Multi-cycle CPU report

Chang Zihao
20206018
Cui Yuxuan
20206019

Contents

1 Introduction

Lab4 is required to design a Multi-cycle CPU. A Multi-cycle CPU processes an instruction per every four clock cycle.

1.1 Data Path

1. Update the PC to hold the address of the next instruction and fetch the instruction at the address in PC.
2. Decode the instruction.
3. Execute the instruction.
4. Save data into the memory.

1.2 Control-Multi

1. Operation to be performed by ALU.
2. Whether register file needs to be written.
3. Signals for multiple intermediate multiplexors.
4. Whether data memory needs to be written.
5. Control the four states of the CPU.

2 Design

As the Description of TB file mentioned, We do not need to implement memory and register ourselves. So we design the path to connect them together. Then we have to design some necessary modules.

2.1 Module design

2.1.1 ALU module

ALU has been designed in lab1. But this time we need to modify the ALU slightly. At the beginning, I took out the Cout alone to deal specifically with branch instructions It works, but not necessary. So I put the Cout back and cancel it.

2.1.2 PC module

The PC is a state element that holds the address of the current instruction. It is updated at the end of every clock cycle.

2.1.3 add module

The adder is responsible for incrementing the PC to hold the address of the next instruction. It takes two input values, adds them together, and outputs the result

2.1.4 Sign extend module

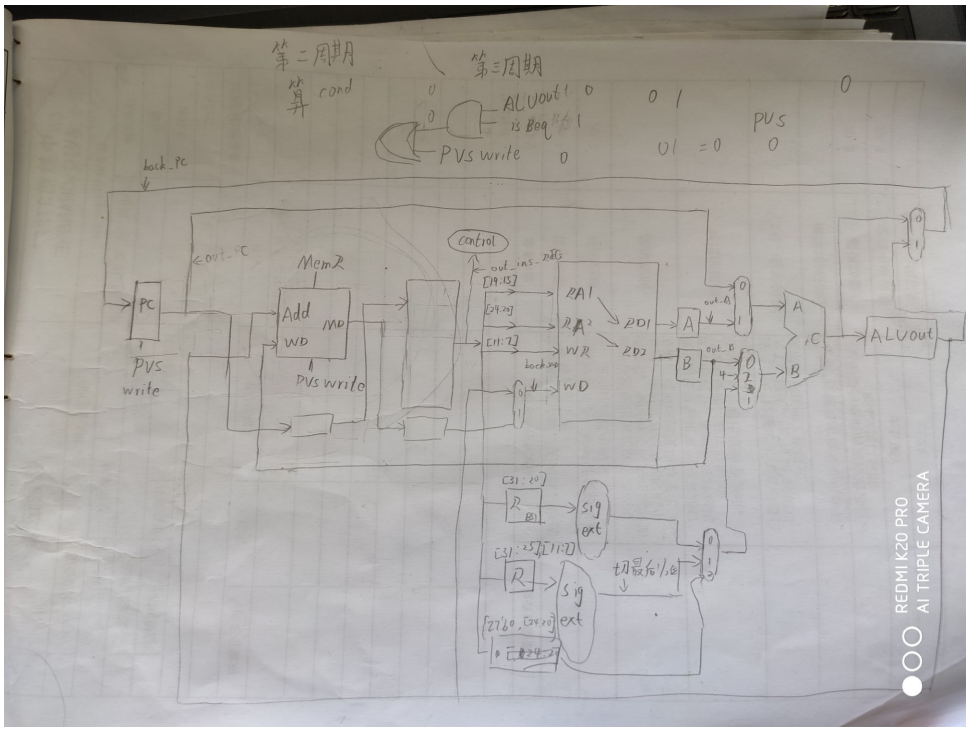
Single extend module is used to increase the number of bits of a binary number while preserving the number's sign and value.

2.1.5 MUX module

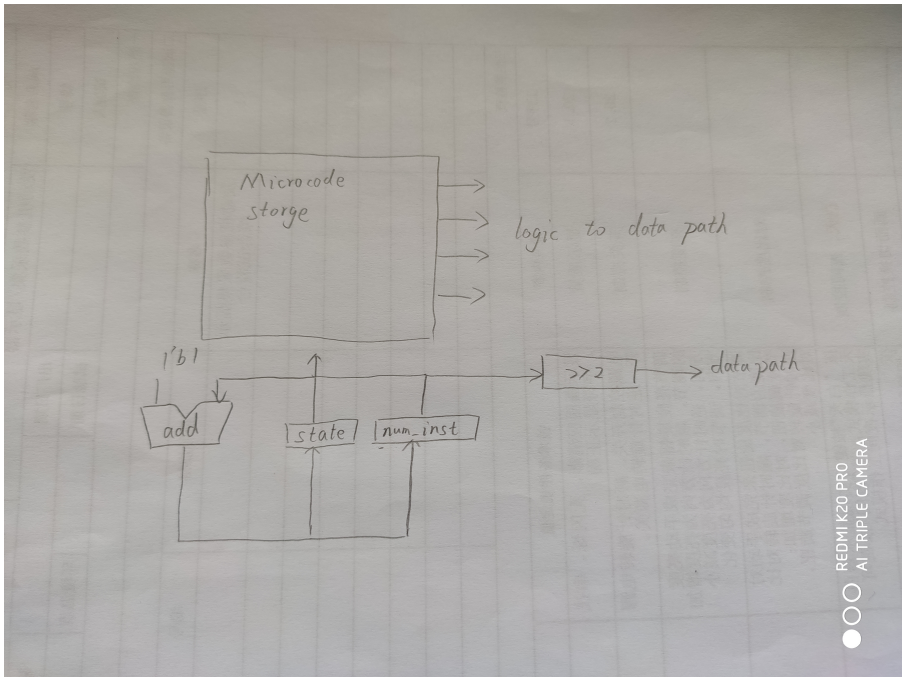
Mux, a data selector, which was used to select which the single will be output.

3 Implementation

The whole circuit was shown below.



There might be something different with the circuit.



The control module has four state, ”01” is IF,”10” is ID, ”11” is EX, and last ”00” is MEM and WB.

4 Evaluation

I pass all the test in the testbench folder.

```
# Finish:          88 cycle
# Success.
# ** Note: $finish    : D:/programming/EE312/lab4/RTL/testbench/TB_RISCV_inst.v
#    Time: 985 ns  Iteration: 1  Instance: /TB_RISCV
# 1
```

```
VSIM 2> run -all
# Finish:         292 cycle
# Success.
# ** Note: $finish    : D:/programming/EE312/lab4/RTL/testbench/TB_RISCV_forloop.v(166)
#    Time: 3025 ns  Iteration: 1  Instance: /TB_RISCV
# 1
```

```
# Finish:       37604 cycle
# Success.
# ** Note: $finish    : D:/programming/EE312/lab4/RTL/testbench/TB_RISCV_sort.v(1
#    Time: 376145 ns  Iteration: 1  Instance: /TB_RISCV
# 1
```

5 Conclusion

The multi-cycle CPU is more diffticult to design but it can avoid the disadvantage of the single cycle CPU. Such as the clock cycle will be determined by the longest possible path, which is not the most common instruction.