CS2323: COMPUTER ARCHITECTURE

VELMA DHATRI REDDY
AI21BTECH11030

1. Maximum value among the delays = 145ps

Clock cycle = (145 + 12) ps = 157 ps

(a) No. of instructions = 1000

No. of cycles = 1000 + 6 - 1 = 1005

Total time to execute = 1005 x 157 ps
= 1,57,785 ps

(b) Total no. of instructions = $1000 + \frac{20}{100} \times 1000$
= 1200

(20% of instructions incurs a 1 cycle stall)

No. of cycles = 1200 + 6 - 1 = 1205

Total time to execute = 1205 x 157 ps
= ~~1,89,185~~ = 189,185 ps

2. (a) Total latency of P1 = 800ps (which is equally divided)

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

ALU latency of P_2 is increased by 250ps

→ ALU Latency = 160 + 250 = 410ps.

Clock period for processor P2 = 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 P1 = (x+5-1)160ps = (160x+640)ps

(T_2) Time for P2 = (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

⇒ 209x + 1000 > 0

⇒ 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

= 8,00,640 ps

Execution time on P_2 = 369(5000) + 1640

= 18,46,640 ps

3) (a)

1. add x_{14}, x_{12}, x_{11}

$NOP_1 \leftarrow$
- add x_0, x_0, x_0
- add x_0, x_0, x_0

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

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

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

$NOP_2 \leftarrow$ add x_0, x_0, x_0

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

$NOP \leftarrow$ add x_0, x_0, x_0

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

$NOP_3 \leftarrow$
- add x_0, x_0, x_0
- add x_0, x_0, x_0

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

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

Reason: (NOP_1)
x_{14} which is needed in instruction 2 is calculated in instruction 1. To get the correct value of x_{14} to 2NOPs are added.

Reason: (NOP_2)
x_{13} needed in instruction 5 is loaded in instruction 3. 1 NOP is added to make a difference b/w these instructions as 2.

Reason:

Reason: (NOP_3)
x_{13} needed in instruction 6 is calculated in instruction 5. Two NOPs are added to make a difference of 2 to get the correct value.

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

```
IF  ID  EXE  MEM  WB        add x14, x12, x11
    IF  ID   EXE  MEM  WB    add x15, x14, x12
```

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

```
IF  ID  EXE  MEM  WB        ld  x13, 8(x13)
    IF  ID   EXE  MEM  WB   ld  x12, 0(x14)
        IF   ID   EXE  MEM  WB   and x13, x15, x13
```

x_{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.

```
IF  ID  EXE  MEM  WB
    IF  ID   EXE  MEM  WB
```

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