CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Pipeline Processing

Lecture No.- 06

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Instruction Pipeline

Topic Data Hazard Classification

Topic RAW, WAW, WAR

Topics to be Covered

Topic CPI in Instruction Pipeline

Topic Classical RISC Pipeline

Topic : Instruction Pipeline

IF:	Instruction Fetch
ID:	Instruction Decode & Address Calculation
OF:	Operand Fetch
EX:	Execution
WB:	Write Back

[NAT]

#Q. Consider a 5-stage instruction pipeline, where stages take delays 5ns, 4ns, 6ns, 4ns and 5ns respectively. The pipeline is used to execute a program in which 25% instructions cause 4 stalls due to hazard. The average instruction execution time in the pipeline is 12 ns?

$$CPI_{avg} = 1 + 0.25 * 4$$
$$= 2$$

$$t_p = max(5, 4, 6, 4, 5)$$
$$= 6ns$$

$$A.I.E.T. = 2 * 6$$
$$= 12ns$$

[NAT] GATE-PYQ

Ans = 1.4

#Q. A CPU has a five-stage pipeline and runs at 1 GHz frequency. Instruction fetch happens in the first stage of the pipeline. A conditional branch instruction computes the target address and evaluates the condition in the third stage of the pipeline. The processor stops fetching new instructions following a conditional branch until the branch outcome is known. A program executes 10^9 instructions out of which 20% are conditional branches. If each instruction takes one cycle to complete on average, the total execution time of the program is ________ seconds?

cycle time $= \frac{1}{1GHz}$

$t_p = 1ns$

stalls due to branch $inst^{ns} = 3 - 1 = 2$

$CPI_{avg} = 1 + 0.2 * 2$

$= 1.4$

Avg. inst^n execut^n time $= 1.4 * 1\,ns$

$= 1.4\,ns$

Prog. (10^9 inst^{ns}) execution time $= 10^9 * 1.4\,ns$

$= 1.4$ seconds

[NAT] GATE- PYQ

#Q. An instruction pipeline has five stages where each stage takes 2 nanoseconds, and all instructions use all five stages. Branch instructions are not overlapped, i.e., the instruction after the branch is not fetched till the branch instruction is completed. Under ideal conditions.

1. Calculate the average instruction execution time assuming that 20% of all instruction executed are branch instructions. Ignore the fact that some branch instructions may be conditional. 3.6 ns

2. If a branch instruction is a conditional branch instruction, the branch need not be taken. If the branch is not taken, the following instructions can be overlapped. When 80% of all branch instructions are conditional branch instructions, and 50% of the conditional branch instructions are such that the branch is taken, calculate the average instruction execution time? 2.96 ns

Sol^n

5-stages

each stage ⇒ 2ns

stalls due to branch = $\boxed{5-1=4}$

(because next inst^n is fetched after branch inst^{ns} completed in all 5-staged.)

