1. IF: The address of instruction lw $t0, -4($sp) is in PC PC+4

ID: The instruction code of lw $t0, -4($sp), which is 100011 11101 01000 1111111111111100

EX: RegWrite=1, Mem2Reg=1, Branch=0, MemRead=1, ALUOp=00, ALUSrc=1
R[$sp], R[$t0] ; imme : -4 in 32 bits ; RegDst=0.
Rs=29, Rt=8, Rd=31.

MEM: Branch=0. MemWrite=0, MemRead=1, RegWrite=1, Mem2Reg=1.
R[$sp]-4 . IF/ID.RegRtd = 8 . Input of ALU: -4
branch address : PC+4-16 ALUzero output: !(R[$sp]-4).

WB: MemRead: 1, RegWrite=1, Mem2Reg=1
value : [$sp]-4 , M[R[$sp]-4]
IF/ID.RegRtd = 8.

2. a) L2, L3 register $3
L3, L4. register $6. ✓

b)
```
SW $18, -12($8)
lw $3, 8($18)
nop
nop
add $6, $3, $3
nop
nop
or $8, $9, $6
```
2 data hazard in (a).

Then, there will be 8+4=12 clock cycles.

c)
```
SW $18, -12($8)
lw $3, 8($18)
nop
nop
add $6, $3, $3
or $8, $9, $6
```
Reg $3 between L2 & L3

There will be 6+4=10 clock cycles.

d) SW $18, -12($8)
lw $3, 8($18)
nop
add $6, $3, $3
or $8, $9, $6

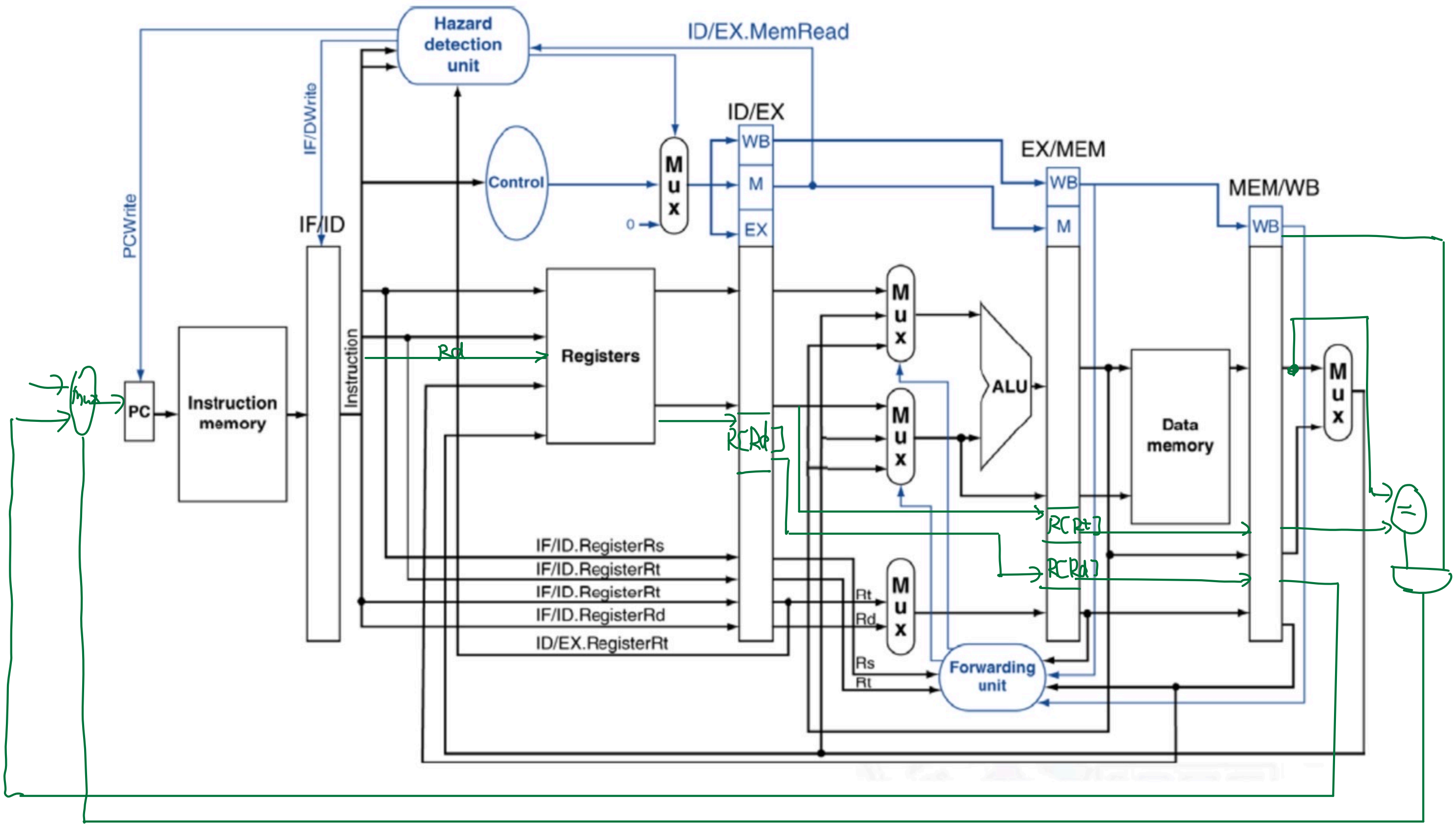
Reg $3 between L2 & L3.

There will be 5+4=9 clock cycles.

3. a) lw $t0, offset(Rs)
beq $t0, Rt, L1
L1: jr Rd

} = beqm Rd, Rt, offset(Rs).

b) A mux is needed before PC to select Rd and PC+4
Also, Register needs 3 read output to give Rs, Rt, Rd
ALU to compare whether two inputs are equal and a AND gate are needed control the mux,



c) A control signal of beqm is needed

d) Yes. For example: add $8, $9, $10
beqm $20, $8, 0($9)
There will occur hazard for $8.

4. a, Since there is no load-Use Data Hazard in the instruction
It will run successfully.

b) 1. There will be hazard of $6. of inst 1. & 2.
Thus, in the fourth clock cycle:
forwarding unit will give forwarding A: 10, forwarding B: 00
hazard detection will give PC, IF/ID: 1,
mux before ID/EX: 0.

2. There will be hazard of $6. of inst 1. & 3.
Thus, in the fifth clock cycle:
forwarding unit will give forwarding A: 01, forwarding B: 00
hazard detection will give PC, IF/ID: 1,
mux before ID/EX: 0.

During other clock cycle, hazard detection will always give PC, IF/ID: 1, mux before ID/EX: 0; forwarding unit will always give forwarding A and B: 00.

c) If no forwarding, we have to use hazard detection to add nop instructions. Then, input of MEM/WB. RegWrite EX/MEM. RegRd, ID/EX. RegRs, ID/Ex. RegRt, MEM/Wb. RegRd Ex/MEM. RegWrite should be added and compared in Hazard Detection unit.

eg: L1: sub $6, $2, $1

L2: lw $3, 8($6)

L3: lw $2, 0($6)

for L1 & L2, Ex/MEM. RegRd, ID/EX. RegRs, ID/EX. RegRt
EX/MEM. RegWrite.

for L1 & L3, MEM/WB. RegWrite, MEM/WB. RegRd
ID/EX. RegRs, ID/EX. RegRt

new output, for each hazard we detect, we need add one more nop.