ECE 385

Fall 2021

Experiment # 3

A Logic Processor

Xu Ke/ Zhu Xiaohan

LA4/Thursday & 18:00-20:50 Huang Tianhao

1) Introduction

a. Summary

Our circuit is designed a 4-bit serial logic processor, which can perform 8 kinds of operation on 4-bit words serially, including AND, OR, XOR, 1111, NAND, NOR, XNOR, and 0000. This circuit can also decide which routine the output goes, the router output of A' B' can be AB, AF, FB, BA.

b. Answers to pre-lab questions

A. Describe the simplest (two-input one-output) circuit that can optionally invert a signal (i.e., one input determines if the output is equal to the other input or equal to the other input inverted). Sketch your circuit.

We can use a XOR gate to solve this question. By the truth table of XOR gate, we find that the output is equal to the input if the selection bit is 0, while the output is equal to the input inverted if selection bit is 1.

Selection Bit	Input Signal	Output
0	0	0
0	1	1
1	0	1
1	1	0

Table 1.1 XOR Gate Truth Table

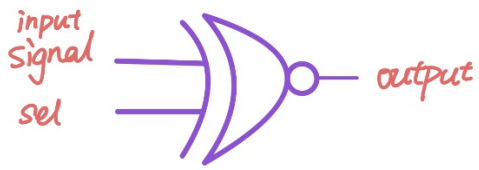

Figure 1.1 XOR Gate

B. Explain how a modular design such as that presented above improves testability and cuts down development time.

Modular design makes it possible to test the circuit in part, so that it improves the testability. Also, when developing each part, we only need to consider using the input signals without consider how to produce them, which help save the development time.

C. Design, document and build the circuit described in Part II. Your circuit should be able to perform correctly all the functions listed. You may use either 7495A or 74LS194A chips for your shift registers. You will want to study each of the chips carefully before deciding on one or the other. Be sure to make your design as efficient as possible (there is more than one way to design this circuit).

We use 74LS194A chips. Our circuit will be include in this report.

2) Operation of the logic processor

a. Describe the sequence of switches the user must flip to load data into the A and B registers.

Load data into A: We need to flip D0-D3 to set data, then we need to flip EXECUTE to 1 and flip LOAD_A to 1.
Load data into B: We need to flip D0-D3 to set data, then we need to flip EXECUTE to 1 and flip LOAD_B to 1.

b. Describe the sequence of switches the user must flip to initiate a computation and routing operation.
We first choose the operation and set the corresponding F0-F2, then flip R0-R1 to the route we need, finally, we flip EXECUTE to 1 to start a computation and routing operation.

3) Written description, block diagram and state machine diagram of logic processor

a.

Register Unit

Since our logic processor need two operators, the register unit need two shift registers (74194), which can parallel load of D3-D0 to A3-A0 (Register A) and B3-B0 (Register B) according to the signal from control unit. The bits hold in registers will shift right and be executed serially in computation unit. The content of these registers should be display before and after the operation.

