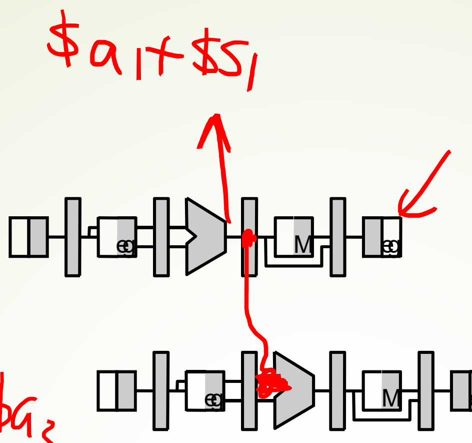
EGC442 Class Notes 4/11/2023

Baback Izadi
Division of Engineering Programs
bai@engr.newpaltz.edu

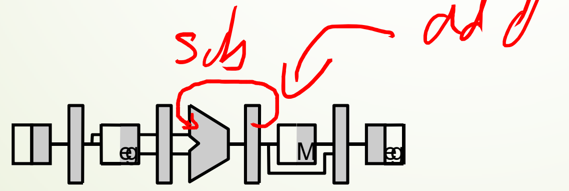
add t_1, $a_1, $s_1

sub $t_2, $t_1, $a_3



$t_1

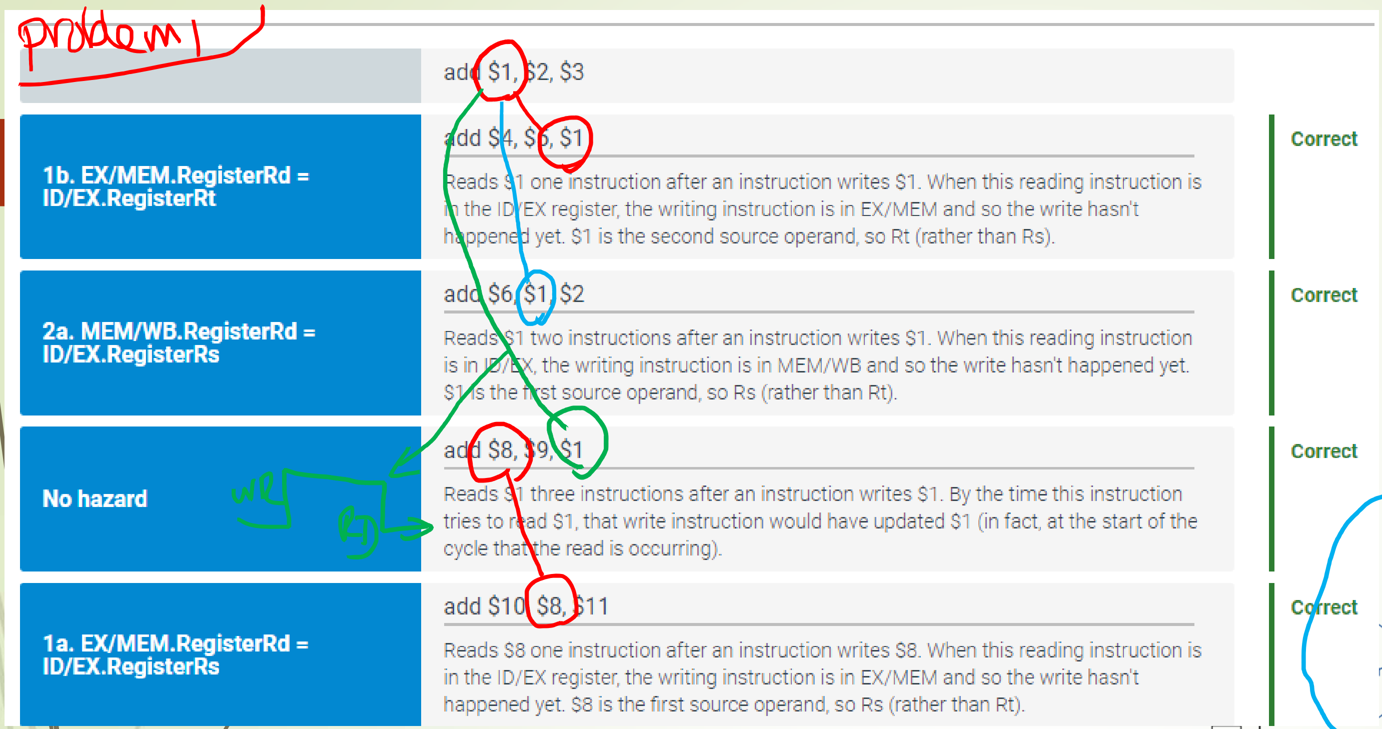
a_1+s_1



vd of add same as rs of sub

∴ Forwarding needed.

