

# COMPUTER SCIENCE

## Computer Organization and Architecture

### Instruction Pipelining

Lecture\_02



Vijay Agarwal sir

A graphic of a construction barrier made of orange and white striped panels, with two yellow bollards on top, positioned on the left side of the slide.

**TOPICS  
TO BE  
COVERED**

**o1**

## **Pipelining Concepts**

Pipeline Concept

Pipeline Design.

Execution Time Calculation

Performance Evaluation of Pipeline Processor

Throughput

Uniform & Non Uniform Delay Pipeline.

# Pipelining Strategy

Similar to the use of  
An assembly line in a  
Manufacturing plant

To apply this concept  
To instruction  
Execution we must  
Recognize that an  
Instruction has a  
Number of stages



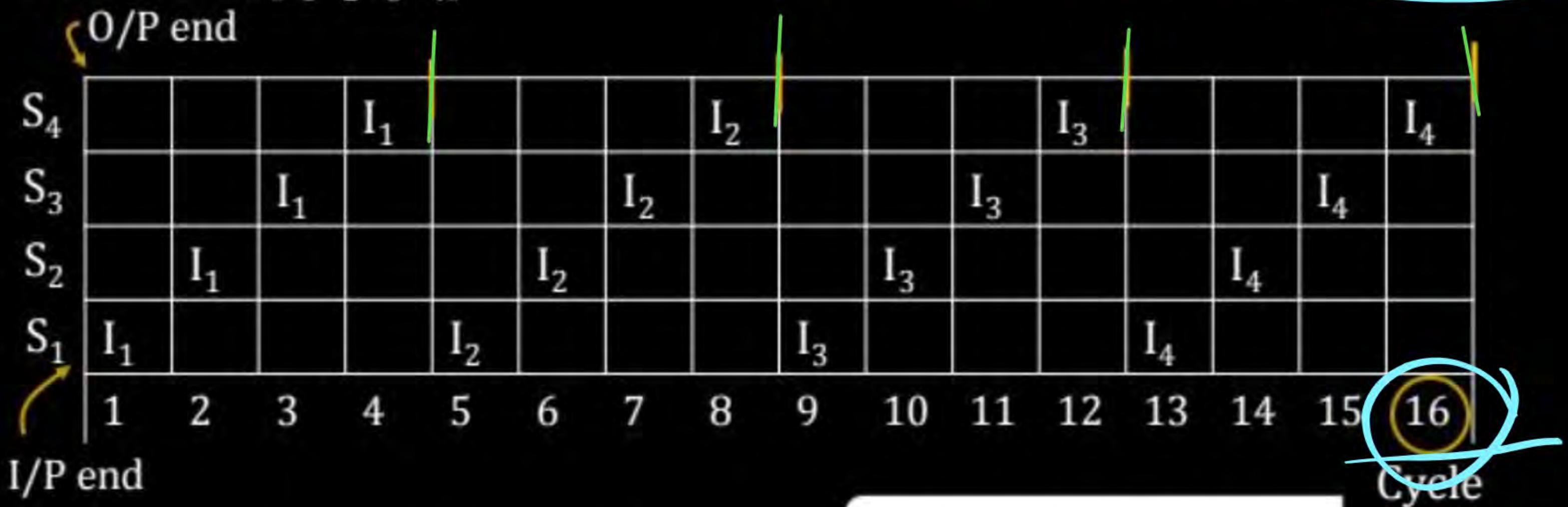
New inputs are  
Accepted at one end  
Before previously  
Accepted inputs  
Appear as output at  
The other end

- Pipelining is a mechanism which is used to improve the performance of the system in which task (Instruction) are executed in overlapping manner.
- Pipelining is a decomposition technique that means the problem is divided into sub problem & Assign the sub problem to the pipes then operate the pipe under the same clock.

Let us consider 4 segment pipeline used to execute 4 instruction the execution sequence as:

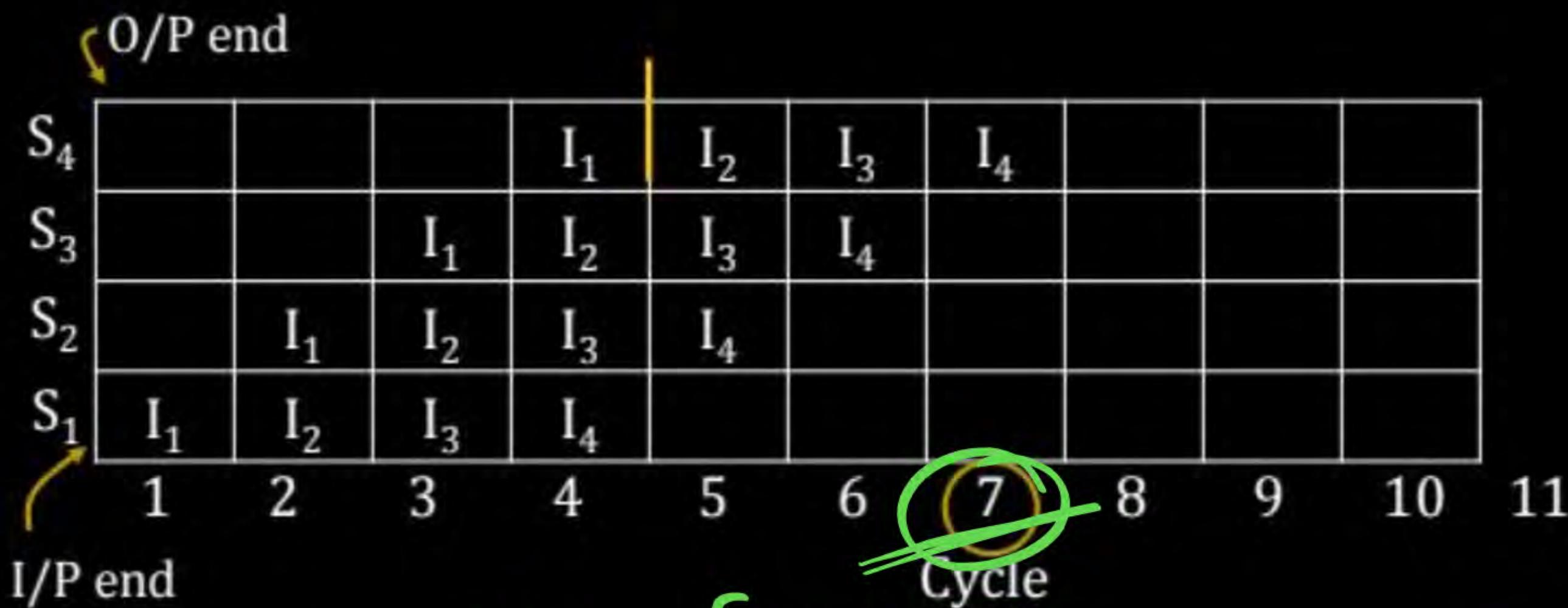
Segment/stages =  $[S_1 \ S_2 \ S_3 \ S_4]$