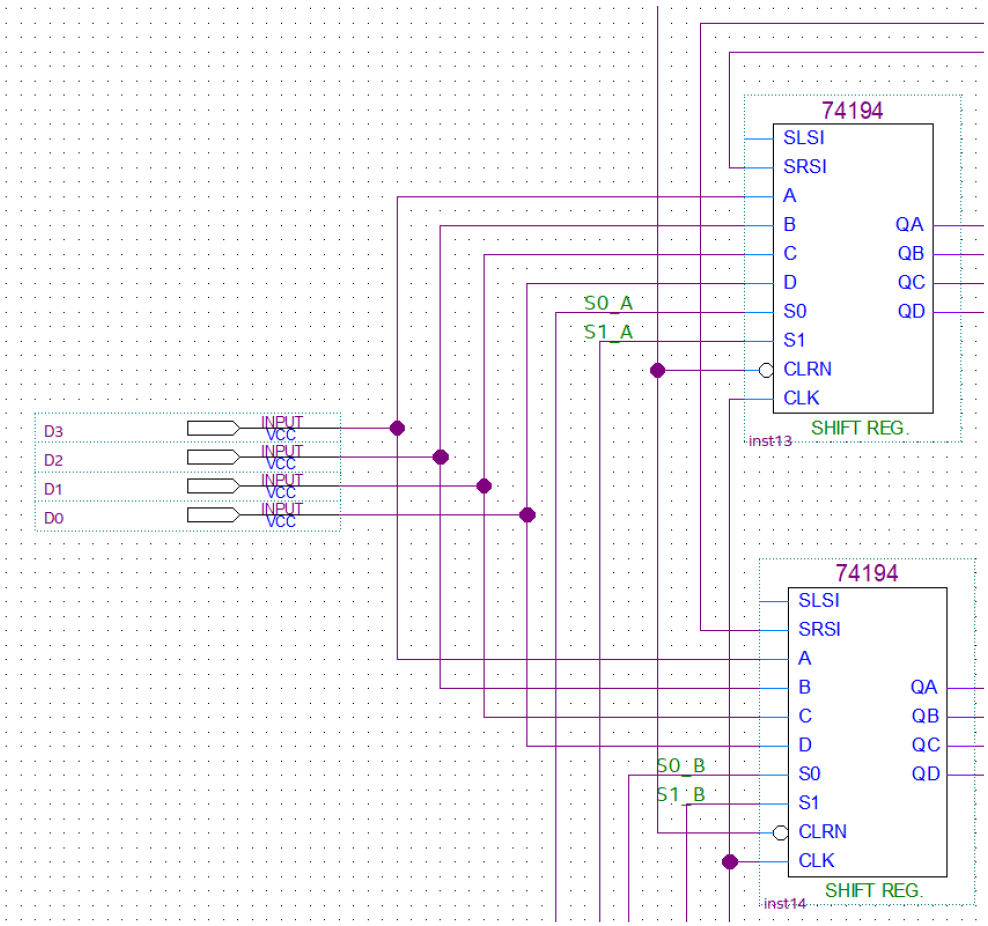

Figure 3.1 Implement of Register Unit

Computation Unit

The computation unit takes A0 and B0 to do one of eight operations according to function selection signal F2-F0,

Function Selection Inputs			Computation Unit Output
F2	**F1**	**F0**	**f(A, B)**
0	0	0	A AND B
0	0	1	A OR B
0	1	0	A XOR B
0	1	1	1111
1	0	0	A NAND B
1	0	1	A NOR B
1	1	0	A XNOR B
1	1	1	0000

Table 3.1 Truth table of F (A, B)

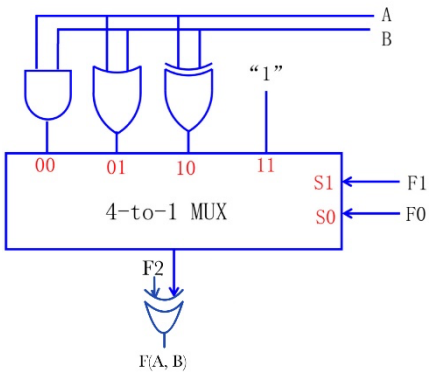

Figure 3.2 Circuit of F (A, B)

