Computer Organization, Fall 2013

Lab 5: Pipelined CPU II

Due: 2013/12/23

1. Goal

Modifying the CPU designed in lab4 and implementing an advanced version pipelined CPU.

2. Demand

a. Basic instruction set(60%): ADD, SUB, AND, OR , SLT, ADDI, LW, SW, MULT

- Must to implement **Hazard Detection** and **Forwarding** Unit.
- Need to forward data if instructions have data dependency.
- Need to stall pipelined CPU if it detects load-use.
- Mu lt rd, rs, rt ；rd=rs*rt

0	Rs	Rt	Rd	0	24(0x18)

b. Advanced instructions 1(40%): BEQ, BNEZ, BGEZ, BGT

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

c. Advanced instruction 2(10%): JR

- Modify JR implemented in previous lab so that can fit in pipeline architecture and be careful about hazard problem.
- TA won't provide any testbench about JR instruction; you have to do verification by yourself.

d. Report (10%)

e. Demo: Got 0 point if you did not present.

CO_P5_test1.txt:

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

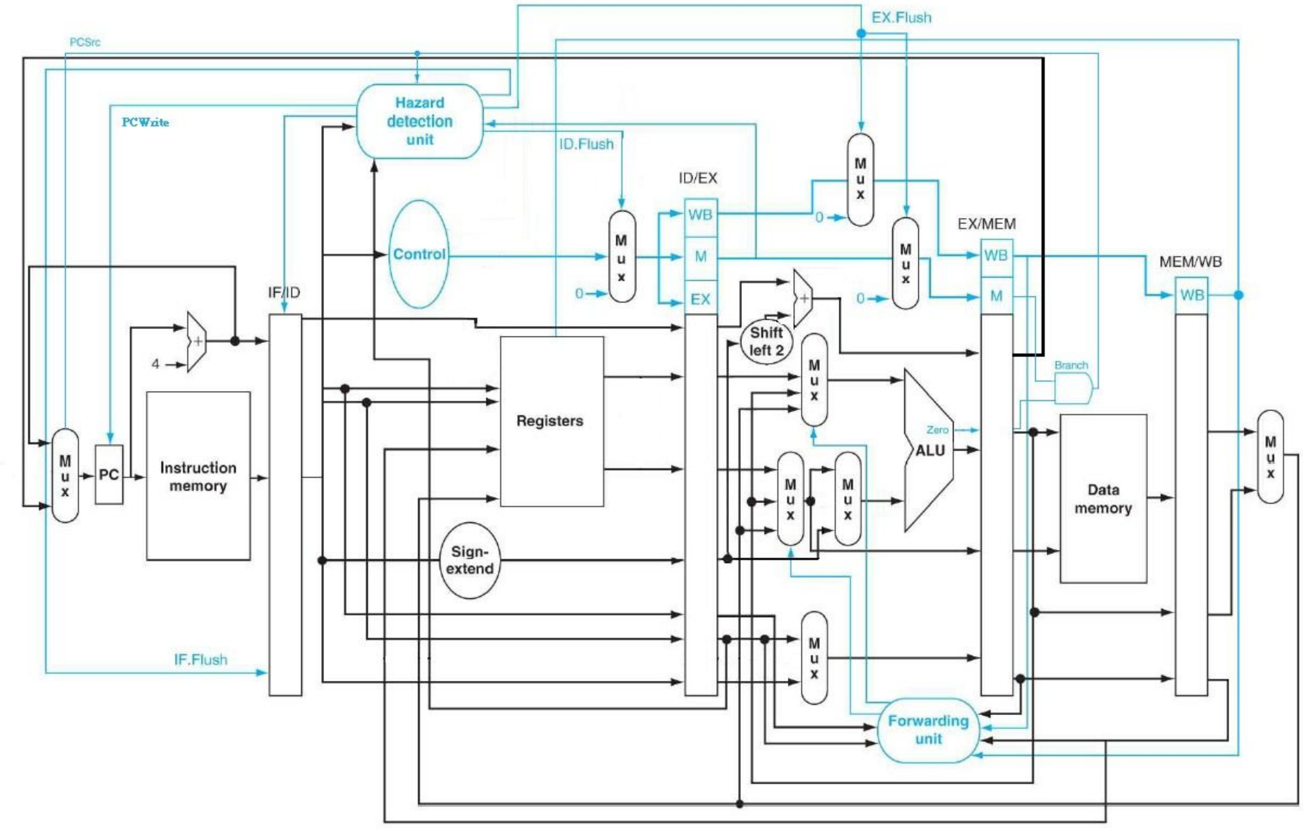
Assembly:

```
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; date_mem[1] = 16;

3. Architecture.



5. Grade

- Total score: 120(report included) pts. COPY WILL GET 0 POINT!
- Delay: 10%off/day
- Demo date: to be announced

6. 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)