Instruction:  $[I_1 \ I_2 \ I_3 \ I_4]$



$n = 4, t_n = 4$ , Non pipeline

**Non-PIPELINE**



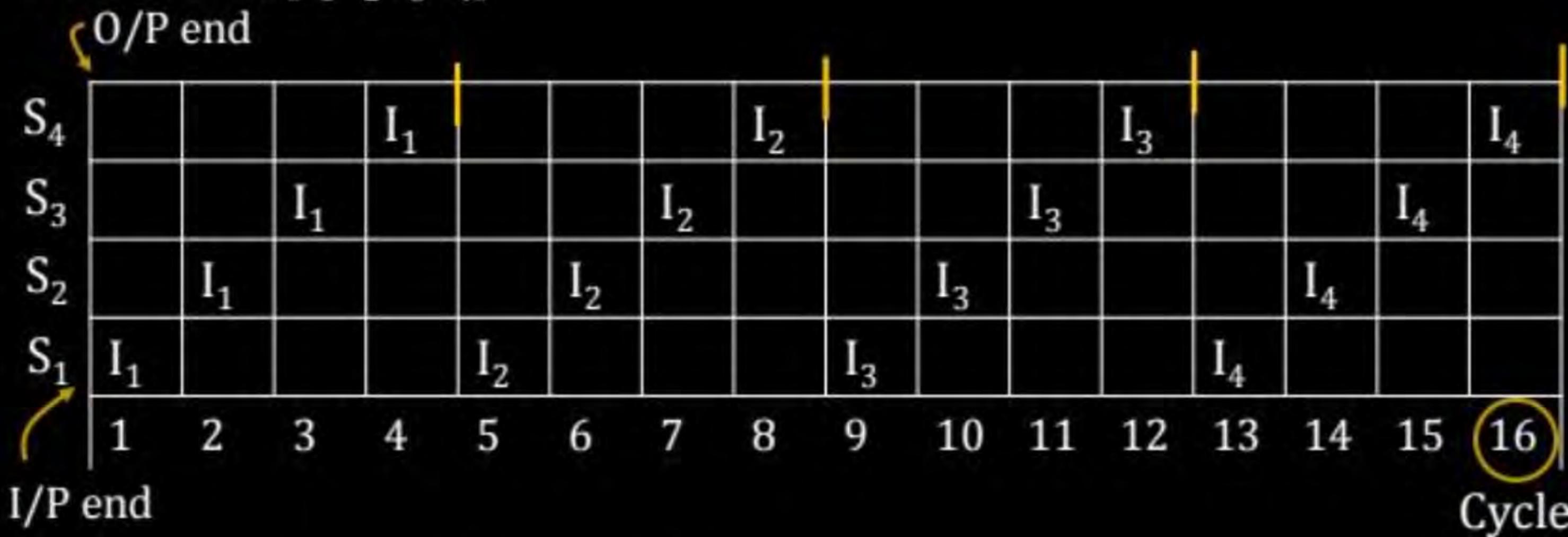
$$\begin{aligned}
 ET_{PIPE} &= [k + (n-1)] t_p \\
 &\Rightarrow [4 + (4-1)] \text{cycle} = 7 \text{ cycle}
 \end{aligned}$$

PIPELINE

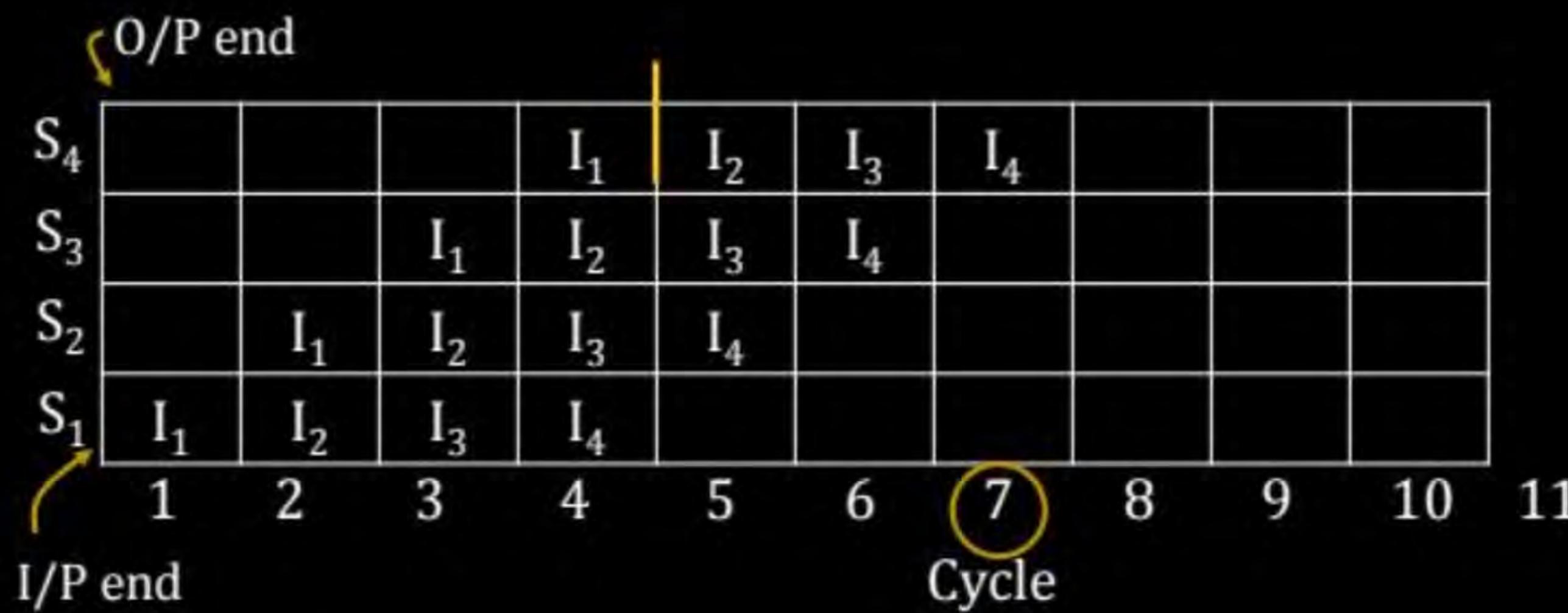
Let us consider 4 segment pipeline used to execute 4 instruction the execution sequence as: PIPELINING