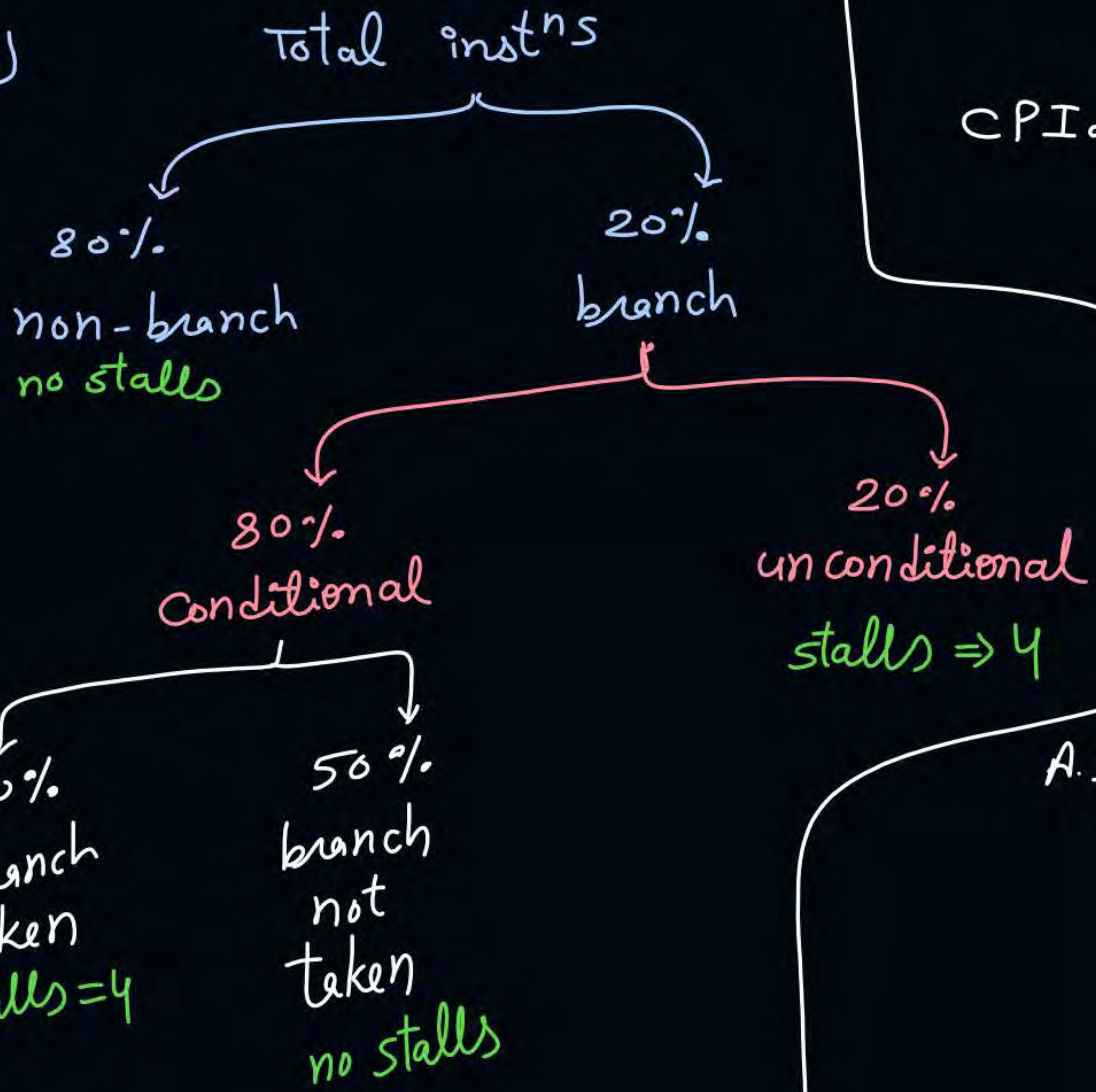
$t_p = 2ns$

1.

$$CPI_{avg} = 1 + 0.2 * 4$$
$$= 1.8$$

$$A.I.E.T. = 1.8 * 2$$
$$= 3.6 \text{ ns}$$

2.)



$$CPI_{avg} = 1 + (0.2 * 0.2 * 4) + (0.2 * 0.8 * 0.5 * 4)$$

$$= 1.48$$

$$A.I.E.T. = 1.48 * 2ns$$

$$= 2.96 \text{ ns}$$

[NAT] GATE- PYQ

#Q. Consider a non-pipelined processor operating at 2.5 GHz. It takes 5 clock cycles to complete an instruction. You are going to make a 5- stage pipeline out of this processor. Overheads associated with pipelining force you to operate the pipelined processor at 2 GHz. In a given program, assume that 30% are memory instructions, 60% are ALU instructions and the rest are branch instructions. 5% of the memory instructions cause stalls of 50 clock cycles each due to cache misses and 50% of the branch instructions cause stalls of 2 cycles each. Assume that there are no stalls associated with the execution of ALU instructions. For this program, the speedup achieved by the pipelined processor over the non-pipelined processor (round off to 2 decimal places) is 2.16 .

