

Solution 5-2)

loop $ld\ x10, 0(x13)$ IF ID EX ME | WB

$ld\ x11, 8(x13)$ IF ID EX | ME WB

$add\ x12, x10, x11$ IF ID | .. EX ME WB

$addi\ x13, x13, -16$ IF | .. ID EX ME WB

$bneq\ x12, 100p$ | .. IF ID EX ME WB

$ld\ x10, 0(x13)$ IF ID EX ME WB

$ld\ x11, 8(x13)$ IF ID EX ME WB

$add\ x12, x10, x11$ IF ID .. EX | ME WB

$addi\ x13, x13, -16$ IF .. ID | EX ME WB

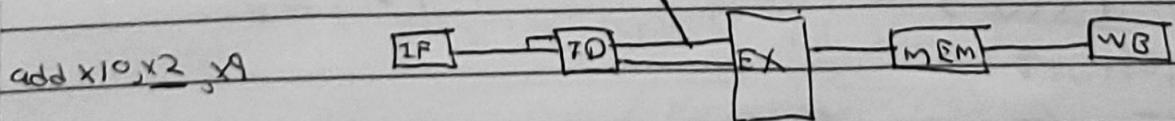
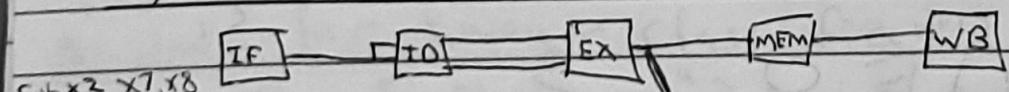
$bneq\ x12, 100p$ IF | ID EX ME WB

Problem 6

Solution 6.1) For each RAW dependency listed in table previous slide, give a sequence of at least three assembly statements that exhibits that dependency.

① EX to 1st Only

200 400 600 800 1000 1200



(1)

(2)

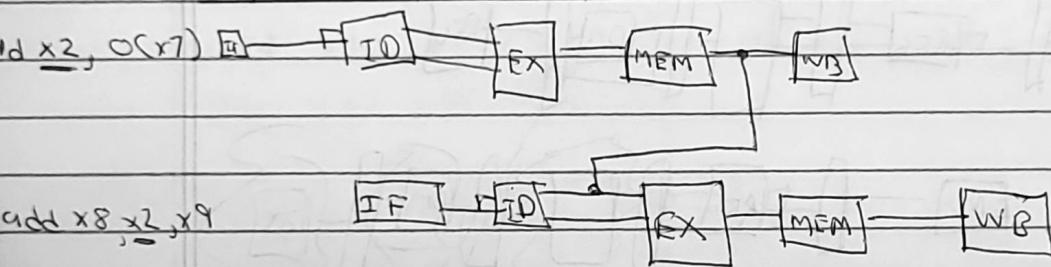
(2) MEM to 1st Only:-

add x2, 0(x7)

