

1. Maximum value among the delays = 145ps

$$\text{Clock cycle} = (145 + 12) \text{ps} = 157 \text{ps}$$

(a) No. of instructions = 1000

$$\text{No. of cycles} = 1000 + 6 - 1 = 1005$$

$$\text{Total time to execute} = 1005 \times 157 \text{ps}$$

$$= 1,57,785 \text{ps}$$

(b) Total no. of instructions =  $1000 + \frac{20}{100} \times 1000$

(20% of instructions incur a 1-cycle stall)

$$\text{No. of cycles} = 1200 + 6 - 1 = 1205$$

$$\begin{aligned} \text{Total time to execute} &= 1205 \times 157 \text{ps} \\ &= \underline{\underline{189,185 \text{ps}}} \end{aligned}$$



2. (a) Total latency of  $P_1 = 800ps$  (which is equally divided)

Clock period for processor  $P_1 = 800/5 = 160ps$

ALU latency of  $P_2$  is increased by  $250ps$

$\Rightarrow$  ALU latency =  $160 + 250 = 410ps$

Clock period for processor  $P_2 = 410ps$

(as  $410ps$  is the max time)

(b) Total no. of instructions to be executed by  $P_1 = x$

Total no. of instructions to be executed by  $P_2 = 9x/10$

( $T_1$ ) Time for  $P_1 = (x+5-1)160ps = (160x+640)ps$

( $T_2$ ) Time for  $P_2 = (9x/10+5-1)410ps = (369x+1640)ps$

for any  $x > 0$ ,  $T_2$  is always greater than  $T_1$ ,

(i.e.  $369x + 1640 > 160x + 640$ )

$$\Rightarrow 209x + 1000 > 0$$

$$\Rightarrow x > -4.78$$

Hence,  $P_1$  executes a given code in smaller time.

(c) Substituting the value of  $x = 5000$  in the above equations of  $T_1$  and  $T_2$



Execution time on  $P_1 = 160(5000) + 640$

Execution time on  $P_2 = 369(5000) + 1640$

$81 = 18,46,640ps$

3) (a) 1. add  $x_{14}, x_{12}, x_{11}$

Reason: ( $NOP_1$ )

$NOP_1 \leftarrow \begin{cases} \text{add } x_0, x_0, x_0 \\ \text{add } x_0, x_0, x_0 \end{cases}$

$x_{14}$  which is needed in instruction

2 is calculated in instruction 1.

2. add  $x_{15}, x_{14}, x_{12}$

To get the correct value of  $x_{14}$  to 2 NOPs are added.

3. ld  $x_{13}, 8(x_{13})$

Reason: ( $NOP_2$ )

4. ld  $x_{12}, 0(x_{14})$

$x_{13}$  needed in instruction 5

$NOP_2 \leftarrow \text{add } x_0, x_0, x_0$

is loaded in instruction 3.

5. and  $x_{13}, x_{15}, x_{13}$

1 NOP is added to make a difference b/w these instructions as 2.

~~$NOP \leftarrow \text{add } x_0, x_0, x_0$~~

Reason:

~~6. and  $x_{13}, x_{15}, x_{13}$~~

$NOP_3 \leftarrow \begin{cases} \text{add } x_0, x_0, x_0 \\ \text{add } x_0, x_0, x_0 \end{cases}$

Reason: ( $NOP_3$ )

