ISTANBUL TECHNICAL UNIVERSITY

Digital System Design Applications

Experiment III
VARIOUS IMPLEMENTATIONS OF BOOLE FUNCTIONS

Preliminary

Students should cover combinational circuits. Make sure that you clearly understand the meaning of Boole functions, reduction methods of functions, and combinational circuits,

Objectives

- Learn synthesizing methods of combinational circuits
- Examine how Xilinx tools make synthesizing of combinational circuits
- Usage of testbench code for making simulation

Requirements

Students are expected to be able to

- have knowledge about what combinational circuits are
- know reduction methods

References:

1. Datasheets of 74153 multiplexer, 7404 inverter, 7408 AND gate, 7432 OR gate, 7486 XOR gate
2. Spartan-3E Libraries Guide for HDL Designs
3. Constraints Guide
4. Testbench code

Implementation of Combinational Circuits with SSI and MSI Libraries

Create a new ISE project with the settings below:

- Family: **Spartan 3E**
- Device: **XC3S500E**
- Package: **FG320**
- Speed: **-4**
- Synthesis Tool: **XST (VHDL\Verilog)**
- Simulator: **ISim (VHDL\Verilog)**
- Preferred Language: **Verilog**

Add copy of SSI_Library.v and MSI_Library.v that you define in previous experiments. Add a new Verilog module named **three_different_methods.v** to your project. Clear all lines in your module. The **truth table** will be given at the beginning of experiment. Download it from the Ninova. You will implement this truth table in three different ways. Implement steps given below respectively. There are f_3, f_2, f_1, and f_0 outputs and *a, b, c,* and *d* inputs in the truth table.

Synthesize with SSI Library (120 Minutes)

1. Firstly, it is necessary to obtain outputs in terms of inputs of the truth table. Primarily, make reduction by using **Karnaugh Maps** and obtain outputs as **two-level**. Add all steps of reduction to the your report.
2. Try to simplify your results as **multi-level** (Hint: try to use XOR). Add your multi-level results and schematic of your circuit to the report.
3. Create a module **with_SSI.v** into **three_different_methods.v**.
4. The module has four 1-bit inputs as *a, b, c, d* and four 1-bit outputs as *f3, f2, f1, f0.*
5. Assign **with_SSI.v** as a top module. Implement your results (multilevel results) in this module by using elements of SSI library.
6. Add a testbench which name is **experiment_3_test.v** (Download it from the Ninova).
7. Change your view to **Simulation** tab and make behavioral simulation.
8. You should see the generated outputs that are relevant to your inputs at the bottom of the screen. Show that these outputs are same with the truth table.
9. Examine the waveform. You should see that new values are assigned to inputs at t = 550ns.
10. When new outputs can be seen after we apply new values to the inputs at t = 550ns? Are new outputs seen at t = 550ns or another time after 550ns? Give your answer with the waveform and explain and show it clearly in your report.
11. Switch back your view *Implementation.* **Synthesize** your circuit, obtain **RTL** and **Technology** schematics and add them to the your report.

12. Create new constraints file as **experiment3.ucf**. In this ucf, connect your a, b, c, d inputs to SW3, SW2, SW1, SW0 respectively. Also, connect outputs f3, f2, f1, f0 to LED3, LED2, LED1, LED0 respectively.

13. Run **FPGA Editor** after Place&Route and add the layout to the your report.

14. Determine **Pad to Pad** delays and add it to the your report.

15. Write **TIMESPEC** constraint to the ucf. This constraints defines pad to pad delays as maximum 7ns. Run again **FPGA Editor** and show new layout in your report. Also, run **Timing Analyzer** and find new pad to pad static timing information. Add it to the your report. What is the difference between before and after TIMESPEC constraint? Clarify it in your report.

16. Remove the time constraint and use **LOC** constraint in **with_SSI.v** on the purpose of placing the gate to **SLICE_X1Y1**. Make sure that you should write the LOC constraint into the verilog module. Implement your circuit again and show the new layout and timing information in your report. Explain the difference between before and after LOC constraint.

17. Now, use TIMESPEC and LOC constraint at the same time. Add the followings, layout view and delay information to your report:
    - After the implementation of new circuit with constraints, are pad to pad delays below 7ns?
    - What is the amount of decrease in the biggest delay?
    - Is there any difference between two layouts? (First layout is with only LOC constraint, second layout is with both LOC and TIMESPEC constraint)
    - Is there a relationship between the decrease of delay time and routing?