Non-pipeline :-

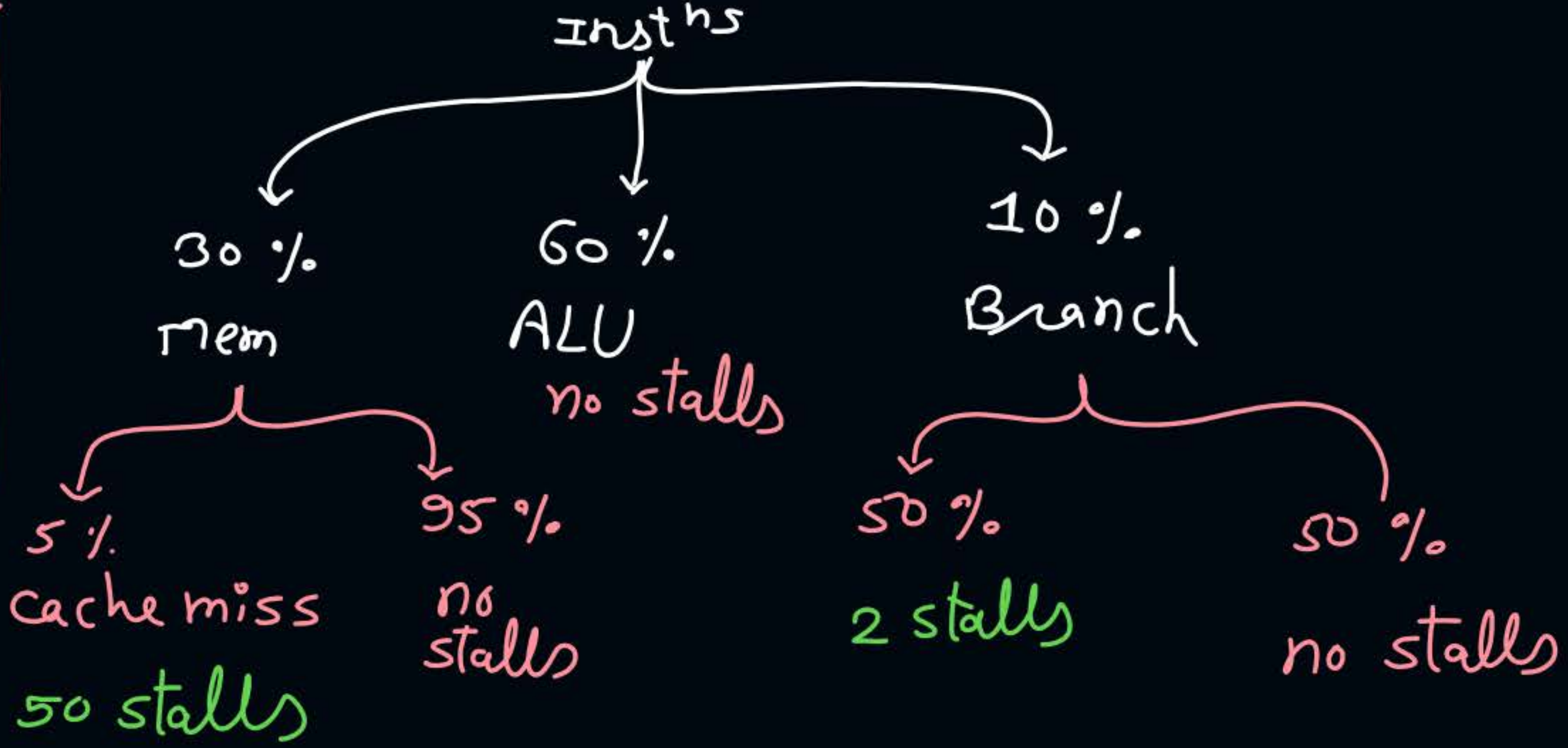
$$t_n = 5 * \frac{1}{2.5\,GHz}$$

$$= 2\,ns$$

Pipeline :-

$$k = 5$$

$$t_p = \frac{1}{2\,GHz} = 0.5\,ns$$



$$CPI = 1 + (0.3 * 0.05 * 50) + (0.1 * 0.5 * 2)$$

$$= 1.85$$

$$\text{speed up} = \frac{t_n}{CPI * t_p} = \frac{2\,ns}{1.85 * 0.5\,ns}$$

$$= 2.162$$

$$= 2.16$$

[NAT] GATE - PYQ

Ans = 1.43

#Q. A processor X_1 operating at 2 GHz has a standard 5-stage RISC instruction pipeline having a base CPI (cycles per instruction) of one without any pipeline hazards. For a given program P that has 30% branch instructions, control hazards incur 2 cycles stall for every branch. A new version of the processor X_2 operating at same clock frequency has an additional branch predictor unit (BPU) that completely eliminates stalls for correctly predicted branches. There is neither any savings nor any additional stalls for wrong predictions. There are no structural hazards and data hazards for X_1 and X_2. If the BPU has a prediction accuracy of 80%, the speed up (rounded off to two decimal places) obtained by X_2 over X_1 in executing P is

