# ECE 385

Spring 2024

Experiment # 6

# Lab6 : Simple Computer SLC-3.2

# in System Verilog

Name:Jie Wang, Shitian Yang

Student ID: 3200112404, 3200112415

Prof. Chushan Li, Prof. Zuofu Cheng

ZJU-UIUC Institute

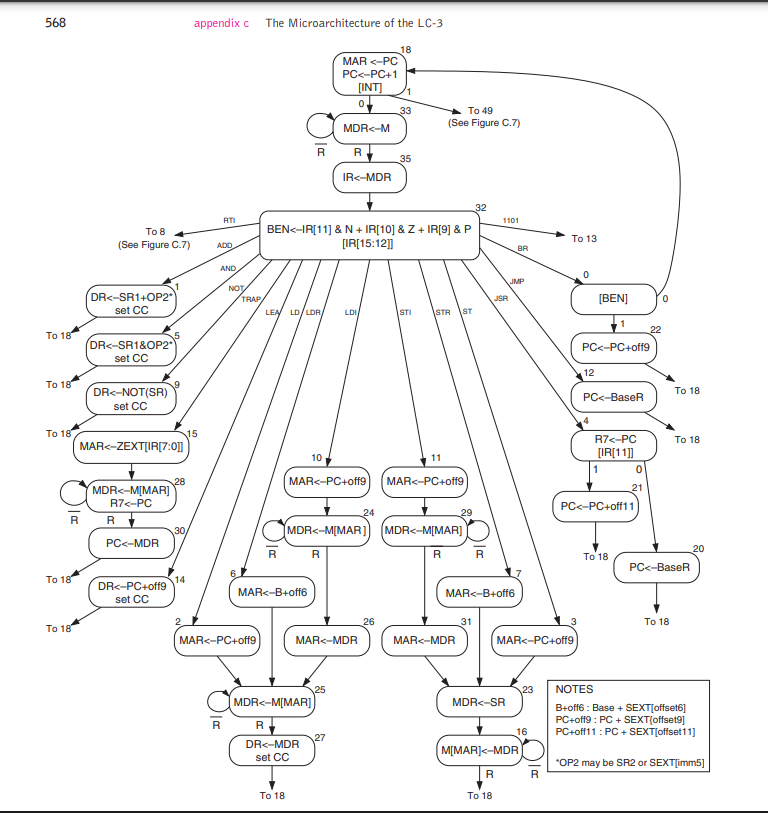
March 22, 2024, Friday D-225

TA: Jiebang Xia

**Demo Point: 5/5**

## 1.Introduction

Let time goes back to the golden time of ECE120/220, we meet the ***Little Computer 3 Assembly(LC-3)*** once more in the ECE385. This lab aimed to design a simple microprocessor using SystemVerilog, implementing a subset of the LC-3 Instruction Set Architecture (ISA). The processor is 16-bit, with a 16-bit Program Counter (PC), 16-bit instructions, and 16-bit registers. The design process involved understanding the central processing unit (CPU), memory storage, and input/output interface.



**Fig-1:**Holy State Diagram of the Whole LC-3

Luckily, we just need to write the simplified version of the LC-3. Operating on a Fetch-Decode-Execute cycle, it includes components such as the PC, Instruction Register (IR), Memory Address Register (MAR), Memory Data Register (MDR), Instruction Sequencer/Decoder, status register (nzp), a general-purpose register file, and an Arithmetic Logic Unit (ALU).

## 2.Prelab Question

### 1

Most of our time are spent on the document reading and environment configuration. We finished the adder code modification very soon, but it is time-consuming to understand the path setting and usage of testbench. Thankfully, we finished the lab in time with the assistance of our fellow classmates and [the detailed blog written by a formal ECE385 TA.](https://kttechnology.wordpress.com/)

Detail information please read Operation of the Adder Circuit.

