ECE 385

2021 FA

LAB 6
Simple Computer SLC-3.2
in Systemverilog

Zhou Qishen, Xie Mu

Introduction

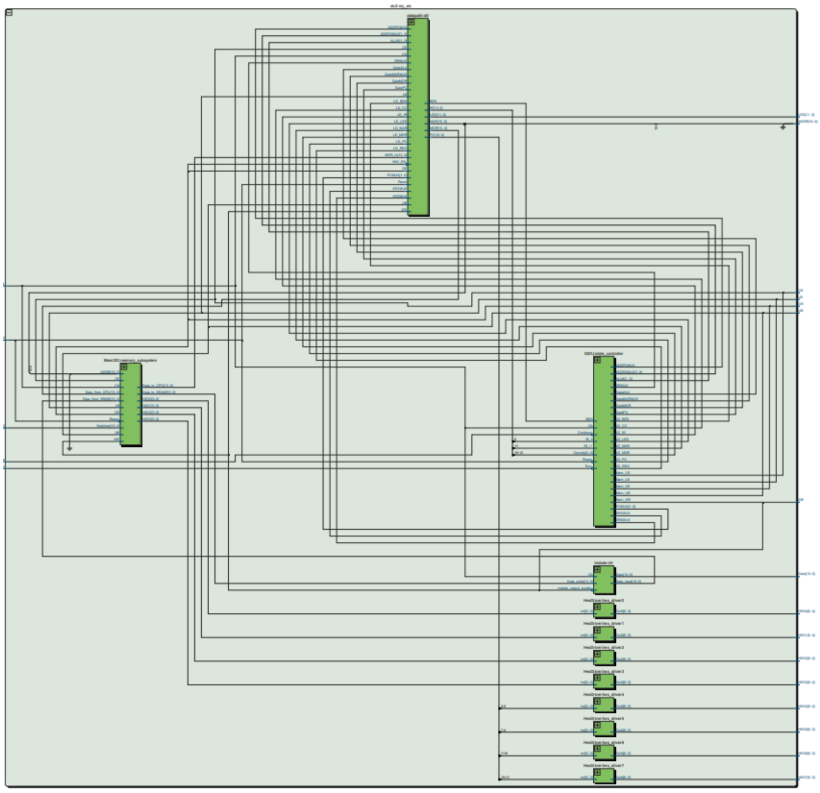
In lab6 we designed a simplified lc3 processor digitally with System Verilog. The processor can load and decode the 16 bits instructions that pre-loaded in the SRAM, and execute some basic logic operations like Add, And, Not, and it can change the Program Counter (PC) based on the instruction to realize Jump and Branch.

Description & Diagram

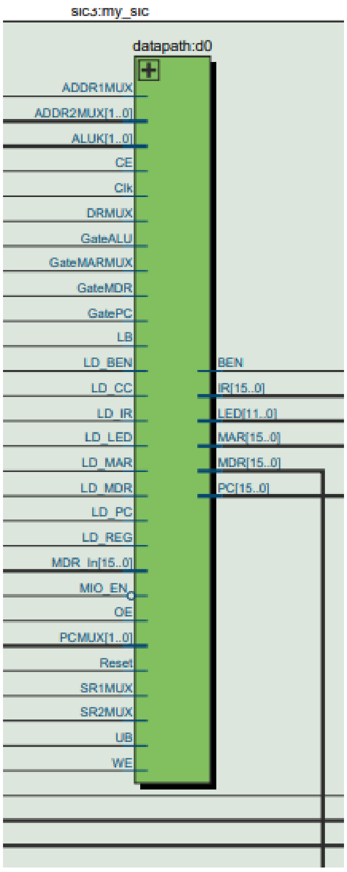
Written Distription

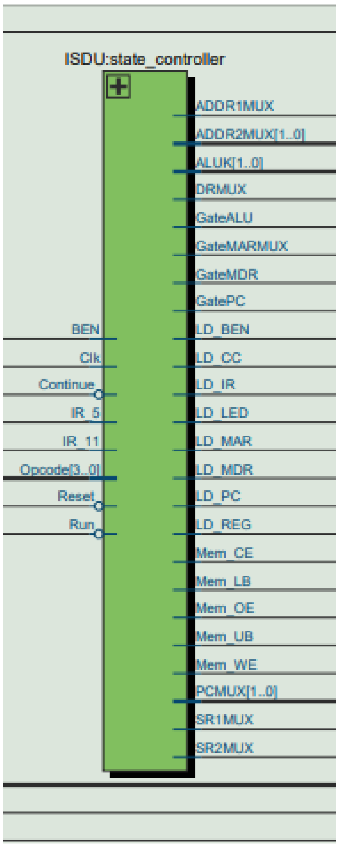
The available operation if the SLC3 is ADD, AND, NOT, LDR, STR, BR, JSR, JMP. The instruction is processed in three steps: Fetch, Decode, and Execute. The Fetch state ask the processor load the value in the PC to the Memory Address Register (MAR), then the memory IO part will load the memory content at that address to the Memory Data Register (MDR), then that will be load to the Instruction Register (IR). At the Decode part, the most 4 bits will be used to determine the detailed instruction. Base on the type of the instruction, the rest part will used to determine the Data Register (Data), Source Register (SR), offset and the operation type. Then the Execute part will carry out the operation and load the result to the target space.

