CS & IT ENGINEERING

Computer Organization Architecture

Pipeline Processing

DPP- 02

Discussion Notes

By- Devvrat sir

[MCQ]

Ans[A]

#Q. Consider the given set of instructions, which are having instruction format as follows :

Opcode, Destination operand, Source1, Source2

1. ADD R2, R1, R0 — R2 ← R1 + R0 — I_1
2. MUL R4, R3, R2 — R4 ← R3 * R2 — I_2
3. SUB R6, R5, R4 — R6 ← R5 – R4 — I_3
4. ADD R6, R7, R8 — R6 ← R1 + R8 — I_4
5. MUL R4, R9, R2 — R4 ← R9 * R2 — I_5
6. SUB R9, R3, R4 — R9 ← R3 – R4 — I_6

The number of RAW, WAW and WAR dependencies are respectively ?

4 2 2

A. 4, 2, 2
B. 5, 2, 2
C. 4, 1, 2
D. 5, 1, 2

RAW

④ I_1-I_2 (R2) I_1-I_5

I_2-I_3 (R4) I_5-I_6

4 2 2

WAW

② I_2-I_5 (R4)

I_3-I_4 (R6)

WAR - ②

[NAT]

#Q. Consider the given set of instructions, which are having instruction format as follows :

Opcode, Destination operand, Source1, Source2

1.	ADD R2, R1, R0	R2 ← R1+ R0
2.	MUL R4, R3, R2	R4 ← R3 * R2
3.	SUB R6, R5, R4	R6 ← R5 - R4
4.	ADD R6, R7, R8	R6 ← R7 + R8
5.	MUL R7, R1, R2	R7 ← R1 * R2
6.	SUB R1, R3, R4	R1 ← R3 – R4

$I_1 - I_2 (R_2)$

$I_3 - I_3 (R4)$

K=5

RAW - 2

The given instructions are executed in a 5 segment instruction pipeline which has segments as: Instruction Fetch, Instruction Decode, Operand Fetch, Execution and Write Back.

$k = 5$

$n = 6$

$(k + n - 1) = 5 + 6 - 1 = 10$ cycles

The speed up is calculated as follows :

$$\text{Speed up} = \frac{\text{Number of cycles needed without operand forwarding}}{\text{Number of cycles needed with operand forwarding}}$$

$10 + 2 \times 2 = 14$

$= 10$ cycles.

$= \frac{14}{10} = 1.4$ Ans.

Speed of the pipeline (correct up to 1 decimal place) for execution of above instructions is 1.4 ?

3 5

IF, ID, OF, EX, WB

$I_1 \Rightarrow R_2 \leftarrow R_1 + R_3$

$I_2 \quad R_6 \leftarrow R_2 + R_4$

# of stall cycle $= 5 - 3 = 2$ stall cycles

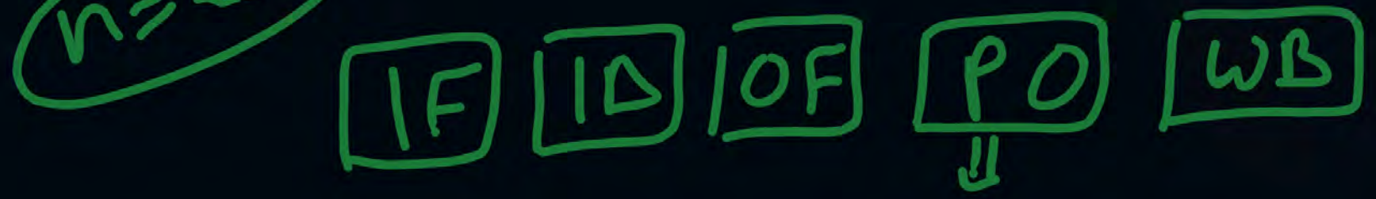
[NAT]

#Q. The instruction pipeline of a RISC processor has the following stages: Instruction Fetch (IF), Instruction Decode (ID), Operand Fetch (OF), Perform Operation (PO) and Writeback (WB), The IF, ID, OF and WB stages take 1 clock cycle each for every instruction. Consider a sequence of 200 instructions. In the PO stage, 40 instructions take 4 clock cycles each, 65 instructions take 2 clock cycles each, and the remaining instructions take 1 clock cycle each. Assume that there are 20 instructions which cause 2 stalls each due to data hazards and there are no control hazards in program.

The number of clock cycles required for completion of execution of the sequence of instruction is 429 ?

$K=5$

$n=200$



# of stalls cycles due to 40 $\ln x$ ⑤ $= 40(4-1) = 120$ stall cycles.

" " " 65 " $= 65(2-1) = 65$ "

$(K+n-1)$

# of stalls due to data Hazard for 20 $\ln x$ ⑤ $= 20 \times 2 = 40$

total stalls $= 120 + 65 + 40$

$= 225$ stall cycle.

total cycles $= (K+n-1) + 225$

$= (5+200-1)+225$

$= 204 + 225 = 429$

[NAT]

$t_p = \max(6, 5, 8, 6, 7) = 8$ n Sec.

#Q. Consider a 5-stage instruction pipeline, where the stages take delays of 6 nanoseconds, 5 nanoseconds, 8 nanoseconds, 6 nanoseconds and 7 nanoseconds respectively. When an application is executing on this 5-stage pipeline, consider 20% of the instructions incur 3 pipeline stall cycles. The speed up (correct up to 2 decimal places) of the pipeline as compared to its corresponding non-pipeline system is 2.50 ?