X_1 :-

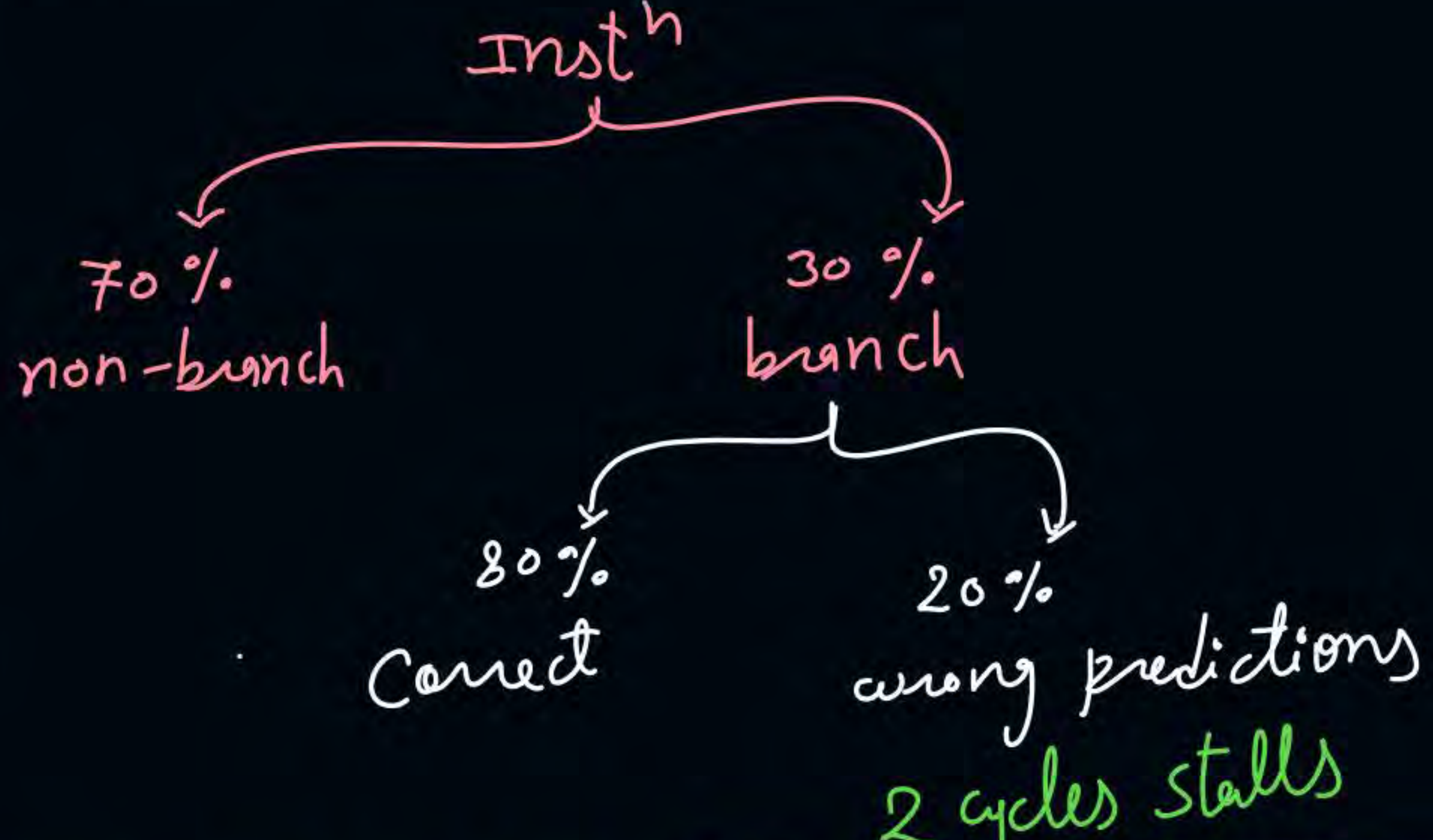
cycle time $= \frac{1}{2GHz} = 0.5\,ns$

$CPI_{avg} = 1 + 0.3 * 2 = 1.6$

Avg inst^n execut^n time $= 1.6 * 0.5\,ns$

X_2 :-

cycle time $= \frac{1}{2GHz} = 0.5\,ns$



$CPI = 1 + 0.3 * 0.2 * 2$
$= 1.12$

$A.I.E.T. = 1.12 * 0.5\,ns$

$$\text{speed up} = \frac{1.6 * \cancel{0.5}\ \text{ns}}{1.12 * \cancel{0.5}}$$

$$= 1.428$$

$$= \underline{\underline{1.43}} \quad \text{Ans.}$$

Topic : Classic RISC Pipeline

1. IF → Inst^n fetch
2. ID → decode & register Read
3. EX → ALU operation
4. MEM → memory access (Read/write)
5. WB → write back to register

RISC ⇒ Reg. - based architecture

↳ operands taken from Reg. only for ALU operation

Topic : Classic RISC Pipeline for Computation (ALU type)

1. IF → Inst^n fetch
2. ID → Decode of inst^n & operand fetch from registers
3. EX → operation in ALU
4. MEM → nothing
5. WB → write back result to register

stalls due to data hazard

= WB stage no. − OF stage no.

= 5 − 2 = 3

Topic : Classic RISC Pipeline for Load

Reg. ← Mem

1. IF → fetch of inst^n
2. ID → decode of inst^n
3. EX → Nothing
4. MEM → Memory Read
5. WB → write back to reg.

Topic : Classic RISC Pipeline for Store

Memory ← Reg.

1. IF → Inst^n fetch
2. ID → Inst^n decode & register read
3. EX → Nothing
4. MEM → Memory write
5. WB → Nothing

Topic : Classic RISC Pipeline for Branch

1. IF → Inst^n fetch

2. ID → decode, target address calculation, condition check, PC updation

3. EX

4. MEM

5. WB

3–5 → Nothing

stalls due to branch inst^n = 2 − 1

= 1

[NAT] GATE-2021 PYQ Ans = 1.88