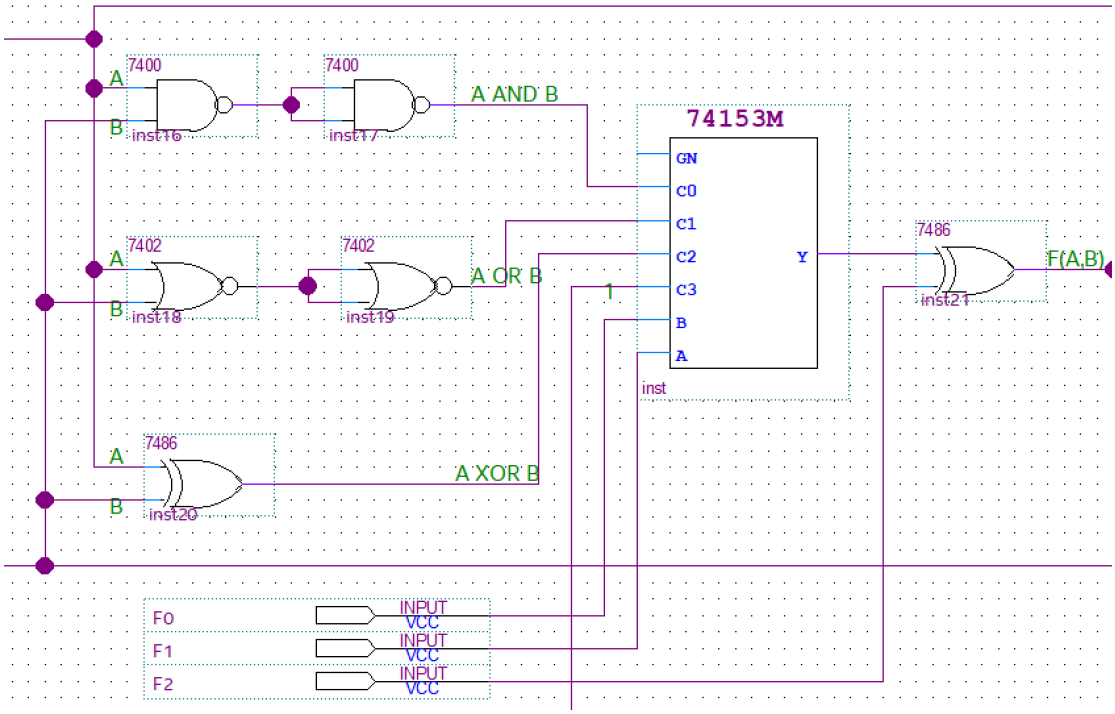

Figure 3.3 Implement of Computation Unit

Routing Unit

The routing unit will accept A, B and F(A, B), then according to routing selection signal R1-R1 to determine where those signals goes to.

Routing Selection		Router Output	
R1	**R0**	**A***	**B***
0	0	A	B
0	1	A	F
1	0	F	B
1	1	B	A

Table 3.2 Routing Unit

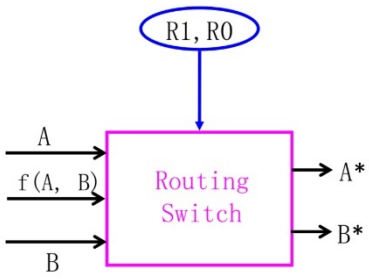

Figure 3.4 Circuit of Routing Unit

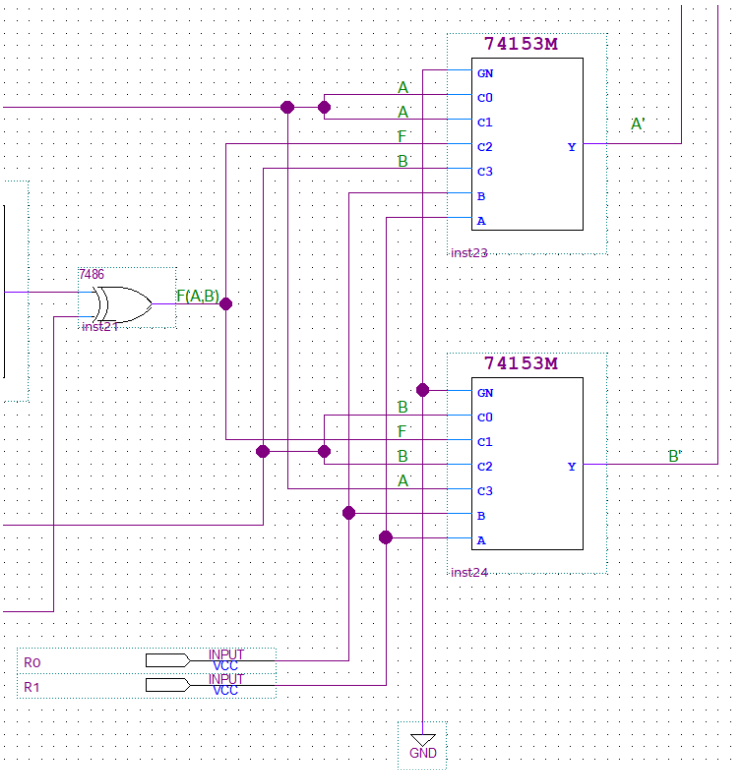

Figure 3.5 Implement of Routing Unit

Control Unit

The control unit manage the selection bit of this logic processor. It has 4 input: EXECUTE, LOAD_A, LOAD_B, CLK. The signals LOAD_A and LOAD_B will be process to tell whether we can load D3-D0 to shift registers A and B respectively, or we need to shift or hold. EXECUTE tells when the circuit start computation and when finish. When EXECUTE is high, the control unit will control the shift unit to load and shift, until the EXECUTE signal is low. The CLK signal will put into counter and it will start counting when EXECUTE is high and counter count to 3 (C1C0=11).