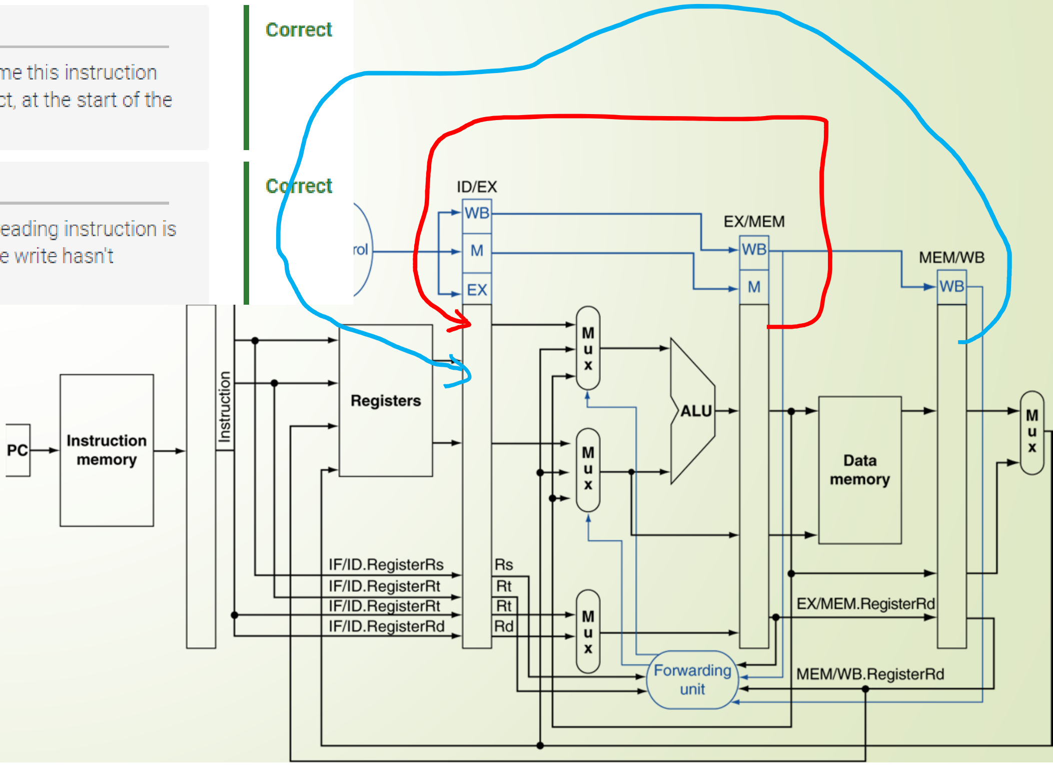
problem 1



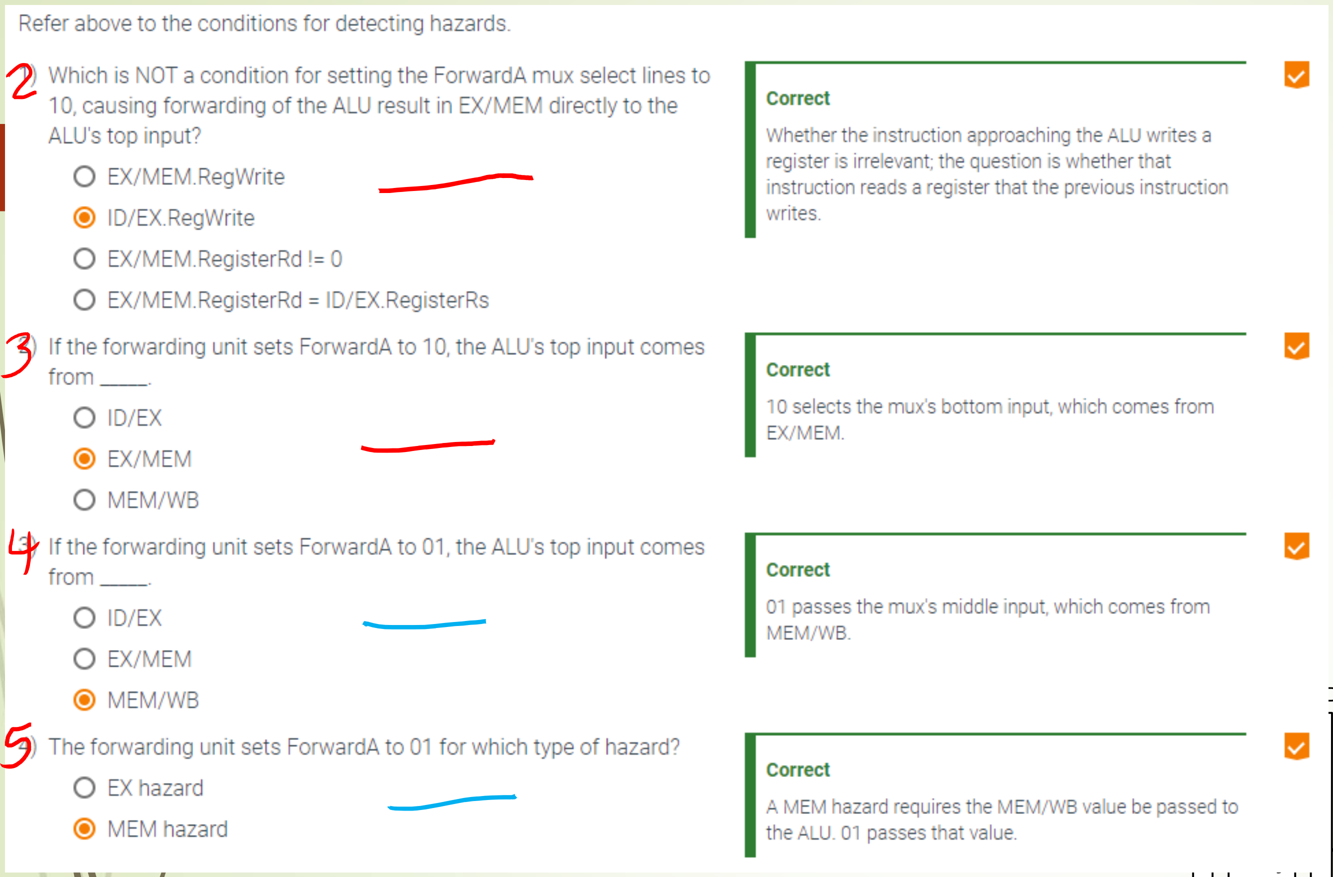
	add $1, $2, $3	
1b. EX/MEM.RegisterRd = ID/EX.RegisterRt	add $4, $5, $1 Reads $1 one instruction after an instruction writes $1. When this reading instruction is in the ID/EX register, the writing instruction is in EX/MEM and so the write hasn't happened yet. $1 is the second source operand, so Rt (rather than Rs).	Correct
2a. MEM/WB.RegisterRd = ID/EX.RegisterRs	add $6, $1, $2 Reads $1 two instructions after an instruction writes $1. When this reading instruction is in ID/EX, the writing instruction is in MEM/WB and so the write hasn't happened yet. $1 is the first source operand, so Rs (rather than Rt).	Correct
No hazard	add $8, $9, $1 Reads $1 three instructions after an instruction writes $1. By the time this instruction tries to read $1, that write instruction would have updated $1 (in fact, at the start of the cycle that the read is occurring).	Correct
1a. EX/MEM.RegisterRd = ID/EX.RegisterRs	add $10, $8, $11 Reads $8 one instruction after an instruction writes $8. When this reading instruction is in the ID/EX register, the writing instruction is in EX/MEM and so the write hasn't happened yet. $8 is the first source operand, so Rs (rather than Rt).	Correct

we

[illegible]



problem 2 → 5



Refer above to the conditions for detecting hazards.