Block Diagram

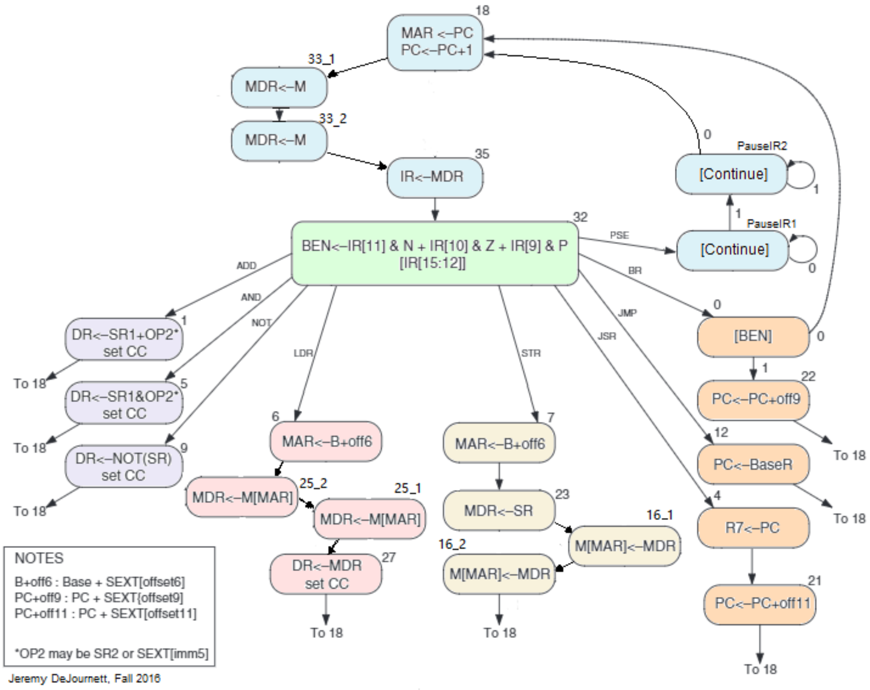


Block Diagram





State Diagram



State Diagram of Control Logic

Module Description

Module: MEM2IO
Description: This module controls the input and output of SRAM.
Purpose: The MEO2IO module connect the SLC3 processor to the SRAM, and read the value loaded by the switch and output the result to the HEX driver when the address is -1.

Module: ALU
Description: This module provides Arithmetic logic operation.
Purpose: This unit take two inputs, one from source register, the other from the MUX controlled by IR. The IR will also control the unit to operate one of the 4 operations: ADD, NOT, AND, or do nothing.

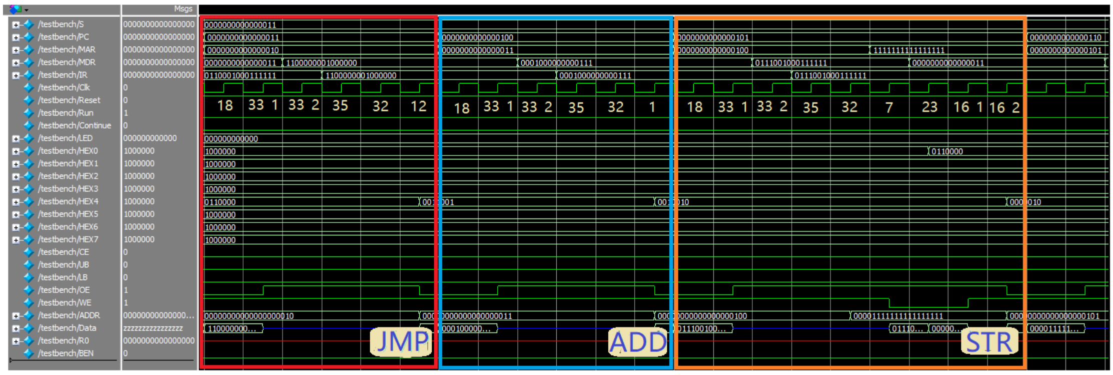
Module: Datapath
Description: this module connects all the part in the slc3 processor.
Purpose: It is used to transfer all the data during several modules, and send the instructions after the decode is finished.