Segment/stages =  $[S_1 \ S_2 \ S_3 \ S_4]$

Instruction:  $[I_1 \ I_2 \ I_3 \ I_4]$



$$n = 4, t_n = 4, \text{ Non pipeline}$$

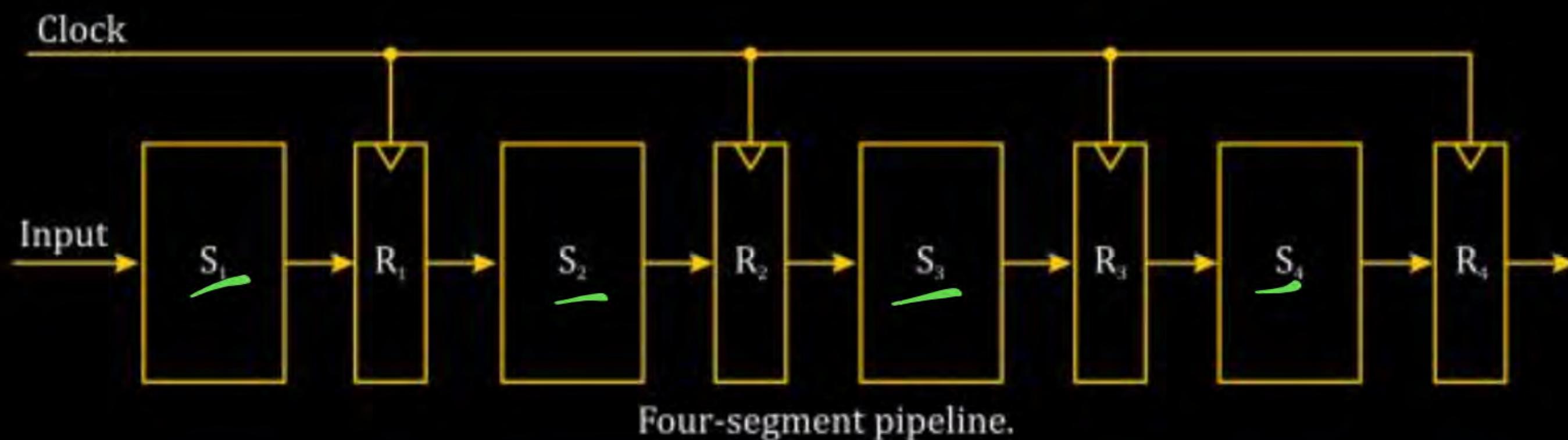


PIPELINE

 $k = 4$  $n = 4$

# PIPELINE Design

if end  
or end



Buffer | Pipeline Register | Interface Register |  
Latch .

Now consider the case where a  $k$ -segment pipeline with a clock cycle time  $t_p$ , is used to execute  $n$  tasks.

The first task  $T_1$  requires a time equal to  $kt_p$ , to complete its operation since there are  $k$  segments in the pipe.

The remaining  $n - 1$  tasks emerge from the pipe at the rate of one task per clock cycle and they will be completed after a time equal to  $(n - 1)t_p$ .

Therefore, to complete  $n$  tasks using a  $k$ -segment pipeline requires  $k + (n - 1)$  clock cycles.

## Space-time diagram for pipeline.

Segment:

	1	2	3	4	5	6	7	8	9	→ Clock Cycles
1	$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$				
2		$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$			
3			$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$		
4				$T_1$	$T_2$	$T_3$	$T_4$	$T_5$	$T_6$	

For example, the diagram of Fig. 9-4 shows four segments and six tasks. The time required to complete all the operations is  $4 + (6 - 1) = 9$  clock cycles, as indicated in the diagram.

So, to complete  $n$  tasks using a  $k$ -segment pipeline is:

$$\text{ET pipe} = k + (n - 1) \text{ clock cycles}$$

$$ET_{PIPE} = [k + (n - 1)] t_p.$$

So, to complete n tasks in Non-pipeline is:

$$\text{ET Non-pipeline} = n * tn$$

$$S = \frac{nt_n}{(k+n-1)t_p}$$

As the number of tasks increases,  $n$  becomes much larger than  $k - 1$ , and  $k + n - 1$  approaches the value of  $n$ . Under this condition, the speedup becomes

$n$  is very large  
~~ok~~

$n$  is Not given

@

Ideal case.

$$S = \frac{t_n}{t_p}$$

## Additional Stages

- ❑ Fetch Instruction (FI)
  - ❖ Read the next expected Instruction into a buffer.
- ❑ Decode Instruction (DI)
  - ❖ Determine the opcode and the operand specifiers.
- ❑ Calculate operands(CO)
  - ❖ Calculate the effective address of each source operand.
  - ❖ This may involve displacement, register indirect or other forms of address calculations.
- ❑ Fetch Operands(FO)
  - ❖ Fetch each operand from memory.
  - ❖ Operands in register need not be fetched.
- ❑ Executed Instruction(EI)
  - ❖ Perform the indicated operation and store the result, if any, in the specified destination operand location
- ❑ Write Operand(WO)
  - ❖ Store the result in memory