add x8, x2, x9

sub x15, x22, x23

200 400 600 800 1000 1200 1400

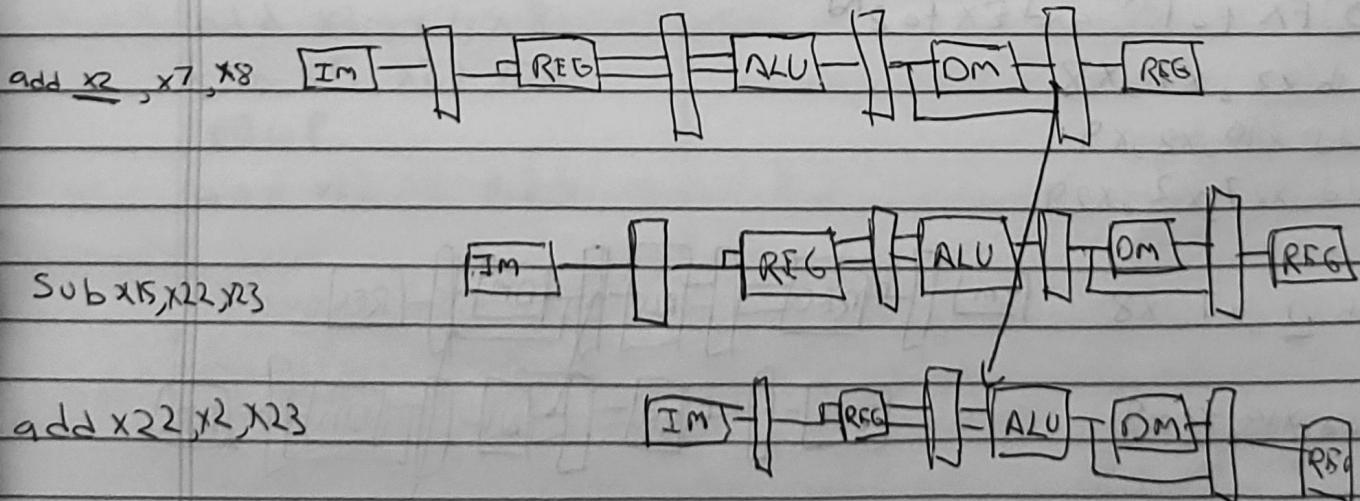


(3) EX to 2nd Only:-

add x2, x7, x8

sub x15, x22, x23

add x22, x2, x23

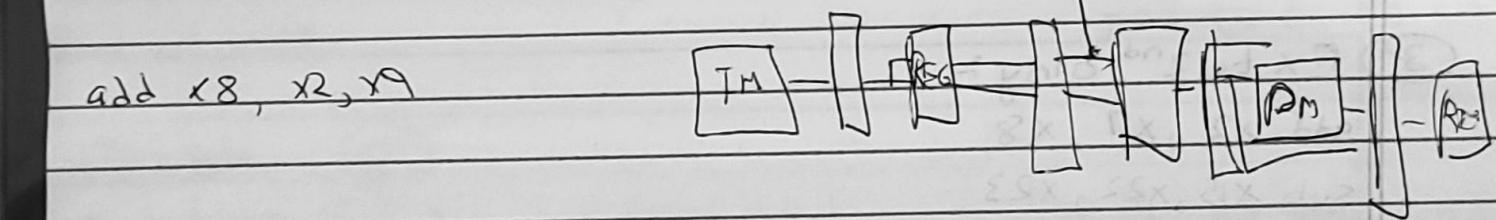
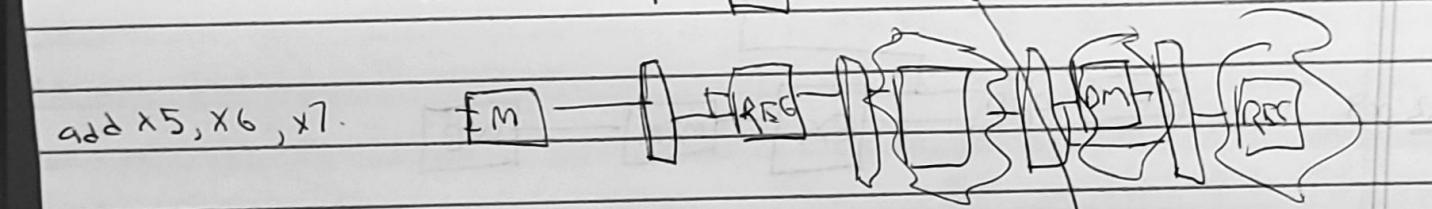
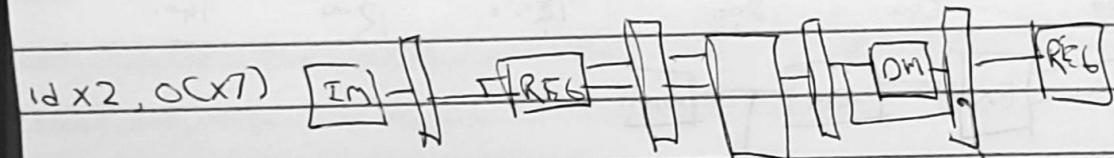


④ MEM to 2nd Only:

id x2, 0(x7)

add x5, x6, x7

add x8, x2, x9

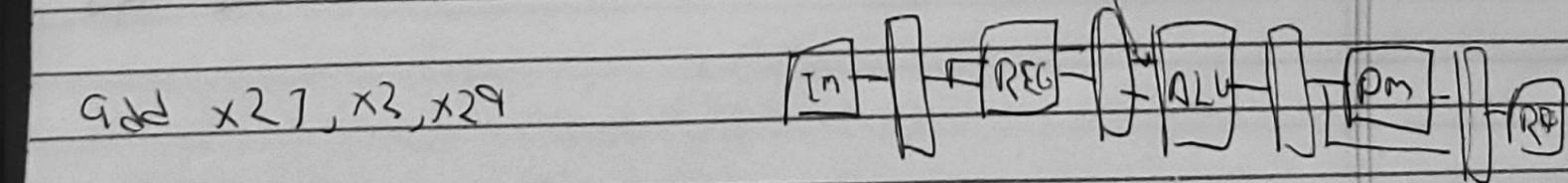
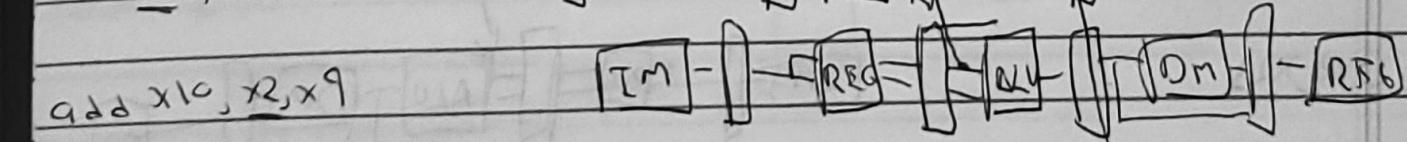
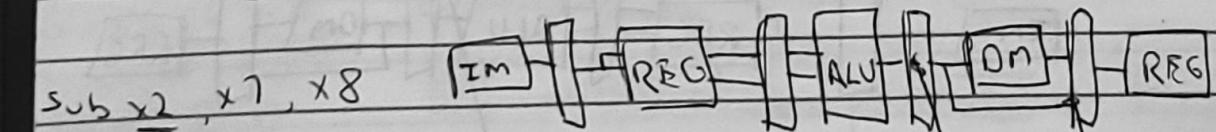


