Computer Organization, Summer 2020

Lab 4 : Pipeline CPU

Due: 2020/08/21 23:55

1. Goal

In Lab 4, you need to modify the single cycle processor designed in Lab3 to a pipelined processor and implement an advanced version pipelined CPU which can handle hazard.

2. Demands

A. Please use ModelSim as your HDL simulator.

B. One person forms a group. Please attach your student ID as comments in the top of the file. (Ex: Lab4_Student ID.zip) The type of compressed file must be "zip". (Other form of file will get -10%.)

C. Reg_file.v(negative-edge triggered), Program_Counter.v, Testbench.v are supplied.

D. For each pipeline register, it should contain the fields for data and control signals.

E. Instruction set: we will test part of the instructions which have been implemented in previous lab: ADD, SUB, AND, OR, NOR, SLT, SLL, SRL, ADDI, lw, sw, beq, bne, "jump". And we will not test "jal", "jr". You may remove the circuits for jr and jal in your design.

3. Data/Control Hazard description

a. Code:

Data hazard:

ADD, ADDI, SUB, AND, OR, NOR, SLT, SLL, LW and SW.

- Need to implement Hazard Detection and Forwarding
  (i.e. Forwarding.v and HazardDetectionUnit.v)
- Need to stall pipelined CPU if it detects load-use.
- Need to forward data if instructions have data dependency.

(Bonus) Control hazard:

BEQ, BNE, JUMP

- **Modify Hazard Detection Unit**. Once a branch instruction is taken or a jump instructions performed, you should flush the IF/ID, ID/EX, and EX/ MEM pipeline registers, and then fetch the correct instruction from the new PC value

b. Testbench :

Please use **CO_P4_test_1.txt** to test **data hazard**, **CO_P4_test_2.txt** to test **control hazard**.

CO_P4_test_1.txt

```
addi   $1, $0, 5
addi   $2, $0, 2
addi   $3, $0, 11
addi   $4, $0, 6
sw     $1, 4($0)
addi   $7, $1, 10
lw     $5, 4($0)
and    $8, $5, $3
```

Result:

r1 = 5; r2 = 2; r3 = 11; r4 = 6; r5 = 5; r7 = 15; r8 = 1; r29 = 128;

data_mem[1] = 5;

others are 0.

CO_P4_test_2.txt

```
       addi  $1,  $0,  2
       addi  $2,  $0,  2
       addi  $3,  $0,  2
       addi  $4,  $0,  4
       addi  $5,  $0,  -1
       beq   $1,  $2,  L1
       slt   $9   $2   $4
       sw    $1   4($0)
       or    $10  $1   $4
L1:    addi  $6   $0   2
       addi  $1   $1   3
       addi  $2   $2   3
       and   $7   $3   $4
       sub   $8   $5   $3
```

Result:

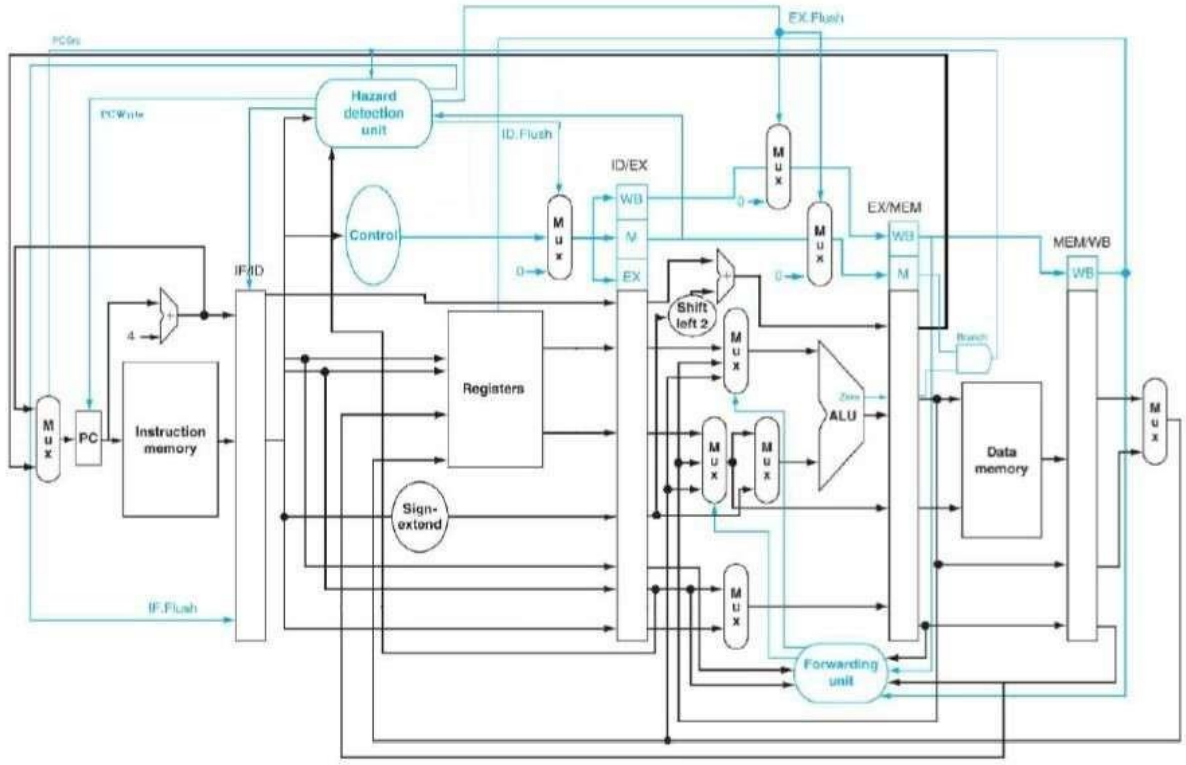
r1 = 5; r2 = 5; r3 = 2; r4 = 4; r5 = -1; r6 = 2; r7 = 0; r8 = -3; r29 = 128;

data_mem[1]=0

others are 0.

4. Architecture

a. Diagram



According to the above diagram, in this lab you should implement a five stage pipelined processor with IF, ID, EX, MEM, and WB stages. You should insert a pipeline register between each two stages. Each pipeline register should contain the fields for **data** and control **signals**. The pipeline registers are written when the positive clock edge occurs.

b. The description of pipeline stage

The function of each stage is described as follows:

IF stage: In this stage, the processor fetches an instruction from the instruction memory and performs PC + 4.

ID stage: In this stage, the processor decodes the instruction to generate the control signals, reads two source registers, and generates the sign- extended immediate value.

EX stage: In this stage, ALU_Ctrl generates control signals for function units according to ALUOp. At the same time, Register Write ID and branch target are also determined in this stage.

MEM stage: In this stage, the processor accesses data memory according to the control signals. The modification of PC from branch taken instruction is also performed in this stage.

WB stage: In this stage, the processor will write the value into register file according to the control signal when negative clock edge occurs.

c. Description of pipeline register

Please design four pipeline registers. Each pipeline register must be “positive-edge triggered”, has default value 0. Then, insert these pipeline registers into your single-cycle CPU designed in Lab3 to accomplish the pipelined CPU required in this lab.

DO NOT set any delay time for the sequential circuits of the pipelined registers designed by you.

5. Report

The context must include:

a. Architecture diagrams
b. Hardware module analysis
c. Finished part
d. Problems you met and solutions
e. Summary

The report is at most 3 pages.

6. Grade

Total score: 100+20 bonus pts. COPY WILL GET A 0 POINT!

a. Data hazard: 80 pts
b. (bonus) Control hazard: 20 pts
c. Report: 20 pts
d. incorrect file form: -10pts

7. Hand in your assignment

Please upload the assignment to the E3.
Put all of *.v source files and report into same compressed file.

(Use your student ID to be the name of your compressed file and must have the form of “Lab4_student ID.zip”)

8. Q&A

If you have any question, just send email to TAs by new e3.