### 3. Performance analysis(Design Analysis Comparison Results)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Carry-Ripple | Carry-Lookahead | Carry-Select |
| Memory (BRAM) | **1** | **1** | **1** |
| Frequency | **1** | **1.284** | **1.1** |
| Total Power | **1** | **1.02** | **1** |

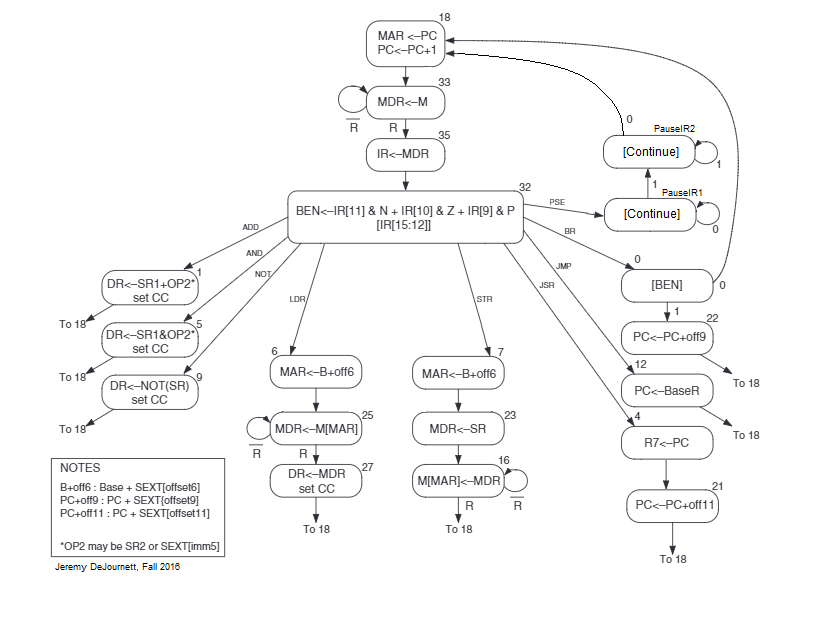
**Table-1 : Memory, Frequency and Power Comparison**

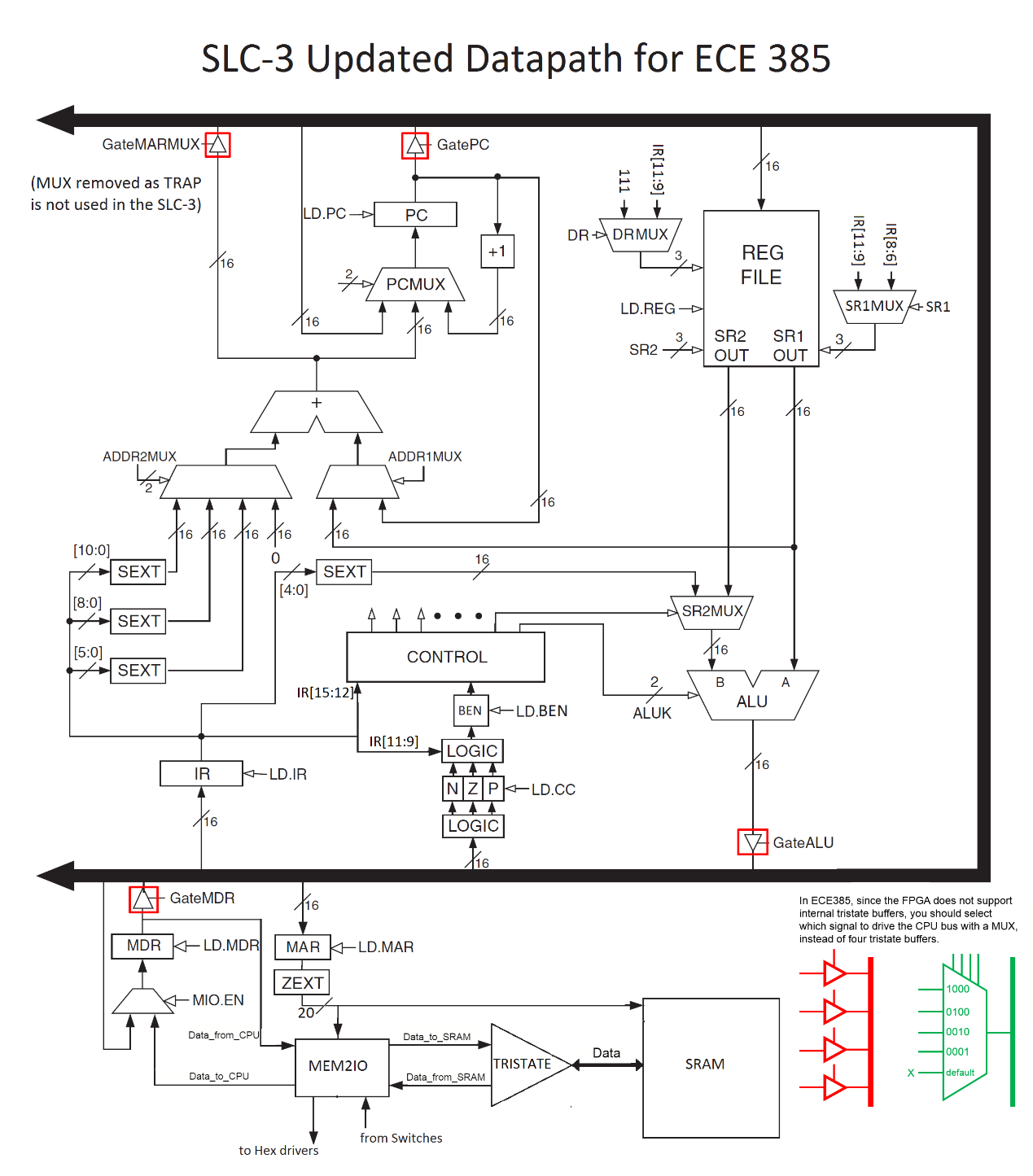
**Table-2 : Memory, Frequency and Power Comparison**