	Time →													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Instruction 1	FI	DI	CO	FO	EI	WO								
Instruction 2		FI	DI	CO	FO	EI	WO							
Instruction 3			FI	DI	CO	FO	EI	WO						
Instruction 4				FI	DI	CO	FO	EI	WO					
Instruction 5					FI	DI	CO	FO	EI	WO				
Instruction 6						FI	DI	CO	FO	EI	WO			
Instruction 7							FI	DI	CO	FO	EI	WO		
Instruction 8								FI	DI	CO	FO	EI	WO	
Instruction 9									FI	DI	CO	FO	EI	WO

Timing Diagram for Instruction pipeline operation

$$ET_{PIPELINE} = [k + (n-1)] \cdot tp$$

$$ET_{NONPIPE} = n \cdot tn$$

$S = \frac{\text{Performance of Pipeline}}{\text{Performance of Nonpipeline}}$

(Performance Gain)

$$S = \frac{n \cdot tn}{[k + (n-1)] \cdot tp}$$

$$S = \frac{tn}{tp}$$

(When  $n$  is very large  
 $n$  is not given  
Ideal Case)

$k$ : No. of Stage (Segment)

$n$ : No. of Instn | Task

$tp$ : Each Stage Delay in Pipeline

$$\text{Throughput} = \frac{n}{[k + (n-1)] \cdot tp}$$

$$\text{Throughput} = \frac{1}{tp}$$

(When  $n$  is very large  
 $n$  is not given  
Ideal Case)

$$\eta = \frac{S}{k}$$

### Uniform Delay Pipeline

$tp = \text{Stage Delay}$

If Buffer Delay is given

$$tp = \text{Stage Delay} + \text{Buffer Delay}$$

### Non Uniform Delay Pipeline

$$tp = \max(\text{Stage Delay})$$

If Buffer Delay is given

$$tp = \max(\text{Stage Delay} + \text{Buffer Delay})$$

$tn$ : Each Instruction ET in Non pipeline

$\eta$ : efficiency

$S$ : Speed Up Factor (Performance Gain).

## Perfectly Balance & Ideal Case.

$$t_n = k * t_p$$

$n$  is Not given  
@ very large.

$$S = \frac{t_n}{t_p}$$

$$S = \frac{k * t_p}{t_p}$$

$S = K$  Maximum Speed up Factor  
No. of Stages ( Perfectly balance  
Uniform Delay + Ideal Case )

Perfectly & Balanced      Ideal Case

$$S_1$$

$$2^{n_3}$$

$$S_2$$

$$2^{n_3}$$

$$S_3$$

$$2^{n_3}$$

$$S_4$$

$$2^{n_3}$$

$k=4$  Stage.  
 $t_p = 2 \text{ nsec}$

$$t_n = 2 + 2 + 2 + 2 = 8 \text{ nsec}$$

$$\begin{aligned} t_n &= k * t_p = 4 * 2 \\ t_n &= 8 \text{ nsec} \end{aligned}$$

Q. 1 Consider on instruction pipeline which has speed up factor 20 while operate with 80% efficiency. What could be the number of stages in the pipeline?

(SOL)

$$\eta = \frac{S}{K}$$

$$80\% = \frac{20}{K}$$

$$K = \frac{20}{80} \times 100$$

$$K = 25$$

Ans

Q.

## COA 2017: CS

## NIC PYQ SERIES



A pipeline is having speed up factor as 10 and operating with efficiency of 80%. What will be the number of stages in the pipeline?

A 10

B 13

C 8

D None

$$\eta = \frac{S}{K}$$

$$80\% = \frac{10}{K}$$

$$K = \frac{10}{80} \times 100$$

$$K = 12.5$$

$$K = 13 \text{ Avg}$$

Q. 2 4 segment pipeline have the respective stage delay of 10ns, 15ns, 20ns and 30ns. What is the efficiency of the pipeline when very large number of task are executed?

$$k = 4 \quad [10, 15, 20, 30]$$

$$S = \frac{tn}{tp} = \frac{75}{30} = 2.5$$

$$tn = 10 + 15 + 20 + 30$$

$$tn = 75 \text{ nsec}$$

$$\eta = \frac{S}{k} = \frac{2.5}{4} = 0.625$$

$$tp = \max(\text{Stage Delay}) \\ \Rightarrow \max(10, 15, 20, 30)$$

$$\boxed{tp = 30 \text{ nsec}}$$

$\Rightarrow 62.5\% \underline{\text{Avg}}$

Q. 3 A 4 segment instruction pipeline has the respective stage delay of 8ns, 11ns 20ns, 2ns respectively. The interface register are used between the stages have a delay of 2ns. What is the approx. speed up factor when very large number of task are pipelined?

$$k=4, \text{ Stage Delay} = (8, 11, 20, 2) \text{ & Buffer Delay} = 2 \text{ nsec}$$

$$t_n = 8 + 11 + 20 + 2 = 41 \text{ nsec}$$

$$t_p = \max \left( \frac{\text{Stage Delay}}{\text{Delay}} + \frac{\text{Buffer Delay}}{\text{Delay}} \right) \Rightarrow 20 + 2 = 22 \text{ nsec}$$

$$t_p = 22 \text{ nsec}$$

$$S = \frac{t_n}{t_p} = \frac{41}{22} = 1.86 \text{ Avg}$$

Q. 4

Consider 4 stage pipeline with the respective stage delay of 20ns, 80ns, 50ns and 10ns. In the enhancement process of the pipeline largest stage is split into 2 equal stage delay.

- (i) What is speed up between new and old design?
- (ii) What is the clock frequency of new pipeline?