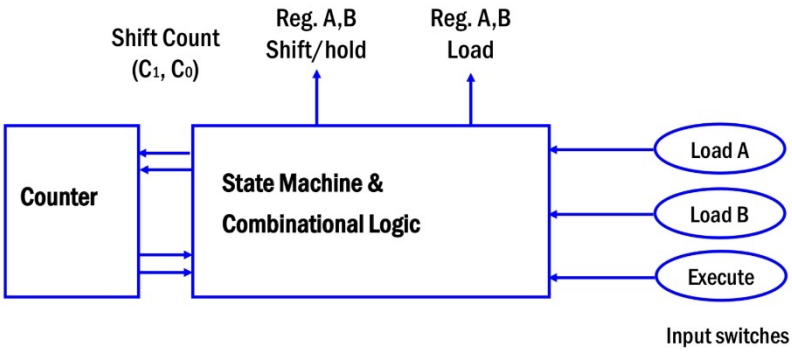

Figure 3.6 Circuit of Control unit

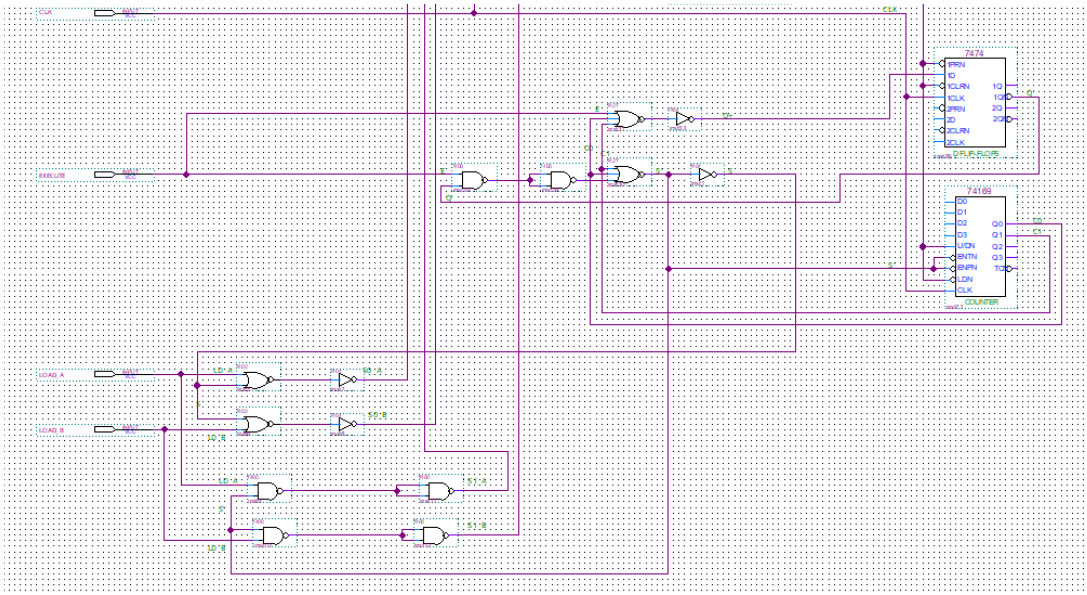

Figure 3.7 Implement of Control Unit

b. high-level block diagram

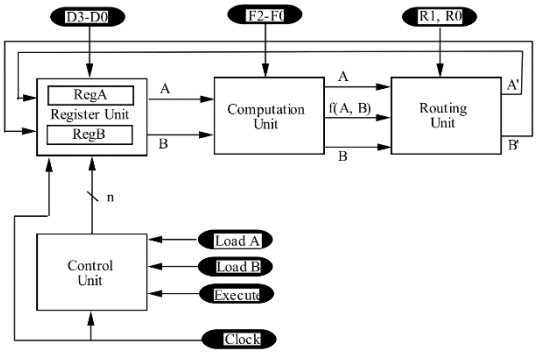

Figure 3.8 high-level block diagram

c. State Machine Diagram

We are using Mealy machine. The state diagrams are shown below:

