this circuit.
 - Use the **Structural Description**: Create a separate .vhd file for the Full Adder, the 5-bit adder, and the top file (absolute value of the difference of two 4-bit unsigned numbers).
 - ✓ Write the VHDL testbench to test the circuit for the following cases:
 - $A=0 \times E, B=0 \times 5 \rightarrow X=01001$
 - $A=0 \times 1, B=0 \times F \rightarrow X=01110$
 - $A=0 \times 6, B=0 \times D \rightarrow X=00111$
 - $A=0 \times B, B=0 \times 7 \rightarrow X=00100$



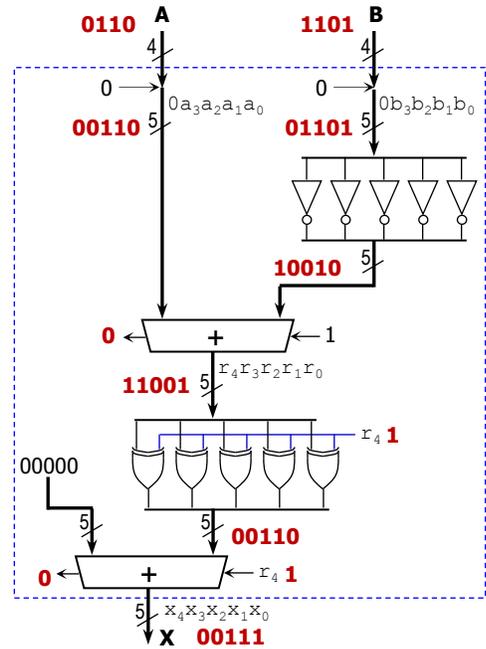
- ✓ Perform Functional Simulation of your design. **Demonstrate this to your TA.**
 - Add the internal signals (e.g.: R) to the waveform view. Go to: SCOPE window: testbench → UUT. Then go to Objects Window → Signal(s) → Add to Wave Window. Finally, re-run the simulation.
 - ✎ This step is extremely 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.

✎ In this circuit, for a given set of input values, we can manually compute the expected output values and the internal signal values. For example, the figure shows the output and internal signal values when A=0110 and B=1101. We can then compare those values with those provided by the simulation:

- If the output x is incorrect (simulation results don't match the expected values), then look at the value of the signal R.
 - If the value of R is correct (i.e., simulation results match the calculated values), then the error is located in the lower 5-bit adder and/or the XOR gates.
 - If the value of R is incorrect, then the error is on the upper 5-bit adder and/or the NOT gates.

- For the following set of inputs, complete the expected values of the listed internal signals. 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 | R | X |
|------|------|---|-------|
| 1110 | 0101 | | 01001 |
| 0001 | 1111 | | 01110 |



- ✓ I/O Assignment: Generate the XDC file associated with your board.

- Suggestion:

| Board pin names | SW7-SW4 | SW3-SW0 | LED5 | LED4 | LED3 | LED2 | LED1 | LED0 |
|----------------------|--------------------------------|--------------------------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Signal names in code | A ₃ -A ₀ | B ₃ -B ₀ | X ₅ | X ₄ | X ₃ | X ₂ | X ₁ | X ₀ |

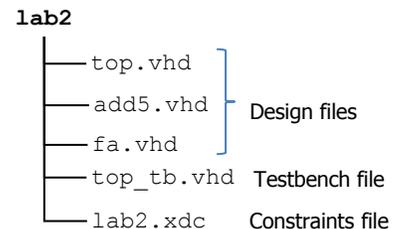
- The board pin names are used by all the listed boards (Nexys A7-50T/A7-100T, Basys 3, Nexys 4/DDR). The I/Os listed here are all active high.

- ✓ Implement your design (Run Implementation) and run Timing Simulation.

- ✓ Generate and download the bitstream on the FPGA, then perform testing. **Demonstrate this to your TA.**

- Submit (as a .zip file) the five generated files: VHDL code (3 files), VHDL testbench, and XDC file to Moodle (an assignment will be created). 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.



TA signature: _____

Date: _____