#Q. Consider a pipelined processor with 5 stages, Instruction Fetch (IF), Instruction Decode (ID), Execute (EX), Memory Access (MEM), and Write Back (WB). Each stage of the pipeline, except the EX-stage, takes one cycle. Assume that the ID stage merely decodes the instruction and the register read is performed in the EX-stage. The EX-stage takes one cycle for ADD instruction and two cycles for MUL instruction. Ignore pipeline register latencies.

Consider the following sequence of 8 instructions:

ADD, MUL, ADD, MUL, ADD, MUL, ADD, MUL

Assume that every MUL instruction is data-dependent on the ADD instruction just before it and every ADD instruction (except the first ADD) is data-dependent on the MUL instruction just before it. The Speedup is defined as follows:

$$\textit{Speed up} = \frac{\textit{Execution time without operand forwarding}}{\textit{Execution time with operand forwarding}} = \frac{30}{16} = 1.875 = 1.88$$

The Speedup achieved in executing the given instruction sequence on the pipelined processor (rounded to 2 decimal places) is ______.

IF — 1

ID — 1

EX —

ADD	MUL
1	2

stall ⇒ 1 for each

MEM — 1

WB — 1

---

with operand forwarding :-

no stalls due to data hazard

no. of cycles w/o hazard = $k+n-1 = 5+8-1 = 12$

no. of stalls due to structural hazard = $4*1 = 4$

Total = 16

w/o operand forw. :-

w/o hazard = 5 + 8 − 1 = 12

stalls for structural hazard = 4

stalls for data hazard = 7 * (5 − 3) = 14

Total = 30

no. of stalls due to each data hazard = WB stage no − operand fetch stage no.

= 5 − 3

= 2

2 mins Summary

Topic CPI in Instruction Pipeline

Topic Classical RISC Pipeline

3 May
2 hour } Doubts

Happy Learning

THANK - YOU