2) Which is NOT a condition for setting the ForwardA mux select lines to 10, causing forwarding of the ALU result in EX/MEM directly to the ALU's top input?

- EX/MEM.RegWrite
- [✓ ID/EX.RegWrite]
- EX/MEM.RegisterRd != 0
- EX/MEM.RegisterRd = ID/EX.RegisterRs

Correct

Whether the instruction approaching the ALU writes a register is irrelevant; the question is whether that instruction reads a register that the previous instruction writes.

3) If the forwarding unit sets ForwardA to 10, the ALU's top input comes from ____.

- ID/EX
- [✓ EX/MEM]
- MEM/WB

Correct

10 selects the mux's bottom input, which comes from EX/MEM.

4) If the forwarding unit sets ForwardA to 01, the ALU's top input comes from ____.

- ID/EX
- EX/MEM
- [✓ MEM/WB]

Correct

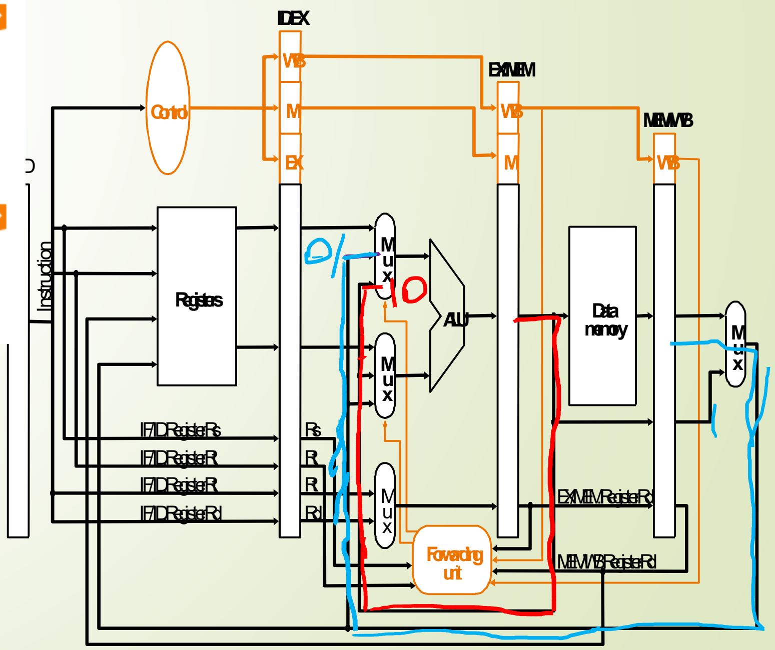
01 passes the mux's middle input, which comes from MEM/WB.

5) The forwarding unit sets ForwardA to 01 for which type of hazard?