$S = \frac{32}{12.8} = 2.50$

Execution time in Non-Pipeline = 6 + 5 + 8 + 6 + 7 = 32 n sec.

Exec^n time in Pipeline = 1.6 × 8 = 12.8

Prog 100% → 20% → 2 stalls

#y Stalls/Ins^n = 0.2 × 3 = 0.6

CPI in Pipeline = 1 + 0.6 = 1.6

[NAT]

#Q. Consider a non-pipelined processor operating at 10 GHz. It takes 4 clock cycles to complete an instruction. The same processor is upgraded to pipelined processor with 5 stages but clock rate is reduced to 8GHz. A program is executed on these processors which has 8% load/store instructions, 12% branch instructions and remaining ALU instructions. The memory access instructions cause 1 clock cycles each if there is no any cache miss but cause 40 cycles with cache miss. The 40% of all branch instructions cause 2 stalls each. There is no any stall associated with ALU instructions. Assume that cache has 94% hit. The speed up (round of up to 2 decimal place) achieved by pipeline over the non-pipeline processor for this program is 2.35 ?

$$S = \frac{0.4}{0.1704} = 2.347 \Rightarrow 2.35$$

Cycle time in non pipeline $= \frac{1}{10GHz} = 0.1$ nsec

$tp = \frac{1}{8GHz}$

$= 0.125$ nsec

Execution time for an Inx^n in NP $= 0.1 \times 4 = 0.4$ nsec

8% load/store Inx^{15}

Exec time in Pipeline

$= 1.3632 \times 0.125$

$= 0.1704$

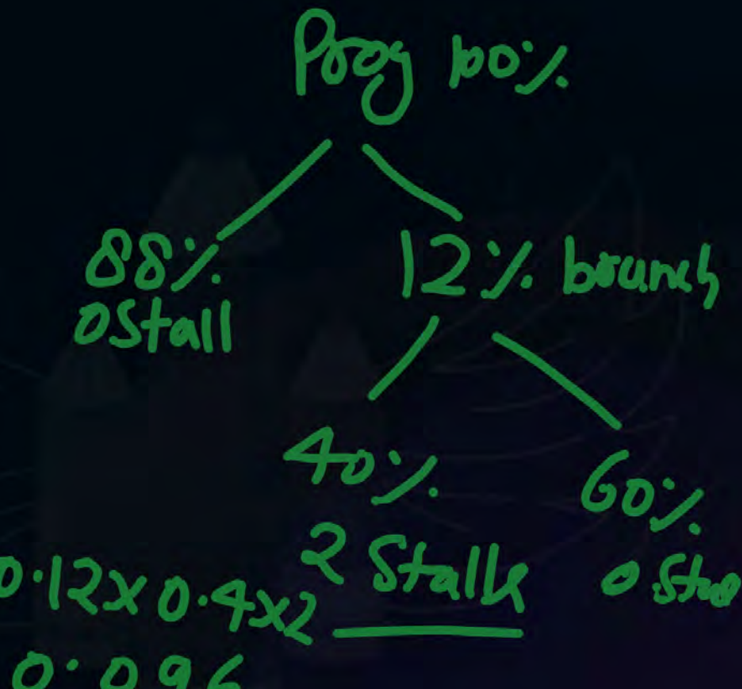
$0.08\,(0.94 \times 1 + 0.06 \times 40)$

$0.08\,(0.94 + 2.4)$

$= 0.2672$

(PI $= 1 + 0.2672 + 0.096$

$= 1.3632$



$= 0.12 \times 0.4 \times 2$

$= 0.096$

[MCQ]

Ans [B]

(6, 5, 8, 6, 7) 6+5+8+6+7 = 32 nsec

NOT true

#Q. Which of the following statements is/are false regarding pipelining ?

n=1

tp = 8 nsec

(k+n-1)8 = (5+1-1)8 = 5×8 = 40 nsec

A T For single input non-pipeline can perform better than equivalent pipeline system

B F Pipeline can be fruitful when multiple different processing is applied over multiple inputs

same

C T Pipeline with single segment cannot provide parallel processing

D T Non-pipeline system does not require intermediate registers/buffer for synchronization

[MCQ]

Ans [C]

#Q. Consider a 6 segment pipeline in which all segments take 1 cycle for each instructions except the execution phase. Execution phase takes 3 cycle for multiply instruction, 6 cycles for division instruction and 1 cycle for addition and 1 cycle subtraction instruction. Suppose a code segment is executed on this pipeline in which total 8 instructions are executed and total 29 cycles needed for execution. Which of the following option(s) is/are correct regarding number of instructions of each type ?

MUL

A

ADD	SUB	MUL	DIV
2	2	2	2

B

ADD	SUB	MUL	DIV
1	1	3	3

C

ADD	SUB	MUL	DIV
1	2	3	2

D

ADD	SUB	MUL	DIV
2	1	2	3

$K=6 \quad n=8 = K+n-1 = (6+8-1) = 13$

#of stalls cycles due to MUL = 3-1 = 2 stall cycles.

" " " " " DIV = 6-1 = 5 " "

" " ADD = 1-1 = 0

SUB = 1-1 = 0

(A) total stalls = 2x2 + 2x5 = 14 + 13 = 27 cycles.

(B) " " = 3x2 + 3x5 = 6 + 15 = 21 + 13 = 34 cycles.

(C) " " = 3x2 + 2x5 = 16 + 13 = 29 cycles

(D) " " = 2x2 + 3x5 = 19 + 13 = 32 cycles.

THANK - YOU