18. Generate **post place&route simulation model**. Switch the view to simulation and choose **post-route**. Run **experiment_3_test module**. Inputs are updated at t = 550ns again. When do outputs change after changing of inputs at t = 550ns? Add reason of this delay and its relation between constraints to your report. Also, explain that why it is not seen in behavioral simulation.

19. Generate **BIT file** and program your FPGA. Show that your circuit working correctly when compared to the truth table.

Synthesizing with Decoder (60 Minutes)

1. Write down **with_decoder.v** into your **three_different_methods.v**.

2. The new module have four 1-bit inputs as a, b, c, d and four 1-bit outputs f3, f2, f1, f0. Assign with_decoder.v as top module.

3. Show that on a paper 4-to-16 decoder's output satisfy **minterms**. Add it to the your report.

4. Find **sum of products** terms by looking outputs in the truth table given before. Add it to the your report.

5. Write your code with structural method. For sum of product terms, use decoder in MSI Library. For addition, use OR gates in your SSI Library. In your design, define minterms with 1**6-bits wire m**.

6. Replace uut (unit under test) in **experiment_3_test.v** with **with_decoder** and make behavioral simulation. If it is not working correct, change your design. Add your waveform to the report.

7. Synthesize the module and add RTL and technology schematics to your report.

8. Synthesize your circuit with 7ns constraint in ucf.

9. Add post place&route static timing information to the your report.

10. Run FPGA Editor and add layout of your circuit to the your report.

11. Find that how many LUTs there are in your circuit in post place&route. Make comparison of LUT number with the circuit that you use only SSI Library. Comment reason of difference and give result. Add them to the your report.

12. Generate BIT file and program your FPGA. Show that your circuit working correctly.

13. Then, cancel optimizations that Xilinx ISE tool apply and examine results again. For this purpose, use **OPTIMIZE** constraint. Write this constraint just before your DECODER instantiate.

14. Place&Route your circuit and show how many LUTs are used in your report.

Synthesizing with MUX (60 Minutes)

1. Write down **with_MUX.v** into your **three_different_methods.v**

2. The new module have four 1-bit inputs as a, b, c, d and four 1-bit outputs as f3, f2, f1, f0. Assign with_MUX.v as top module.

3. You will use four multiplexers for generating f3, f2, f1, f0 outputs. For this purpose, connect **a, c** to select inputs of multiplexers (a is the most significant bit).

4. Draw the schematic on the paper by showing all ports and truth table. Add them to the report.

5. Write your structural HDL code by using SSI and MSI libraries.

6. Replace uut in **experiment_3_test.**v with **with_MUX** and make behavioral simulation. If it is not working correct, change your design. Add waveform to the report.

7. Synthesize your module, add RTL and Technology schematics to the report.

8. Synthesize your module with 7ns time constraint in the ucf. Add post place&route static timing information to the report.

9. Add following to the report:
   - Layout of your circuit (From the FPGA Editor)
   - How many LUTs are used in the circuit after generating layout? (After implementing P&R)
   - Compare LUT numbers in with_SSI and with_decoder. Make comments about similarities or differences.

10. Generate BIT file and program FPGA.

Experiment Report

Each group member is going to prepare his/her own report. Reports should include:

- Screenshots and results asked above
- Your comments on results
- Which arithmetic operation is implemented in the Truth Table that you have used in the experiment? (Hint: Think that you have two numbers as inputs and there are one 4-bits output)
- What is the difference between behavioral simulation and post place&route simulation? When should we use behavioral simulation and post place&route simulation?
- Make comparison about three methods that you made in the experiment in terms of:
  - Difficulty of design
  - Difficulty of coding
  - LUT usage
  - Delays
- Implement **with_mux** circuit again with discrete components. Use 74153 MUX, 7404 Inverter, 7408 AND Gate, 7432 OR Gate and 7486 XOR Gate. Draw the schematic of circuit. Do not forget to make supply and ground connections. YOU MUST HAVE THIS SCHEMATIC FOR THE NEXT EXPERIMENT. DO NOT FORGET BRING IT NEXT WEEK.

Reports are to be submitted before next experiment. Submission includes reports and project files.