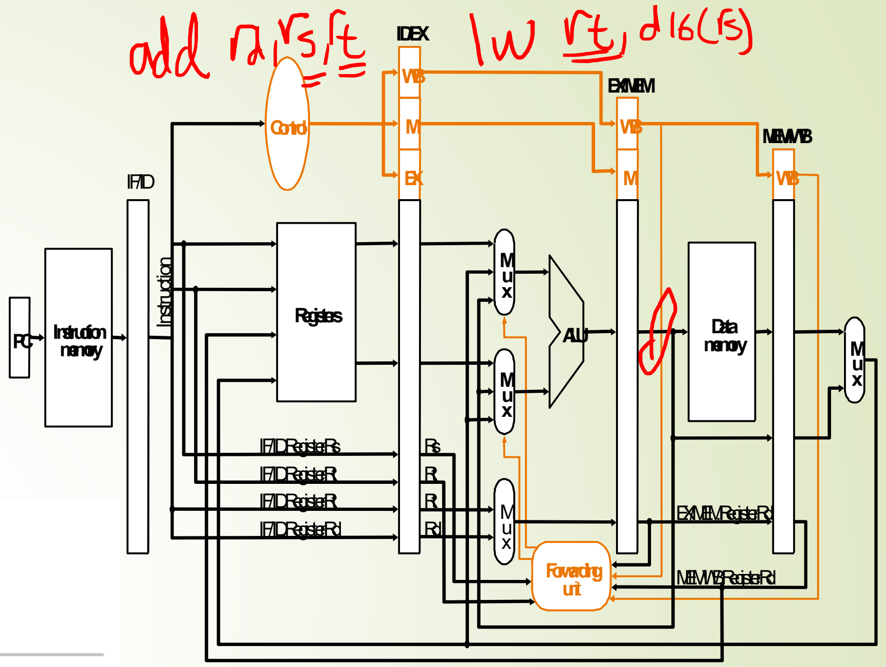
- EX hazard
- [✓ MEM hazard]

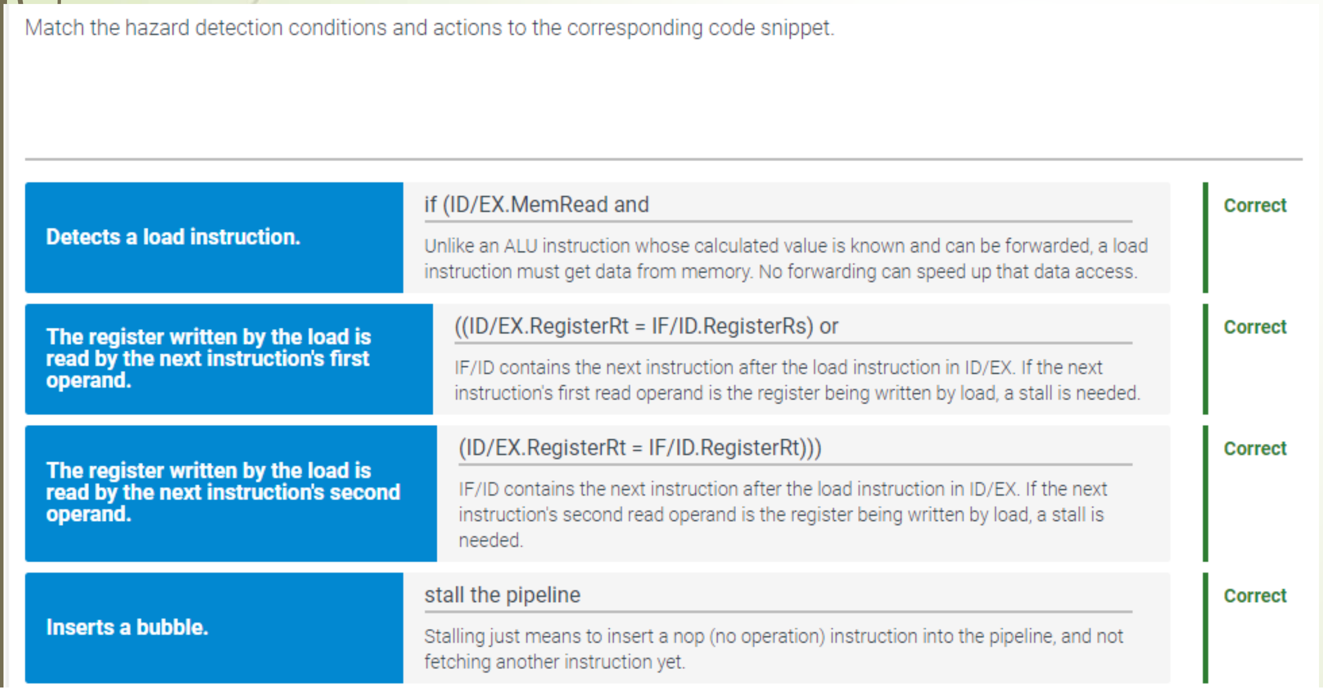
Correct

A MEM hazard requires the MEM/WB value be passed to the ALU. 01 passes that value.