Soln

4 stages

Stage Delay = [20ns, 80ns, 50ns, 10ns]

$$t_p = 80\text{ nsec}$$

New Design K=5 Stage

Largest stage

80

40 40

Stage Delay : [20nsec, 40nsec, 40nsec, 50nsec, 10nsec]

$$t_{p\text{new}} = 50\text{ nsec}$$

$$S = \frac{\text{Performance of New}}{\text{Performance of OLD}}$$

$$\Rightarrow \frac{ET_{\text{Old}}}{ET_{\text{New}}}$$

$$S = \frac{80}{50}$$

$$S = 1.6 \quad \underline{\text{Ans}}$$

$$P \propto \frac{1}{ET}$$

(Spin) Frequency  $\propto \frac{1}{\text{cycle time}}$ .

$$\text{Frequency} = \frac{1}{50 \times 10^{-9}}$$

$$\Rightarrow \frac{1}{50} \times 10^9 \Rightarrow \frac{1000 \times 10^6}{50}$$

$$= 20 \text{ MHz} \quad \underline{\text{Ans}}$$

Q. 5

The stage delays in 5 stage pipeline are 900, 600, 550, 450 and 400 nanoseconds. The largest stage delay is replaced with a functionally equivalent design involving two stages with respective stage delay 440 and 460 nanoseconds. What is the throughput increase of the pipeline?

- (a) 25%
- (b) 33.3%
- (c) 50%
- (d) Does not change

P  
W

### Old Design

$K=5$

Stage Delay = (900, 600, 550, 450, 400)

$$t_P = \max(\text{Stage Delay})$$

$$t_{P\text{old}} = 900 \text{ nsec}$$

$$ET = [k + (n-1)] t_P \Rightarrow [5 + (n-1)] \times 900 \text{ nsec}$$

$$\text{Throughput} = \frac{n}{[k + (n-1)] t_P} = \frac{n}{[5 + (n-1)] \times 900 \text{ nsec}}$$

$$\text{Throughput} = \frac{1}{900} \text{ nsec}$$

$$\text{Throughput}_{\text{OLD}} = \frac{1}{900 \text{ nsec}}$$

### New Design

$K=6$

Stage Delays = (440, 460, 600, 550, 450, 400)

$$t_P = \max(\text{Stage Delay})$$

$$t_{P\text{new}} = 600 \text{ nsec}$$

$$ET_{\text{PIPE}} = [6 + (n-1)] 600 \text{ nsec}$$

$$\text{Throughput} = \frac{n}{[k + (n-1)] t_P} = \frac{n}{[6 + (n-1)] \times 600}$$

$$\text{Throughput} = \frac{1}{600 \text{ nsec}}$$

### OLD Design

$$K=5$$

Stage Delay = (900, 600, 550, 450, 400)

$$t_p = \max(\text{Stage Delay})$$

$$t_{p\text{old}} = 900 \text{ nsec}$$

$$\perp \text{Instn in OLD Pipeline} = 900 \times 10^{-9} \text{ sec}$$

$$\text{In 1 sec \#Instn Executed} = \frac{1}{900 \times 10^{-9}}$$

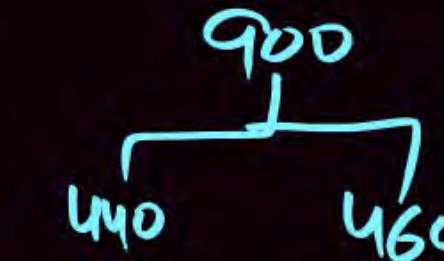
$$\text{Throughput} = \frac{1}{t_p}$$

$$\text{Throughput}_{\text{old}} = \frac{1}{900 \text{ nsec}}$$

### New Design

$$K=6$$

Stage Delays = (440, 460, 600, 550, 450, 400)



$$t_p = \max(\text{Stage Delay})$$

$$t_{p\text{new}} = 600 \text{ nsec}$$

$$\text{Throughput}_{\text{new}} = \frac{1}{600}$$

$$\% \text{ of Increment in Throughput} = \frac{\text{New} - \text{OLD}}{\text{OLD}} \quad (\text{Profit Loss})$$

$$\Rightarrow \frac{\frac{1}{600} - \frac{1}{900}}{\frac{1}{900}} \Rightarrow \frac{\frac{1}{6} - \frac{1}{9}}{\frac{1}{9}}$$

$$\Rightarrow \frac{\frac{9-6}{54}}{9} \Rightarrow \frac{3}{54} \times \frac{9}{1} = \frac{27}{54} = \frac{1}{2}$$

$\Rightarrow 50\% \underline{\text{Avg}}$

Q. 1

A 4-stage pipeline has the stage delays as 150, 120, 160 and 140 nanoseconds respectively. Registers that are used between the stages have a delay of 5 nanoseconds each. Assuming constant clocking rate, the total time taken to process 1000 data items on this pipeline will be

- A 120.4 microseconds
- B 160.5 microseconds
- C 165.5 microseconds
- D 590.0 microseconds

$$N = 1000$$

$$k = 4,$$

[GATE-2004: 2 Marks]

$$t_P = \max(\text{Stage Delay} + \text{Buffer Delay})$$

$$\boxed{t_P = 165 \text{ nsec}}$$

$$160 + 5 = 165 \text{ nsec}$$

$$E_{PIPE} = [k + (n-1)]t_P$$

$$\Rightarrow [4 + (1000-1)] \times 165 \times 10^{-9}$$

$$\Rightarrow 1003 \times 165 \times 10^{-9}$$

$$\Rightarrow 165.5 \times 10^{-6}$$

Ans (C).

Q. 2

We have two designs D1 and D2 for a synchronous pipeline processor. D1 has 5 pipeline stages with execution times of 3 nsec, 2 nsec, 4 nsec, 2 nsec and 3 nsec while the design D2 has 8 pipeline stages each with 2 nsec execution time.