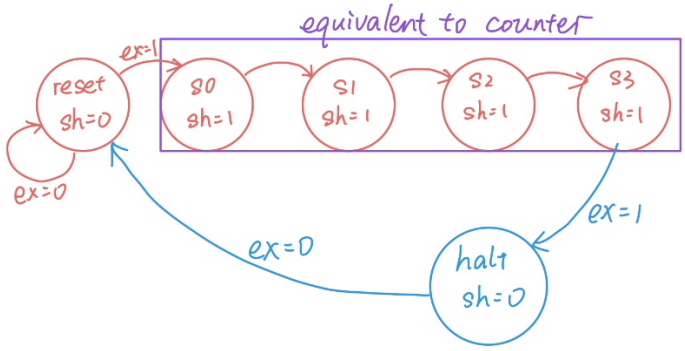

Figure 3. Rough state diagram

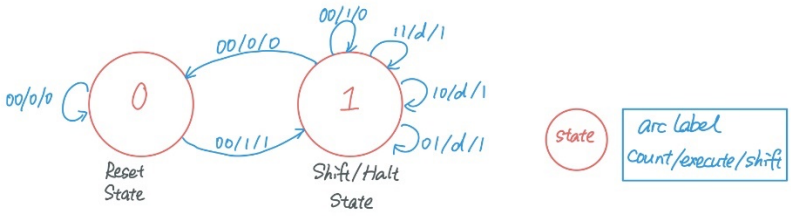

Figure 3. Mealy state diagram

4) Design steps taken and detailed circuit schematic diagram

a. Procedure of the design steps

For this part, we also consider using Moore machine, however, there will be more states in this implement, we need two bits to represent the state while the Mealy machine only need one bit. Thus, we use the Mealy machine to implement our control unit.

Truth table

E	Q	C1	C0	S	Q+	C1+	C0+
0	0	0	0	0	0	0	0
0	0	0	1	d	d	d	d
0	0	1	0	d	d	d	d
0	0	1	1	d	d	d	d
0	1	0	0	0	0	0	0
0	1	0	1	1	1	1	0
0	1	1	0	1	1	1	1
0	1	1	1	1	1	0	0
1	0	0	0	1	1	0	1
1	0	0	1	d	d	d	d
1	0	1	0	d	d	d	d
1	0	1	1	d	d	d	d
1	1	0	0	0	1	0	0

1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	1	1	0	0

K-maps

S (Register Shift)		C1C0			
		00	01	11	10
EQ	00	0	X	X	X
	01	0	1	1	1
	11	0	1	1	1
	10	1	X	X	X

S = EQ' + C0 + C1

Q+ (Next State of Q)		C1C0			
		00	01	11	10
EQ	00	0	X	X	X
	01	0	1	1	1
	11	1	1	1	1
	10	1	X	X	X

Q+ = E + C0 + C1

For this part, the differences appear only in the shifters' select bits (S1S0) chosen. We know that when S = 1, we do the shifting. When S = 0, LD = 1, we load new values and LD = 0, we do nothing. There are many kinds of options and we simply choose this state representation:
S1S0 = 00 => Halt
S1S0 = 01 => Shift right
S1S0 = 11 => Load values
Notice that the selection is same for SHIFTER A and B, so the K-maps should be same.