Problem 6




Match the hazard detection conditions and actions to the corresponding code snippet.

Detects a load instruction.	if (ID/EX.MemRead and Unlike an ALU instruction whose calculated value is known and can be forwarded, a load instruction must get data from memory. No forwarding can speed up that data access.	Correct
The register written by the load is read by the next instruction's first operand.	((ID/EX.RegisterRt = IF/ID.RegisterRs) or IF/ID contains the next instruction after the load instruction in ID/EX. If the next instruction's first read operand is the register being written by load, a stall is needed.	Correct
The register written by the load is read by the next instruction's second operand.	(ID/EX.RegisterRt = IF/ID.RegisterRt))) IF/ID contains the next instruction after the load instruction in ID/EX. If the next instruction's second read operand is the register being written by load, a stall is needed.	Correct
Inserts a bubble.	stall the pipeline Stalling means to insert a nop (no operation) instruction into the pipeline, and not fetching another instruction yet.	Correct

7) For the code below,

a. On the diagram, mark and identify all the data dependencies in the code given below and identify which depe
will cause data hazards without forwarding hardware.

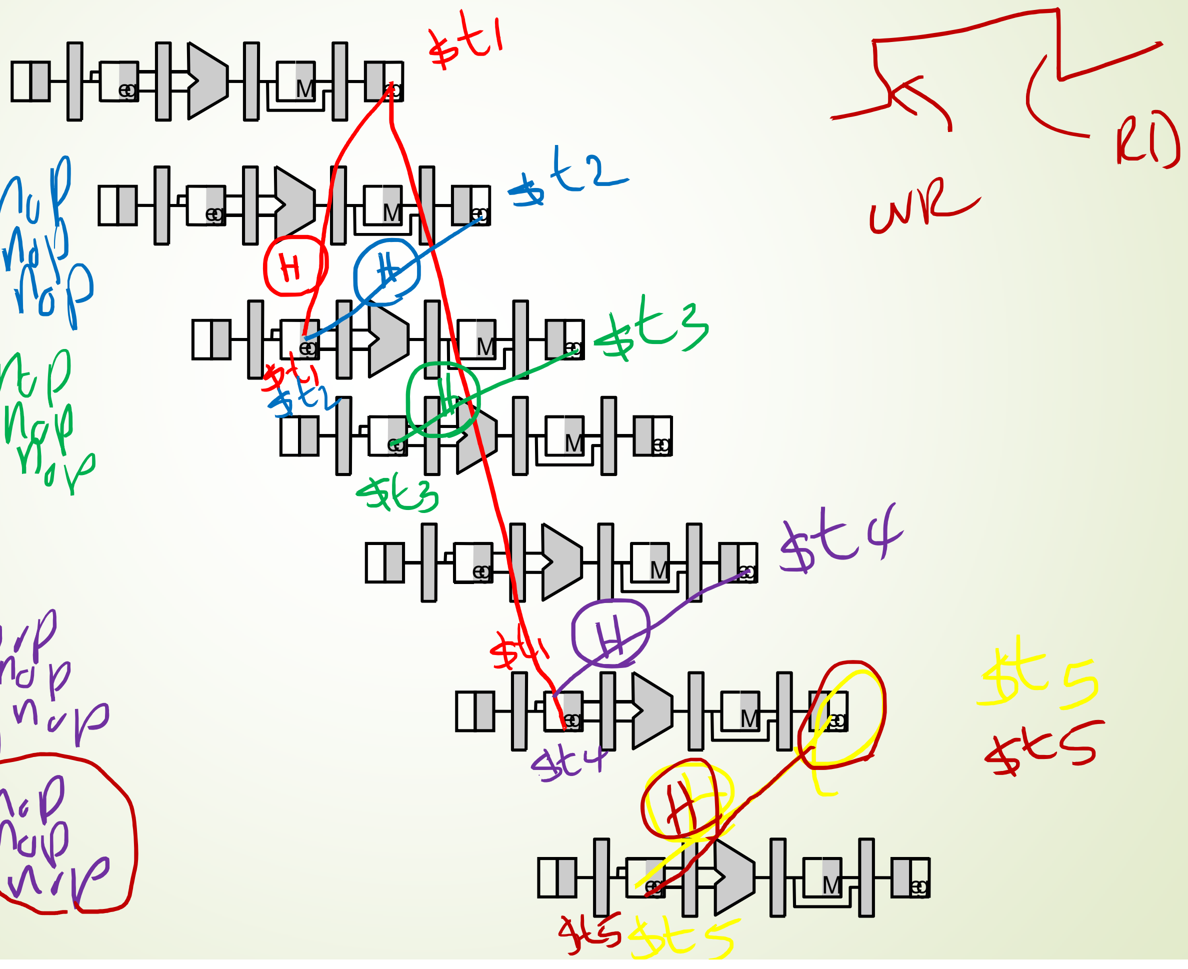
```
lw   $t1, 0($t0)
lw   $t2, 4($t0)
add  $t3, $t1, $t2
sw   $t3, 12($t0)
lw   $t4, 8($t0)
add  $t5, $t1, $t4
sw   $t5, 16, ($t0)
```