How much time can be saved using design D2 over design D1 for executing 100 instructions? [GATE-2005: 2 Marks]

A 214 nseconds

B 202 nseconds

C 86 nseconds

D -200 nsecond

Ans (B)

### Design D<sub>1</sub>

$$k=5, \text{ Stage Delays} = (3, 2, 4, 2, 3)$$

$$t_P = \max(3, 2, 4, 2, 3)$$

$$\boxed{t_P = 4 \text{nsec}} \quad \boxed{k=5} \quad n=100$$

$$ET_{D_1} = [k + (n-1)] t_P$$

$$\Rightarrow (5 + (100 - 1)) \times 4$$

$$\boxed{ET_{D_1} = 416 \text{nsec}}$$

Time Saved =  $416 - 214 = 202 \text{nsec}$  Avg

### Design D<sub>2</sub>

$$k=8 \quad t_P=2 \text{nsec} \quad n=100$$

$$ET_{D_2} = [k + (n-1)] t_P$$

$$\Rightarrow (8 + (100 - 1)) \times 2$$

$$\boxed{ET_{D_2} = 214 \text{nsec}}$$

Q. 3

A nonpipelined single cycle processor operating at 100 MHz is converted into a synchronous pipelined processor with five stages requiring 2.5 nsec, 1.5 nsec, 2 nsec, 1.5 nsec and 2.5 nsec, respectively. The delay of the latches is 0.5 nsec. The speedup of the pipeline processor for a large number of instructions is [GATE-2008: 2 Marks]

- A 4.5
- B 4.0
- C 3.33
- D 3.0

Ans [C]

## Non pipeline

$$\text{Cycle time} = \frac{1}{\text{Freq}} = \frac{1}{100\text{MHz}}$$

$$\Rightarrow \frac{1}{100 \times 10^6} \text{ sec}$$

$$\Rightarrow 10^{-8} \times \frac{10}{10} = 10 \times 10^{-9}$$

$t_n = 10 \text{ nsec}$

$t_n = 2.5 + 1.5 + 2 + 1.5 + 2.5 = 10 \text{ nsec}$

## Pipeline $k=5$

$$\text{Stage Delays} = (2.5, 1.5, 2, 1.5, 2.5)$$

$$\text{Bubble Delay} = 0.5$$

$$t_p = \max \left( \frac{\text{Stage Delay}}{\text{Delay}} + \frac{\text{Bubble Delay}}{\text{Delay}} \right) = 2.5 + 0.5$$

$t_p = 3 \text{ nsec}$

$$S = \frac{t_n}{t_p} = \frac{10}{3} = 3.33 \underline{\text{Ans}}$$

$$3+3=6$$

$$3 \times 3 = 9$$

$$2+2=4$$

$$2 \times 2 = 4$$

$$+ = \times$$

Q. 4

Consider an instruction pipeline with four stages (S1, S2, S3 and S4) each with combinational circuit only. The pipeline registers are required between each stage and at the end of the last stage. Delays for the stages and for the pipeline registers are as given in the figure

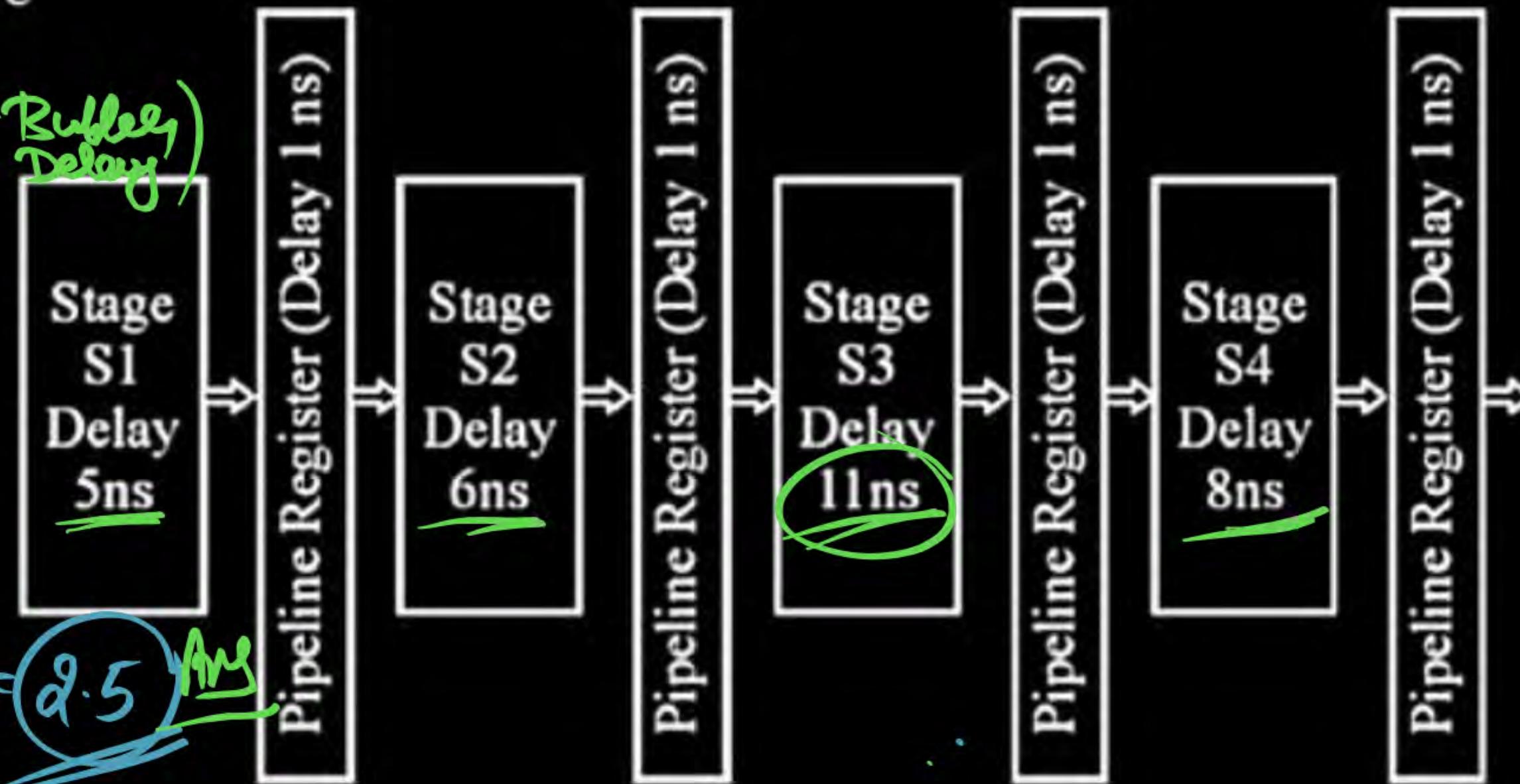
$$t_p = \max(\text{Stage Delay} + \text{Bubble Delay})$$

$$\boxed{t_p = 12 \text{ nsec}}$$

$$t_n = 5 + 6 + 11 + 8$$

$$\boxed{t_n = 30 \text{ nsec}}$$

$$S = \frac{t_n}{t_p} = \frac{30}{12} = 2.5 \text{ ms}$$



What is the approximate speed up of the pipeline in steady state under ideal conditions when compared to the corresponding non-pipeline implementation?