Truth Table

S	LD_A/B	S1_A/B	S0_A/B	Operation
0	0	0	0	Halt
0	1	1	1	Load values
1	0	0	1	Shift right
1	1	0	1	

K-maps

S1_A/B		LD_A/B	
		0	1
S	0	0	1
	1	0	0

S1_A = S'LD_A
S1_B = S'LD_B

S0_A/B		LD_A/B	
		0	1
S	0	0	1
	1	1	1

S0_A = S + LD_A
S0_B = S + LD_B

b. Detailed Circuit Schematic

Control Unit

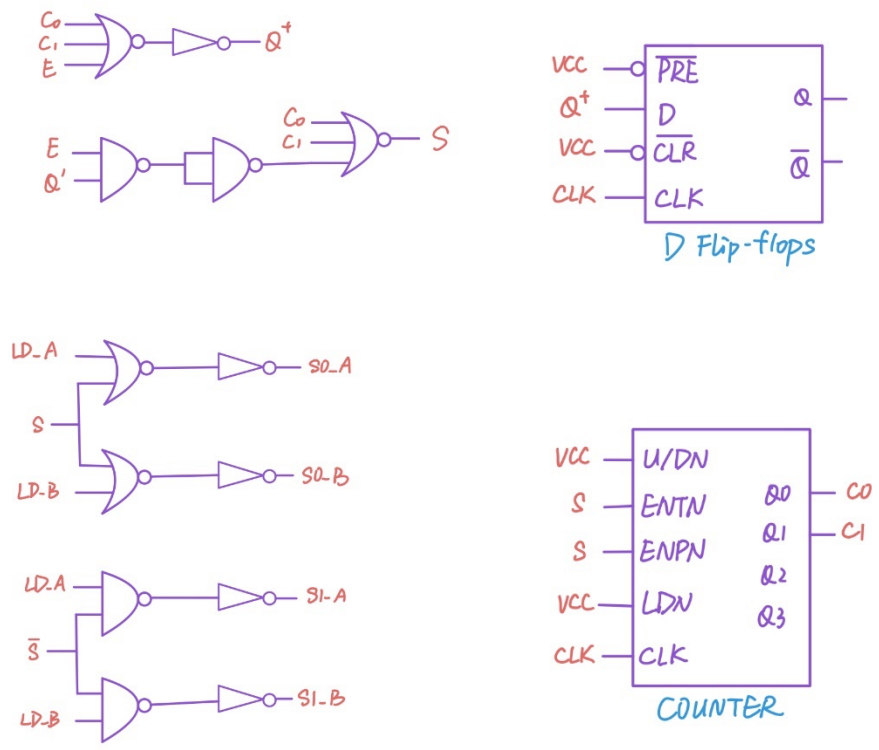

Figure 4.1 Control Unit in gate level

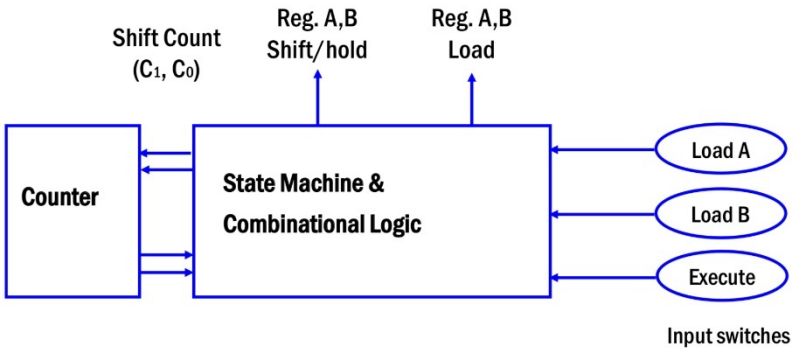

Figure 4.2 Control Unit

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

The circuit went well this time, we didn't encounter too many bugs, just forgot to connect some wires at first time, but fix it very quickly.

6) **Conclusion**

   a. **Summary**

   In this lab, we set up a logic processor. We learned the differences between Moore and Mealy machine, and how to use modular approach to build up our circuit and debug.

   b. **Answer to all post-lab questions**

   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.

   Modular approach helps us divided our circuit in 4 parts, so that we can test and debug each part individually. It makes debugging much simpler. For example, we first build up the control unit and make sure it works, when we found the computation output is incorrect, we just looked at the computation unit to find the error, which save time.

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

   The only difference between a Mealy machine and a Moore machine is that, for Moore machine, outputs are determined by only current state, however, for Mealy machine, outputs depend on both state and inputs. Mealy machine is more efficient for out design, since we have fewer states than using Moore machine. The logic implement will be much simpler. Also, the output of Mealy machine change at the rising edge of the clock while output of Moore machine only changes when logic process is done. So, we can have synchronously design by using Mealy machine.