6. ld  $x_{11}, 4(x_{13})$

$x_{13}$  needed in instruction 6

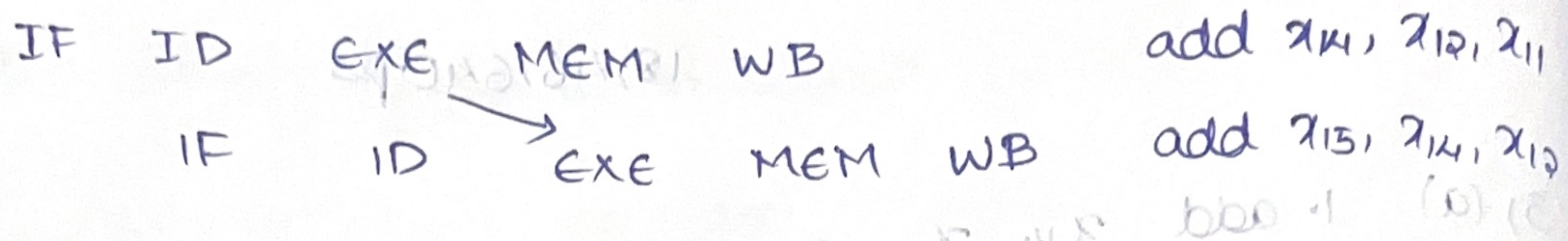
is calculated in instruction

7. sd  $x_{13}, 0(x_{15})$

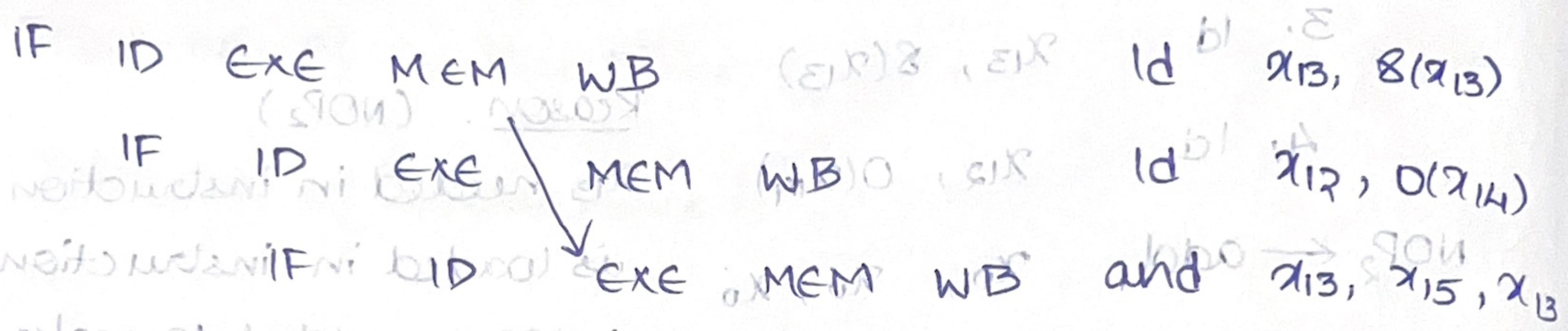
5. Two NOPs are added to make a difference of 2 to get the correct value.



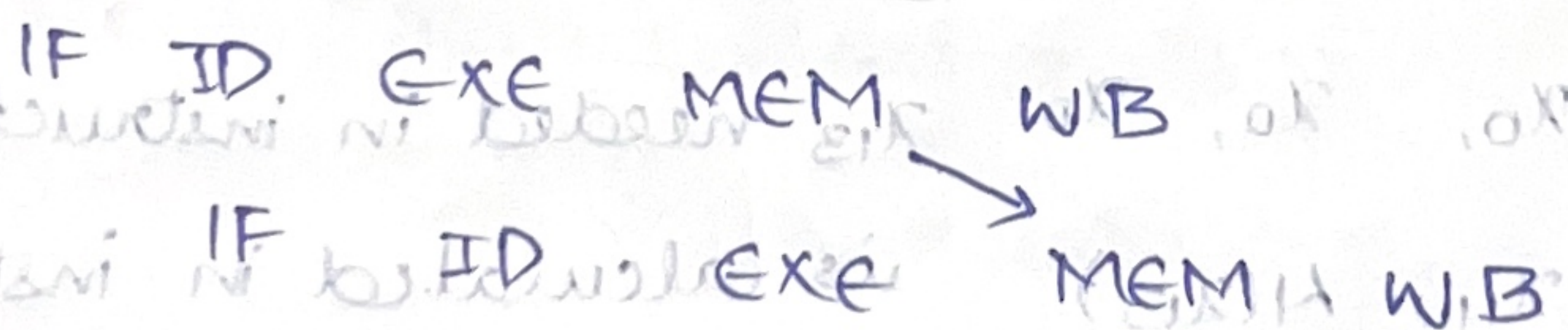
(b)  $r_{14}$  which is needed in instruction 2 in EXE stage can be forwarded from EXE stage of instruction 1. Hence, no NOPs needed.



$r_{15}$  which is needed in instruction 5 in EXE stage can be forwarded from MEM stage of instruction 3. Hence, no NOPs needed.



$r_{13}$  which is needed in instruction 6 in MEM stage (as it is ld) can be forwarded from MEM stage of instruction 5. Hence, no NOPs needed.



Hence, the code remains the same as no NOPs are to be inserted.