VE370 Hw 4 周维凯 518021911039

1. IF: address of "lw $to, -4($sp)".

ID: IF/DWrite: X

Instruction: 100011 11101 01000 1111111111111100

EX: ReadData1: Reg[$sp]

ReadData2: Reg[$t0]

RegWrite: 1

MemToReg: 1

MemWrite: 0

MemRead: 1

ALUOp: 00

RegDst: 0

ALUSrc: 1

SignExtend: $\underbrace{1111 \cdots\cdots}_{30'b1} 1100$

Rs: 11101

Rt: 01000

Rd: 11111

MEM: RegWrite: 1

MemToReg: 1

MemWrite: 0

MemRead: 1

Result: Reg[$sp]-4

WriteData: Reg[$t0]

Reg: 0100

WB: RegWrite: 1

MemToReg: 1

Reg: 0100

Result: Reg[$sp]-4

MemData: Mem[Reg[$sp]-4]

2.(a). $3 between L2 and L3

$6 between L3 and L4.

(b). Hazards between L2 and L3,

Hazards between L3 and L4.

```
L1: sw   $18, -12($8)
L2: lw   $3,  8($18)
      nop
      nop
L3: add $6, $3, $3
      nop
      nop
L4: or  $8, $9, $6
```

8+4=12 clock cycles

(c). Hazards between L2 and L3.

```
L1: sw   $18, -12($8)
L2: lw   $3,  8($18)
      nop
      nop
L3: add $6, $3, $3
L4: or  $8, $9, $6
```

6+4=10 clock cycles

(d). Hazards between L2 and L3

```
L1: sw  $18, -12($8)
L2: lw  $3, 8($18)
      nop
L3: add  $6, $3, $3
L4: or   $8, $9, $6
```

5+4=9 clock cycles

3. (a).
```
    lw  $t0, Offs(Rs)
    bne Rt, $t0, L1
    jr  Rd
L1: .....
```

(b). First, we can change the register file so that it can support reading three registers at one time.

Besides, we should pass the calculated signal in ID stage to MEM stage. In MEM stage, we should add a compartor to compare between R[Rt] and M[R[Rs]+Offs]. Besides, we should add a MUX in MEM stage to choose between the R[Rd] and the calculated address from ID stage to be the new PC.

(c). We must add beqmcontrol signal. If there is a beqm instruction, it will be 1 otherwise it will be 0. Together with zero signal, it will help the MUX in MEM stage to choose between the calculated address and R[Rd].

(d). It may introduce control hazards.

```
beqm $16, $17, Offs($18)
nop
add  $19, $20, $21
```

Here is control hazards because we know new PC in MEM stage, which is 2 clock cycles later.

4. (a). There will be no problems with this MIPS.

(b). 1CC: PCWrite=1, IF/DWrite=1, IDstall=0, MUX1=X, MUX2=X
2CC: PCWrite=1, IF/DWrite=1, IDstall=0, MUX1=X, MUX2=X
3CC: PCWrite=1, IF/DWrite=1, IDstall=0, MUX1=00, MUX2=00
4CC: PCWrite=1, IF/DWrite=1, IDstall=0, MUX1=10, MUX2=00
5CC: PCWrite=1, IF/DWrite=1, IDstall=0, MUX1=01, MUX2=00

(C). new inputs: Rd from ID/EX, the writeback register number from EX/MEM, RegWrite
new outputs: no extra output needed.

For the first two instructions: sub and lw. Since we don't have forwarding unit, $6 is written back by sub in CC5. But lw will get $6 in CC3 from register file. So we need to add nops between these two instructions.

Similarily, for the following instructions, we also need add nops.