[GATE-2011: 2 Marks]

- A 4.0
- B 2.5
- C 1.1
- D 3.0

Ans (B)

(Uniform Delay)

Perfectly  
Balanced

& Ideal Case

$$S = k (\# \text{ Stages})$$

$$S=4$$

But Only When Ideal &  
Perfectly Balance [Uniform  
Delay].

A five-stage pipeline has stage delays of 150, 120, 150, 160 and 140 nanoseconds. The registers that are used between the pipeline stages have a delay of 5 nanoseconds each.

The total time to execute 100 independent instructions on this pipeline, assuming there are no pipeline stalls, is 17160 nanoseconds.

$$k=5$$

$$\text{Stage Delay} = (150, 120, 150, \cancel{160}, \cancel{140}) \quad [\text{GATE-2021(Set-1)-CS: 2M}]$$

$$\text{Buffer Delay} = 5 \text{ nsec}$$

$$t_p = \max(\text{Stage Delay} + \text{Buffer Delay}) = 160 + 5 = 165 \text{ nsec}$$

$$104N65$$

$$ET_{PIPE} = [k + (n-1)] t_p$$

$$\Rightarrow [5 + (100-1)] \times 165$$

$$= 104 \times 165 = 17,160 \text{ nsec}$$

$$\underline{\text{Ans}}(17160)$$

Consider a 3-stage pipelined processor having a delay of 10 ns (nanoseconds), 20 ns, and 14 ns, for the first, second, and the third stages, respectively. Assume that there is no other delay and the processor does not suffer from any pipeline hazards. Also assume that one instruction is fetched every cycle.

The total execution time for executing 100 instructions on this processor is

2040 ns. Avg

[GATE-2023-CS: 1M]

$$k=3, \quad \text{Stage Delay} = (10, 20, 14 \text{ nsec}). \quad n=100.$$

$$t_p = \max(10, 20, 14) \Rightarrow t_p = 20 \text{nsec}.$$

$$\begin{aligned} ET_{PIPE} &= [k + (n-1)] t_p \\ &= [3 + (100-1)] \times 20 \\ &= 102 \times 20 = 2040 \text{ nsec Avg.} \end{aligned}$$

Instruction execution in a processor is divided into 5 stages. Instruction Fetch (IF), Instruction Decode (ID), Operand Fetch (OF), Execute (EX) and Write Back (WB). These stages take 5, 4, 20, 10 and 3 nanoseconds (ns) respectively. A pipelined implementation of the processor requires buffering between each pair of consecutive stages with a delay of 2 ns.

Two pipelined implementations of the processor are contemplated:

- (I) a naive pipeline implementation (NP) with 5 stages and
- (II) an efficient pipeline (EP) where the OF stage is divided into stages OF1 and OF2 with execution times of 12 ns and 8 ns respectively.

The speedup (correct to two decimal places) achieved by EP over NP in executing 20 independent instructions with no hazards is 1.508 Avg

[GATE-2017(Set-1)-CS: 2M]

## Naive Pipeline (NP)

5 Stage

$k=5$  IF, ID, UF, EX, WB

Stage Delay: 5 4 20 10 3

Bubble Delay = 2 nsec

$$t_p = \max \left( \frac{\text{Stage Delay}}{\text{Delay}} + \frac{\text{Bubble Delay}}{\text{Delay}} \right)$$

$$t_p \Rightarrow 22 \text{ nsec}$$

$$\begin{aligned}
 ET_{NP} &= [k + (n-1)] t_p \\
 &= [5 + (20-1)] \times 22 \\
 &= 24 \times 22 \text{ nsec}
 \end{aligned}$$

$$ET_{NP} = 528 \text{ nsec}$$

$$n = 20$$

$k=6$

6 Stage

	IF, ID	OF1	OF2	EX	WB
Stage Delay	5	4	12	8	10

## Efficient Pipeline (EP)

$$ET_{EP} = [k + (n-1)]t_p$$

$$\text{Bubble Delay} = \frac{2 \text{ nsec}}{1}$$

$$\Rightarrow [6 + (20 - 1) \times 14]$$

$$t_p = \max \left( \frac{\text{Stage Delay} + \text{Bubble Delay}}{12+2} \right)$$

$$t_p \Rightarrow 14 \text{ nsec}$$

$$ET_{EP} = 350 \text{ nsec}$$

$$n=20$$

$$\text{Speed up} = \frac{\text{EP Performance}}{\text{NP Performance}} \Rightarrow \frac{Y_{\text{EP}}}{Y_{\text{NP}}}$$

$$\Rightarrow \frac{ET_{NP}}{ET_{EP}}$$

$$= \frac{528}{350} = \boxed{1.508}$$

Consider the following processors (ns stands for nanoseconds). Assume that the pipeline registers have zero latency.

P1: Four -stage pipeline with stage latencies 1ns, 2 ns, 2 ns, 1 ns.

P2: Four-stage pipeline with stage latencies 1ns, 1.5 ns, 1.5 ns, 1.5 ns.

P3: Five-stage pipeline with stage latencies 0.5 ns, 1 ns, 1 ns, 0.6 ns. 1 ns.

P4: Five-stage pipeline with stage latencies 0.5 ns, 0.5 ns, 1 ns, 1 ns, 1.1 ns.

Which processor has the highest peak clock frequency?

