Computer Organization
Lab 5: Advanced Pipelined CPU

Updated: 2022/6/17

Due:2022/6/23 23:55

1. Goal

Modify your Lab 4 CPU design, and implement an advanced pipelined CPU with (1)hazard detection and (2) data forwarding.

2. Homework Requirement

a. Please use Vivado as your HDL simulator (if the execution results in your environment is different from ours, you need to bring your laptop to the lab and demo it to us).

b. Please **attach student IDs as comments** at the top of each file.
The file type of your report should be **PDF**

c. Please add the files listed below into one directory named "your_student_id", and zip itas " your_student_id.zip".

The file structure in this lab should be (for example, id=310551072):

```
310551072/
├── report_310551072.pdf
└── (Any .v file that you need except testbench.v )
```

d. Please **do not add unnecessary or given files and folders** (like .DS_Store, __MACOSX)

e. In this lab, you can directly **modify your design in Lab 4**, and add some files if you need. Please make sure that your design can be run correctly on the testbench we provided.

f. If your CPU is not pipelined, you'll get 0 scores

g. Your CPU needs to support the following instructions:

I. Basic Instructions:

- **i. ADD**
- **ii. ADDI**
- **iii. SUB**
- **iv. AND**
- **v. OR**
- **vi. SLT**
- **vii. SLTI**
- **viii. LW**
- **ix. SW**
- **x. MULT**

II. Advanced Instructions

Instruction	Op
BEQ	000 100
BNE	000 101
BGE	000 001
BGT	000 111

(Modify **Hazard Detection Unit** to flush useless pipeline registers (IF/ID, ID/EX, EX/MEM) if a branch launch)

III. You must implement (1) **Hazard Detection Unit** and (2) **Forwarding Unit**.

IV. Your CPU needs to forward data if instructions have data dependency.

V. Your CPU needs to stall pipelined CPU if it detects a load-use

h. Testbench ("CO_P5_test_1.txt"):

Try to solve the date hazards in I1/I2, I5/I6, I8/I9, I9/I10 by using forwarding unit and Hazard Detection Unit.

```
I1:   addi   $1,$0,16
I2:   mult   $2,$1,$1
I3:   addi   $3,$0,8
I4:   sw     $1,4($0)
I5:   lw     $4,4($0)
I6:   sub    $5,$4,$3
I7:   add    $6,$3,$1
I8:   addi   $7,$1,10
I9:   and    $8,$7,$3
I10:  slt    $9,$8,$7
```

Result:

r1 =16;
r2 = 256;
r3 = 8;
r4 = 16;
r5 = 8;
r6 = 24;
r7 = 26;
r8 = 8;
r9 = 1;
data_mem[1] = 16;
others = 0

* Maximum clock count: 17

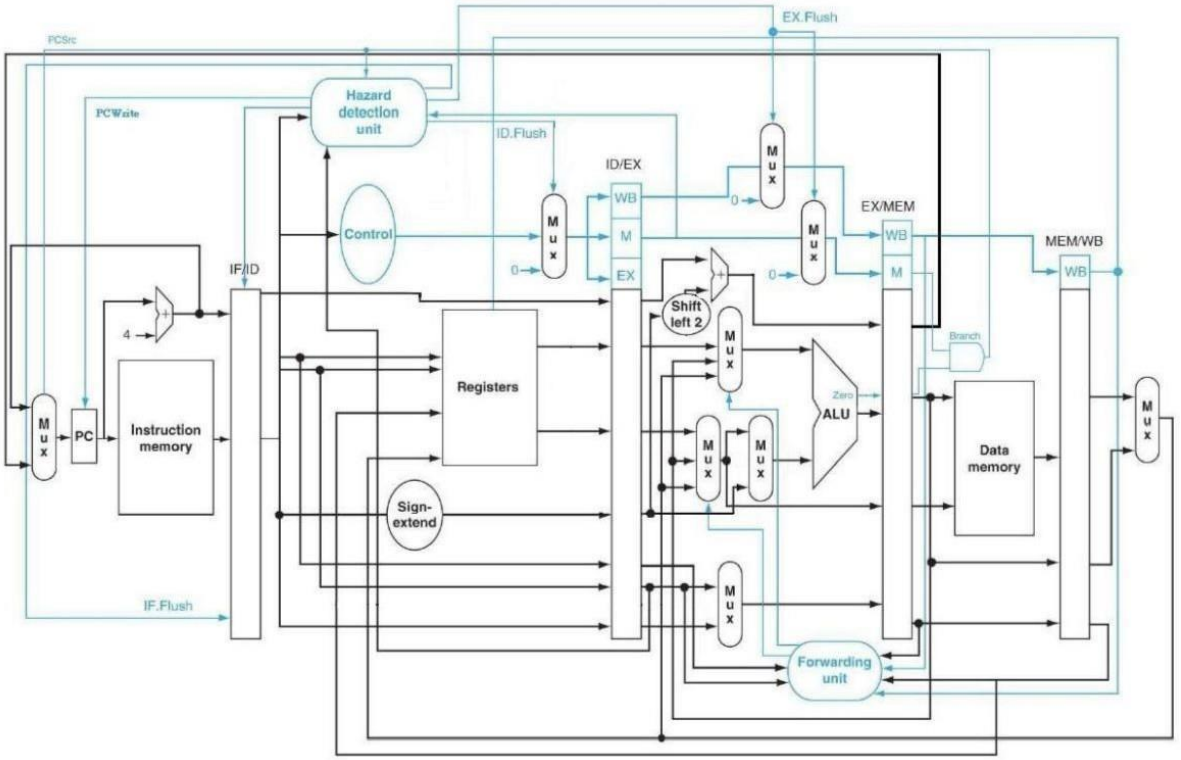
i. **Testbench (“CO_P5_test_2.txt”):**

```
I1:     addi    $2, $0, 3
I2:     sw      $2, 0($0)
I3:     addi    $2, $0, 1
I4:     sw      $2, 4($0)
I5:     sw      $0, 8($0)
I6:     addi    $2, $0, 5
I7:     sw      $2, 12($0)
I8:     addi    $2, $0, 0
I9:     addi    $5, $0, 16
I10:    addi    $8, $0, 2
I11:    beq     $0, $0, 2
I12:    addi    $2, $2, 4
I13:    bge     $2, $5, 6
I14:    lw      $3, 0($2)
I15:    bgt     $3, $8, 1
I16:    beq     $0, $0, -5
I17:    addi    $3, $3, 1
I18:    sw      $3, 0($2)
I19:    beq     $0, $0, -8
```

Result:
r2 = 16;
r3 = 6;
r5 = 16;
r8 = 2;
data_mem[0] = 4
data_mem[1] = 1
data_mem[3] = 6
others = 0

* Maximum clock count: 70

3. Architecture Diagram



4. Report

a. Your Architecture
b. Hardware Module Analysis
c. Problems You Met and Solutions
d. Result
e. Summary

5. Grade

a. **Total:** 100 points (plagiarism will get 0 points)
   - Report: 20 points(please use **pdf format**)
   - Hardware design: 80 points (including hidden case)
b. **Late submission:** Score * 0.8 before **6/27**. After 6/27, you will get 0.
c. **Wrong format:** 10 points punishment

6. Q&A

If you have any question, it is recommended to ask in the facebook discussion forum