Module: HexDriver
Description: Define the HEXDRIVER to display on LCD screen
Purpose: Transfer the binary signal to hex and display it in the board.

Module: ISDU
Description: The FSM of the whole slc3 processor that control the slc3 based on the instruction and program counter and send control signal to other modules.
Purpose: The ISUD contains all the state that the slc3 processor can achieve, and it will control the other modules by sending control signals that suitable for the current state.

Module: regfile
Input: LD_REG, Clk, Reset, BUS, DR_MUX, SR1_MUX, SR2,
Output: SR1_out, SR2_out, SR_test
Description: This is register array of the processor
Purpose: Store data in register R1 - R7

Waveform



Post-Lab

What is MEM2IO used for, i.e. what is its main function?

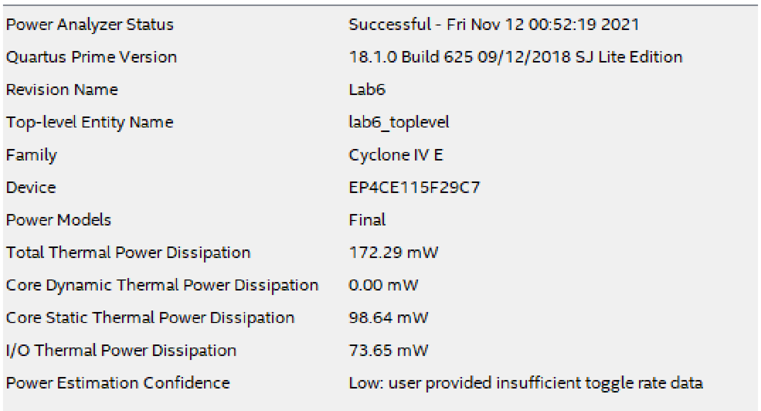
MEM2IO is the control logic of the SRAM module and connect it to the SLC3 processor. It can also load the value loaded in the switch and output the result to the HEX display when XFFFF is in the MAR.

What is the difference between BR and JMP instructions?

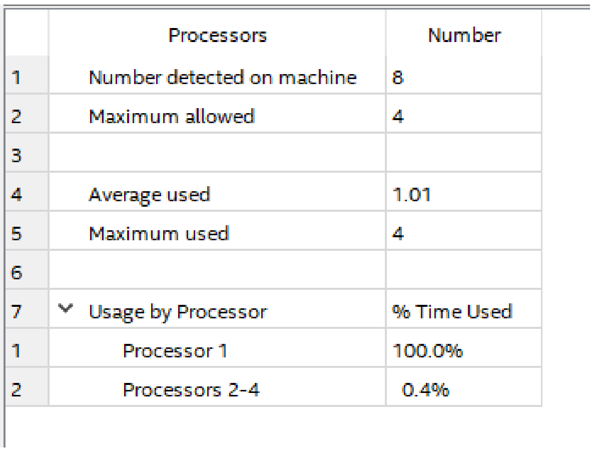
The JMP instruction changes PC without any condition. The BR can change the PC when the nzp part is meet the current condition.

What is the purpose of the R signal in Patt and Patel? How do we compensate for the lack of the signal in our design? What implications does this have for synchronization?

The R signal will be sent to the control logic when the memory is ready. However, in the FPGA design, the R signal will create a latch, and to design it is difficult. Since the memory are guaranteed to be ready within two clock cycles, so we use two clock cycles to ensure that the memory is ready instead of R signals to avoid synchronize problem.



Power Analyzer Status	Successful - Fri Nov 12 00:52:19 2021
Quartus Prime Version	18.1.0 Build 625 09/12/2018 SJ Lite Edition
Revision Name	Lab6
Top-level Entity Name	lab6_toplevel
Family	Cyclone IV E
Device	EP4CE115F29C7
Power Models	Final
Total Thermal Power Dissipation	172.29 mW
Core Dynamic Thermal Power Dissipation	0.00 mW
Core Static Thermal Power Dissipation	98.64 mW
I/O Thermal Power Dissipation	73.65 mW
Power Estimation Confidence	Low: user provided insufficient toggle rate data



	Processors	Number
1	Number detected on machine	8
2	Maximum allowed	4
3		
4	Average used	1.01
5	Maximum used	4
6		
7	⌄ Usage by Processor	% Time Used
1	Processor 1	100.0%
2	Processors 2-4	0.4%

Conclusion

1. Finally, our project is success. However, we met several problems during the whole progress, one of the problems is that our MUXs are connect mistakenly during the demo part, and the other is that our SRAM are not correctly flashed, and the result is quite different from the ModelSim. To avoid this kind of problem, we should not only flash the SRAM multiple times, but also check the content manually by the control panel.
2. In addition, the provided pin assignment is not correct, and this will lead to serious problem when loading the project to the FPGA board. We should check all the provided materials before using them next time.