## 3. SLC-3 Design

The SLC-3 processor's design includes several key modules:

* **CPU Core**: Contains the PC, IR, MAR, MDR, ALU, and register file.
* **Instruction Sequencer/Decoder (ISDU)**: Generates control signals for the processor's operations.
* **Memory Interface**: Facilitates communication between the CPU and memory.
* **I/O Interface (Mem2IO)**: Manages I/O with physical devices like switches and displays.





We carefully modified the SystemVerilog code for key modules including ***`Control.sv`***, ***`Register\_unit.sv `,`Reg\_4.sv` and `Processor.sv `***. We then integrated an existing ***`testbench\_8.sv`*** file, ensuring it was fully compatible with our enhanced 8-bit processor. Our modifications were compiled and subjected to simulation, which we completed successfully with no errors, confirming the functional correctness of our updates.

**Fig 2:** FSM Viewer Output, including 4 more state G, H, I, J

**Fig-3: 0-1000ns Waveform, ErrorCnt == 0**

**Fig-4: Passed all the testcase in testbench\_week1.sv**

## 4.Operation of the Adder Circuit

### Overview:

As demonstrated in Fig-3, The data flow in our adders can be considered as below:

**Fig-5:** RTL Viewer of the Whole Circuit

### Ripple adder:

## 5.Post-lab Questions

### 1. Design Resources and Statistics table

|  |  |
| --- | --- |
| **LUT** |  |
| **DSP** |  |
| **Memory (BRAM)** |  |
| **Flip-Flop** |  |
| **Frequency** |  |
| **Static Power** |  |
| **Dynamic Power** |  |
| **Total Power** |  |

### 2. What is the function of the MEM2IO.sv module?

### 3. What is the difference between the BR and JMP instructions?

## 6. Bug Log

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

### 1. We fail to load the MDR into IR

2. We don’t know how to load data into

## 7.Conclusion

Through lab5, we practiced on the overall deployment of System Verilog together with the

performance analysis as below:

# 8. References

[1] KTTECH. (2017, January 31). ECE 385 Lab 5: An 8-bit Multiplier in SV. Retrieved from <https://kttechnology.wordpress.com/2017/02/10/ece-385lab5-an-8-bit-multiplier-in-sv/> Teaching Assistant Blog

[2] ECE385 Faculty. (n.d.). [Lab 5 description](https://learn.intl.zju.edu.cn/bbcswebdav/pid-101276-dt-content-rid-1361038_1/xid-1361038_1)

[3] ECE385 Faculty. (n.d.). [Introduction to SystemVerilog (pdf)](https://learn.intl.zju.edu.cn/bbcswebdav/pid-101280-dt-content-rid-1361044_1/xid-1361044_1)

[4] ECE385 Faculty. (n.d.). [Introduction to Quartus Prime in the lab manual.](https://learn.intl.zju.edu.cn/bbcswebdav/pid-101280-dt-content-rid-1361046_1/xid-1361046_1)