ECE385

*SPRING 2020*

*Experiment #3*

A Logic Processor

Jieting Chen & Xinhao Tong

LA-3

Shihao Su

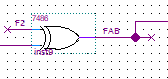
1. **Introduction:**

a.

This experiment is to design a logic processor, which has eight operations: AND, OR, XOR, 1111, NAND, NOR, NXOR, 0000. It can operate on two 4-digit binary numbers. What’s more, the routing unit of it can decide where to store the results as well.

b.

A. The XOR gate (e.g. 7486):



In this figure, when F2 is high, FAB is the inverted value of the other input. Otherwise, FAB will has same value with the other input.

B. By using the chip above, we can reduce the number of gates from 4 NAND to represent a XOR function to 1 XOR gate.

It will be more convenient and intuitive for us to analyze the circuits. The modular design is more readable for human. So, it will be easier for us to debug with modular design as well. What’s more, when the same partial design will be used more than once in the circuits, it will save time with modular design.

1. **Operations of the operation processor:**
2. **Describe the sequence of switches the user must flip to load data into the A and B registers.**

When need to load data into A\B register:

1. Give data to D0~D3 pins
2. Signal LOAD\_A/LOAD\_B should be high
3. Signal EXECUTE should be low

Note: step 1 should be done before step 2 and 3.

1. **Describe the sequence of switches the user must flip to initiate a computation and routing operation.**

After the above operations in part a, next will be the computation operation and routing operation in

**3. Written description, block diagram and state machine diagram of logic processor:**

**4. Control Unit**

i.

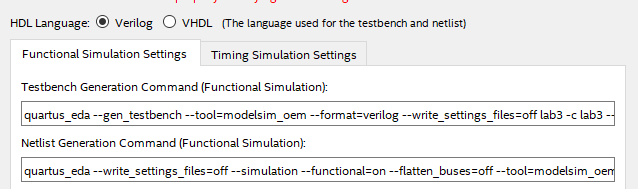
ii.

iii.

**5. Description of all bugs encountered, and corrective measures taken**

Bug 1: When build the circuits for control logic unit, we misuse the AND gate to present Q = E + C0 + C1 rather than OR gate.

Bug 2: At first, I cannot do the functional simulation on my computer although I change the path correctly for the .vwf file. Finally, we found that we should change both the two paths at top (shown in following picture). We fixed the bug by this way successfully.



**6. Conclusion:**

1. In this experiment, we designed a logic processor, which can perform eight different logical computation on two 4-digit binary numbers successfully. The eight kinds computation are: AND, OR, XOR, 1111, NAND, NOR, NXOR, 0000. The routing unit of it decide how to save back the values to the registers. There are four different ways of storing the data back. All in all, we have learned the way of designing a simple processor, and our experiment is totally successful. Attached is our final waveform.
2. Q1 **Document changes to your design and correct your Pre-Lab write-up, explaining any difficulties you had in debugging your circuit. Outline how the modular approach proposed in the pre-lab help you isolate design and wiring faults, be specific and give examples from your actual lab experience.**

We have made one change on the register unit connection. Firstly, we decide to combine the SHIFT signal with A\*/B\* in some ways as the input for SRSI input-pin on 74194 chips. Then we found that was a wrong design, so we change to use (SHIFT XOR CLK) as the input for CLK pin.

Modular approach: We divide the circuits into four parts: Control unit, register unit, computing unit and routing unit. We test each of them separately and combined them at last.

Q2 **Discuss the design process of your state machine, what are the tradeoffs of a Mealy machine vs a Moore machine?**

We use a Mealy machine in this experiment. A mealy machine will be easier for us to design and build the circuits; however, a Moore machine can be with less/no glitches, it will be more stable. The disadvantage of Moore machine is difficult to build while a mealy machine may leads some glitches and unstablity.