[GATE-2014(Set-3)-CS: 1M]

A P1

B P2

C P3

D P4

Processor P<sub>1</sub>  $\Rightarrow$   $t_{P_1} = \max(1, 2, 2, 1) \Rightarrow t_{P_1} = 2 \text{ nsec}$

Processor P<sub>2</sub>  $\Rightarrow$   $t_{P_2} = \max(1, 1.5, 1.5, 1.5) \Rightarrow t_{P_2} = 1.5 \text{ nsec}$

Processor P<sub>3</sub>  $\Rightarrow$   $t_{P_3} = \max(0.5, 1, 1, 0.6, 1) \Rightarrow t_{P_3} = 1 \text{ nsec}$

Processor P<sub>4</sub>  $\Rightarrow$   $t_{P_4} = \max(0.5, 0.5, 1, 1, 1.1) \Rightarrow t_{P_4} = 1.1 \text{ nsec}$

Frequency  $\propto \frac{1}{\text{Time}}$

$t_{P_3}$  has the lowest time, so

P<sub>3</sub> has Highest Clock Frequency.

The stage delays in a 4-stage pipeline are 800, 500, 400 and 300 picoseconds. The first stage (with delay 800 picoseconds) is replaced with a functionally equivalent design involving two stages with respective delays 600 and 350 picoseconds. The throughput increase of the pipeline is 33.33 percent.

A<sub>3</sub>(33.33)

### OLD Design

$$t_{P_{OLD}} = \max(800, 500, 400, 300)$$

$$t_{P_{OLD}} = 800 \text{ ps}$$

$$\text{Throughput}_{OLD} = \frac{1}{800}$$

[GATE-2016(Set-1)-CS: 2M]

### New Design

$$t_{P_{NEW}} = \max(600, 350, 500, 400, 300)$$

$$t_{P_{NEW}} = 600 \text{ ps}$$

$$\text{Throughput}_{new} = \frac{1}{600}$$

% Increment =  $\frac{\text{New} - \text{OLD}}{\text{OLD}}$   
in throughput

$$= \frac{\frac{1}{600} - \frac{1}{800}}{\frac{1}{800}} \Rightarrow \frac{\frac{1}{6} - \frac{1}{8}}{\frac{1}{8}}$$
$$\Rightarrow \frac{8-6}{48} \times \frac{8}{1} \Rightarrow \frac{2}{48} \times 8 = \frac{16}{48} = \frac{1}{3}$$

= 33.33% Ans

Consider a non-pipelined processor with a clock rate of 2.5 gigahertz and average cycles per instruction of four. The same processor is upgraded to a pipelined processor with five stages; but due to the internal pipeline delay, the clock speed is reduced to 2 gigahertz. Assume that there are no stalls in the pipeline. The speed up achieved in this pipelined processor is \_\_\_\_.

[GATE-2015(Set-1)-CS: 2M]

**MCQ** Q. 11

Consider a 4-stage pipeline processor. We want to execute a loop:  
`For(i=1;i<=1000;i++){ I1, I2, I3, I4}` where the time taken (in ns) by instruction I1 to I4 for stages S1, S2, S3, S4 is shown below:

	S1	S2	S3	S4
I1	1	2	1	2
I2	2	1	2	1
I3	1	1	2	1
I4	2	1	2	1

The Output of I1 for i=2 will be available after ?

[GATE-2004-CS: 2M]

A 11ns

B 12ns

C 13ns

D 28ns

**MCQ** Q. 12

Consider a 4-stage pipeline processor. The number of cycles needed by the four instructions I1, I2, I3, I4 in stages S1, S2, S3, S4 is shown below:

	S1	S2	S3	S4
I1	2	1	1	1
I2	1	3	2	2
I3	2	1	1	3
I4	1	2	2	2

What is the number of cycles needed to execute the following loop?

for (i = 1 to 2) {I1; I2; I3; I4;}

[GATE-2009-CS: 2M]

A 16

B 23

C 28

D 30

Consider a 3 GHz (gigahertz) processor with a three-stage pipeline and stage latencies  $\tau_1$ ,  $\tau_2$ , and  $\tau_3$  such that  $\tau_1 = 3\tau_2/4 = 2\tau_3$ . If the longest pipeline stage is split into two pipeline stages of equal latency, the new frequency is \_\_\_\_\_ GHz, ignoring delays in the pipeline registers.

[GATE-2016(Set-2)-CS: 2M]

**NAT Q. 14**

Consider two processors  $P_1$  and  $P_2$  executing the same instructions set. Assume that under identical conditions, for the same input, a program running on  $P_2$  takes 25% less time but incurs 20% more CPI (clock cycles per instruction) as compared to the program running on  $P_1$ . If the clock frequency of  $P_1$  is 1GHz, then the clock frequency of  $P_2$  (in GHz) is \_\_\_\_\_.

[GATE-2014(Set-1)-CS: 2M]

Consider the following processor design characteristics:

- I. Register-to-register arithmetic operations only.
- II. Fixed-length instruction format.
- III. Hardwired control unit.

Which of the characteristics above are used in the design of a RISC processor?

[GATE-2018-CS: 1M]

A I and II only

B II and III only

C I and III only

D I, II and III

Q. 16.

Consider a machine with 40 MHz processor which has run a benchmark program. The executed program consists of 100,000 instruction executions, with the following instruction mix and clock cycle count. What will be the effective CPI, MIPS rate, and execution time.

Instruction Type	Instruction Count	Cycles/ Instructions
Integer arithmetic	45000	1
Data Transfer	32000	2
Floating point	15000	2
Control transfer	8000	2

-  A CPI:3.55; MIPS: 30; Execution time:1.87 ms
-  B CPI:1.55; MIPS: 25.8; Execution time:3.87 ms
-  C CPI:5.60; MIPS: 45.8; Execution time:2.87 ms
-  D CPI:2.55; MIPS: 35.8; Execution time:4.87 ms

Which of the following is not a form of main memory ?

- A Instruction cache
- B Instruction register
- C Instruction opcode
- D Translation look-aside buffer

Q.18

COA 2017: CS

NIC PYQ SERIES

P  
W

In a 10-bit computer instruction format, the size of address field is 3-bits. The computer uses expanding OP code technique and has 4 two-address instructions and 16 one-address instructions. The number of zero address instructions it can support is

A 256

C 640

B 356

D 756

## A micro programmed control unit

- (A) is faster than a hardwired unit
- (B) Facilitates easy implementation of a new instruction
- (C) is useful when small programs are to be run
- (D) All of the above

**THANK  
YOU!**