⑤ EX to 1st and EX to 2nd

sub x2, x7, x8

add x10, x2, x9

add x27, x2, x29



(4)

Solution G.2, G.3

EX to 1st only :-

sub x2, x7, x8

NOP

NOP

add x10, x2, x9

add x21, x28, x29

EX to 1st and EX to 2nd

sub x2, x7, x8

NOP

NOP

add x10, x2, x9

add x27, x2, x29

MEM to 1st only.

ld x2, 0(x7)

NOP

NOP

add x8, x2, x9

EX to 2nd Only

add x2, x7, x8

sub x5, x22, x23

NOP

add x22, x2, x23

MEM to 2nd only

ld x2, 0(x7)

add x5, x6, x7

NOP

add x8, x2, x9

Solution 6-4

Taking a weighted average of the number of NOPs for each from 6.2 gives $0.05 \times 2 + 0.2 \times 2 + 0.05 \times 1 + 0.1 \times 1 + 0.1 \times 2 = 0.85$ NOPs per instruction.

A CPT of 1.85, so $0.85 / 1.85 = 46\% \text{ are NOPs}$

Solution 6-5

The only RAW dependency that cannot be handled by load-use-data hazard with forwarding is from the MEM stage to the next instruction. 20% of instructions will generate one NOP for CPS of 1.2, $0.2 / 1.2 = 17\% \text{ NOPs}$

Solution 6-6

If we forward from the EX/MEM register only we have the following stalls/NOPs

RX to 1st : 0

This hazard needs EX/MEM register with which the forwarding unit eliminates the NOPs so NOPs = 0

MEM to 1st : 2

Needs the MEM/WB register cannot be solved so 2 NOPs must be inserted

Ex to 2nd : (1)

This hazard needs MEM/WB register and cannot be resolved so NOP must be inserted

MEM to 2nd (1)

This hazard needs MEM/WB register and cannot be resolved, so a NOP is inserted

Ex to 1st and 2nd

Ex to 1st can be resolved with the EX/MEM register but the Ex to 2nd needs the MEM/WB register and cannot be resolved so a NOP is inserted to resolve the Ex to 2nd hazard.

\Rightarrow If we forward from the MEM/WB register, we have the following stalls / NOPs

Ex to 1st : 1

This hazard needs the EX/MEM register and can be resolved with the MEM/WB register if the MEM/WB register forwards to the ALU after 1 NOP, So NOP = 1

MEM to 1st (1)

This hazard needs the MEM/WB register but 1 NOP must be inserted.

EX to 2nd: 0

This hazard needs the MEM/WB register and is resolved with 0 NOPs

MEM to 2nd: 0

This hazard needs the MEM/WB register and is resolved with 0 NOPs

EX to 1st and 2nd

EX to 1st cannot be resolved and needs 1 NOP, but the EX to 2nd can be resolved with the MEM/WB register

\Rightarrow CPI with Partial forwarding with MEM/WB register unavailable

$$\text{Avg NOP} = 0.05(0 + 0.2^*2 + c.Q^*1) + 0.1^*1 + 0.1^*1$$

$$= 0.85 \text{ stalls/Instruction}$$

$$CPI = 1.85$$

with EX/MEM to f, the unavailable.

$$\text{Avg NOP, of } 0.05(1 + 0.2^*1 + c.P)$$

$$= 0.35 \text{ stalls / Instruction}$$

$$CPI = 1.35$$

(9) Solution (6.7)

	No forwarding	EXMEM	MEM/WB	FU1 forwarding
CPI	1.85	1.65	1.55	1.2
Period	120ps	120ps	120ps	130ps
Time	222ps	198ps	162ps	156ps
Speedup	-ref	1.12	1.37	1.42

Solution (6.8)

CPI with full forwarding is 1.2

CPI for time travel for zeroing is 1.0

Clock period for full forwarding is 130

Clock period for zeroing by 2nd forwarding is 280

$$\text{Speedup} = T_{old}/T_{new} = (1.2 - 130)/(1 \times 280) = 0.28$$

Zero wait-state forwarding actually slows the CPU

~~Solutions~~ Problem ②

Solution 7.1

add $x15, x12, x11$

NOP

NOP

ld $x13, 4(x15)$

ld $x12, 0(x2)$

NOP

or $x13, x15, x13$

NOP

NOP

sd $x13, 0(x15)$

Solution 7.2

Not possible to reduce the Number of NOP

Solution 7.3

The code executes correctly, we need hazard detection only to insert a stall when the instruction following a load uses the result of the load. That does not happen in this case.

Solution 7.4

Because there are no stalls in the code
PCwrite and IF/IDWrite are

10

add TF ID EX ME WB

12 IF ID EX ME WB

1d IF ED EX ME WB

or IF ED EX ME WB

sd IF ED EX ME WB

Solution 48

the instruction that's currently in top ED

stage needs to stand it