nop
nop
nop

nop
nop
nop

nop
nop
nop

nop
nop
nop



a. Assuming there is no special hardware that is added for forwarding. Add "nop" instructions to the code to avoid the data hazards.

```
lw $t1, 0($t0)
lw $t2, 4($t0)
nop
nop
nop
add $t3, $t1, $t2
nop
nop
nop
sw $t3, 12($t0)
lw $t4, 8($t0)
nop
nop
nop
add $t5, $t1, $t4
nop
nop
nop
sw $t5, 16 ($t0)
```

b. How many clock cycles does it take to execute the code in part b. 23

c. Using forwarding, clearly show how it can be used to resolve data hazards. If a bubble needs to be added, simply make a marking as below. (use the next page for your answer) lw $t1, 0($t0)

lw $t2, 4($t0)

(hardware nop)

add $t3, $t1, $t2

sw $t3, 12($t0)

mem

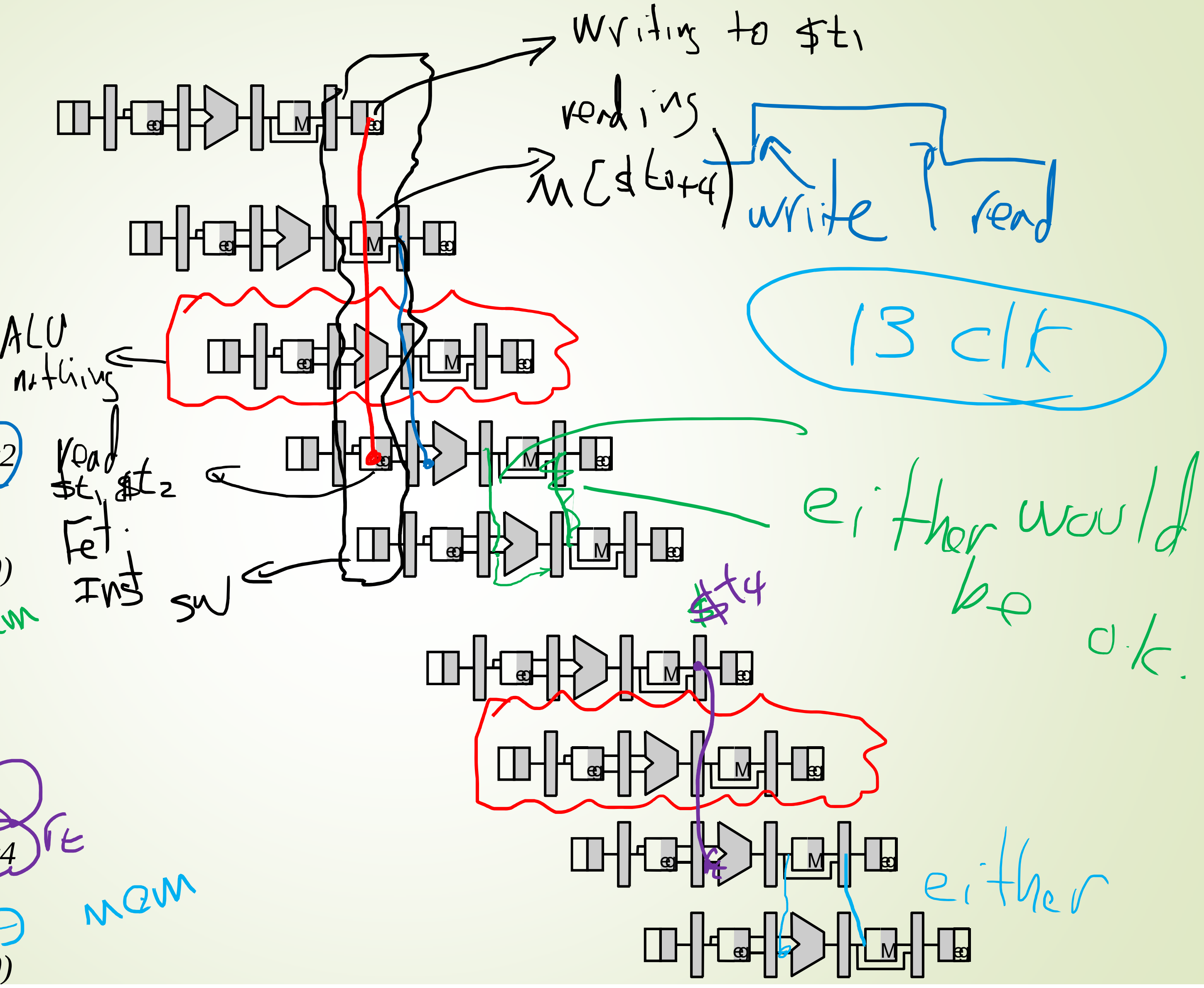
lw $t4, 8($t0)

rt

add $t5, $t1, $t4

mem

sw $t5, 16($t0)

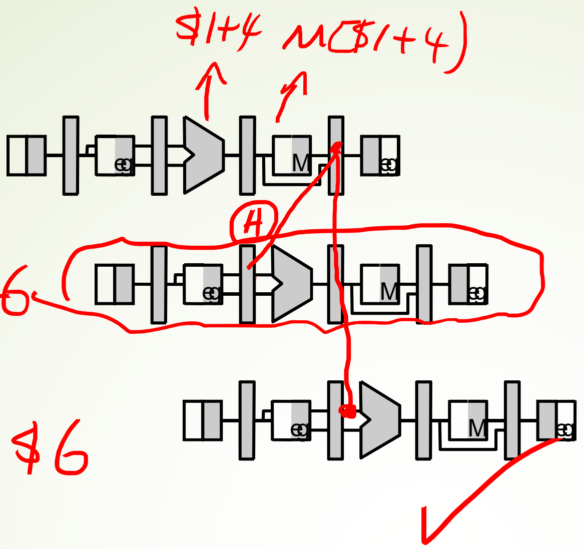


lw $2,($1)4

nop

~~add $4,$2,$6~~

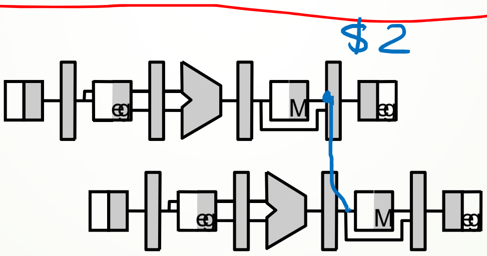
add $4,$2,$6



$2 ← M($1+4)

lw $2,($1)4

sw $2,($8)12



No bubble needed