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.

Written Description and Diagram of the SLC3

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.

Description to the Module

Post Lab Question

1. 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.
2. The JMP instruction changes PC without any condition. The BR can change the PC when the nzp part is meet the current condition.
3. 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.

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.