# ECE 385

Spring 2024

Experiment # 3

# Lab3: A Logic Processor

Name:Jie Wang, Shitian Yang

Student ID: 3200112404, 3200112415

Prof. Chushan Li, Prof. Zuofu Cheng

ZJU-UIUC Institute

Jan. 26, 2024, Friday D-225

TA: Jiebang Xia

Demo: 5/5

## Introduction

### Purpose of Circuit:

**In this lab, we are required to design and build a bit-serial logic operation processor using 4-bit shift registers, multiplexers, a counter, and a finite state machine (FSM) for the control unit. Simulated on Quartus 2 Environment, the processor will perform eight different logic functions and route the results in four different ways. The design includes a register unit with two registers (74194), a computation unit for logical operations, a routing unit for signal direction, and a control unit realized by an FSM to manage the register inputs.**

**图示

描述已自动生成**

**Fig-1：Logic Block Diagram of the circuit**

**The expected instruction can refer the following data sheet, with:**

1. **4 bit data input D[3:0] and Load A&B signal, loading into register A, B separately.**
2. **Control signal F[2:0] to select the logic operation function.**
3. **Routing signal R[1:0] to select the router output register.**
4. **Execute signal EXE[0] to turn on the shifting based processer.**

**图示, 表格

描述已自动生成**

**#### Step 1: Understand the Components**

**- \*\*Register Unit\*\*: Consists of two 4-bit shift registers to hold RegA and RegB.**

**- \*\*Computation Unit\*\*: Takes inputs from RegA and RegB and computes logical operations based on function selection inputs.**

**- \*\*Routing Unit\*\*: Determines which signals will be routed to the outputs A' and B' based on routing selection inputs.**

**- \*\*Control Unit\*\*: Manages the inputs to the register unit, including Load A, Load B, Execute, and Clock.**

**#### Step 2: Design the Computation Unit**

**1. Identify the logic functions (AND, OR, etc.) wer circuit needs to perform.**

**2. Design logic circuits for each of the eight functions using basic gates.**

**3. Make sure the computation unit can output both the result of the function and the unchanged inputs A and B.**

**#### Step 3: Design the Routing Unit**

**1. Understand the routing table which defines how A, B, and f(A, B) are routed based on R1, R0 inputs.**

**2. Design a routing scheme using multiplexers that selects the correct input for A' and B' outputs.**

**#### Step 4: Design the Control Unit using FSM**

**1. Determine whether we'll design a Moore or Mealy machine; here, a Mealy machine is recommended for fewer states.**

**2. Understand the state diagram and create a state transition table based on the operation cycle.**

**3. Fill out the state transition table and highlight any 'don't care' conditions.**

**4. Create K-maps for each output (Reg. Shift, Q+, C1+, C0+) and determine the required logic for transitions.**

**#### Step 5: Implement the Design on Hardware**

**1. Choose the appropriate shift register chips (7495A or 74LS194A) and understand their functionalities.**

**2. Connect the circuit according to wer designs for the computation, routing, and control units.**

**3. Use switches for inputs like Load A, Load B, Execute, and function/routing selectors.**

**4. Set up the clock using a square wave from a pulse generator.**

**5. Display the contents of RegA and RegB on LEDs.**

**#### Step 6: Debugging and Testing**

**1. Load values into RegA and RegB and perform logical operations to see if the correct results are displayed.**

**2. Test all eight functions and four routing options.**

**3. Ensure the computation cycle completes correctly, even if the EXECUTE switch is turned off mid-cycle.**

**#### Step 7: Documentation and Post-Lab**

**1. Document any changes we made to wer initial design based on the debugging process.**

**2. Explain the modular design benefits and how it helped during the testing and debugging phases.**

**3. Discuss the trade-offs between a Mealy and Moore state machine based on wer design process.**

## Prelab Question

1. **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 XOR. The XOR gate acts on the input signal and a control bit. If the control bit is 0, the output is the same as the input signal; if the control bit is 1, the output is the inverted signal.

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

It can improve testability and reduce development time by allowing designers to focus on individual components rather than the entire system. Each module can be tested independently, making it easier to locate and fix errors. Moreover, modular designs enable parallel development of components, speeding up the overall design process.

1. **Design, document and build the circuit described in Part II. Your circuit should be able to perform correctly all the functions listed.**

We use two 74194s to build the shift memory, A~D ports to load data, S0~S1 to control the shift, hold or read status. And the we use a 74157 and a 74153M in series as 8-1 MUX and load F2~F0 to choose which one to output. Then use a 74153 (8-2 MUX) to decide four 2-membered pairs by using R0~R1. For control unit, we decide to use a counter and a flip-flop to simulate the state machine. The control unit will read EXECUTE and its outputs to send signal to the S0 port for those two shift memory registers.

## Bug Log

* **Description of all bugs encountered, and corrective measures taken:**

Some functions are swapped when we read the output from wmf.

The order of F2-F0 is wrong. We adjust the F2-F0 to the correct port of chip, it works.

1. The Q state cannot be hold, E high and Q high, E low and Q low.

We need to lock the state. So we add two flip-flops to lock the Q state, ensure that E’s flip-flop will be locked when Q is high.

1. The len of shift signal is not correct, 3 clk long instead of 4 clk long.

The design of trigger condition is wrong. Because flip-flop and counter has some signal delay. So we redesign the judged condition, we get 4 clk long shift time signal.

1. The counter cannot work, QB port do not output when use 7493 chip.

We read the description of 7493 chip again and find we need to connect QB to clkB, then it can count more than 2.

## Postlab Question