** # "Pipeline" :- (4 marks)

Function :-

* Pipelining uses the decomposition technique that means problem statement is divided into a Independent Subproblems, Assign them to a independent Hardware later connect the Hardware in a pipeline sequence.

Def :-

"Accepting the New Inputs at one end "before" the previously accepted Input is appeared as an output at the other end."

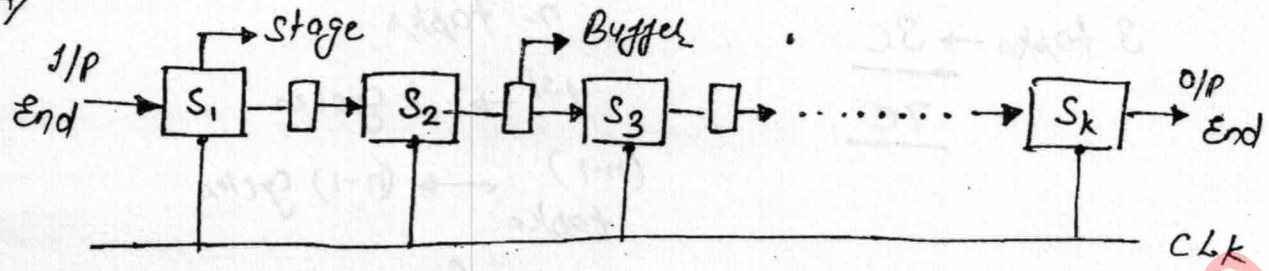
* Definition states that, Insert the New Input into a pipeline before completion of a old Input so New Input is overlapped with a old Input called as overlapping Execution.

* In A Non-pipeline System, New Input Inserted "After" the completion of old Input called as Non-overlapping Execution.

* Overlapping Execution sequence in the pipeline is described using Space-time diagram i.e.

in the Pipeline design become most important.

⇒

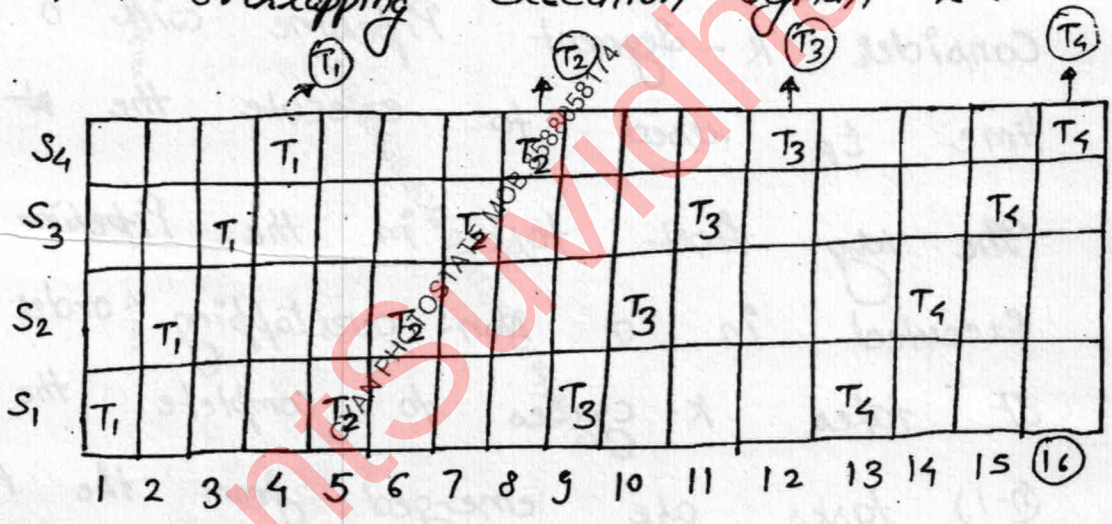


Eg: Consider 4-Stage (S_1, S_2, S_3, S_4)

Pipeline used to execute 4-tasks

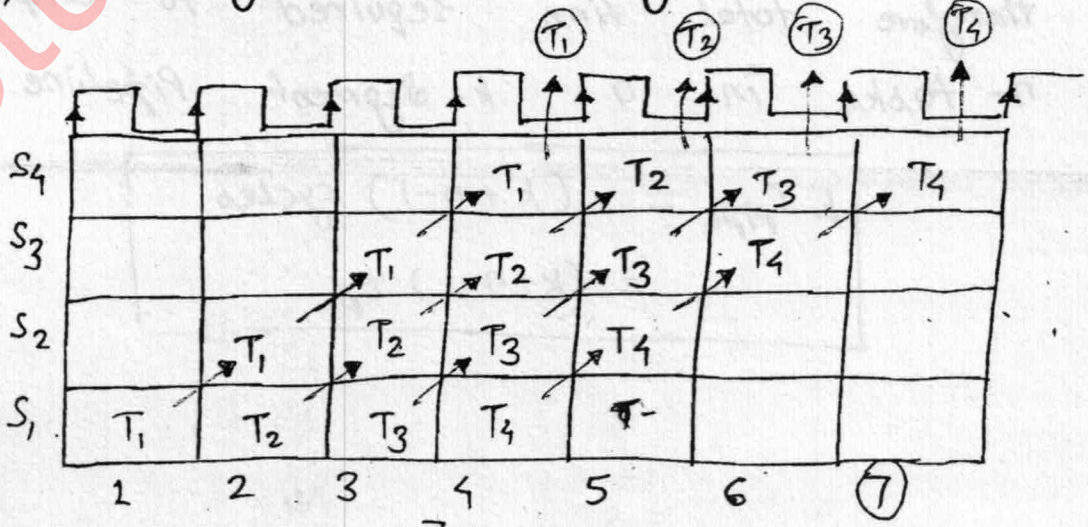
(T_1, T_2, T_3, T_4)

(a) Non-overlapping Execution segment is :



16 cycles

(b) overlapping Execution segment is :



1st → 4c

3 tasks → 3c

7c

k - stage

n - tasks

1st → k cycles

(n-1) tasks → (n-1) cycles

cycles to execute 'n' tasks : ⇒ (k + n-1) cycles

k: stages

Performance Analysis :-

★ Consider k - segment Pipeline with a cycle time t_p used to execute the n tasks

★ The very first task in the Pipeline is Executed in a Non-overlapping order so, it takes k - cycles to complete. the Remaining (n-1) tasks are emerged from the Pipe at the Rate of 1 task / cycle. So (n-1) cycles are required to complete the (n-1) task. Therefore total time required to complete the n - tasks in a k - segment Pipeline is :

$$ET_{pipe} = (k + n-1) \text{ cycles}$$
$$= (k+n-1)\, t_p$$

★ Consider Non-Pipeline System, used to execute n-tasks in which each task takes t_n time to complete therefore total time required to complete the n-tasks in a Non-Pipeline system is :

$$E.T._{nonpipeline} = n \cdot t_n$$

★ Performance gain of a Pipeline is :

$$S = \frac{\text{Performance}_{pipe}}{\text{Performance}_{Nonpipe}}$$

$$S = \frac{\frac{1}{ET_{pipe}}}{\frac{1}{E.T._{Nonpipe}}}$$

$$S = \frac{E.T._{Nonpipe}}{E.T._{pipe}}$$

$$S = \frac{n \cdot t_n}{(k + n - 1)\, t_p} \quad ; \quad n \leftarrow \text{Limited}$$

★ When No. of tasks are increases then n becomes much larger than (k-1) so (k + n-1) value will be approach to n. In this condition :

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

$$\boxed{S = \frac{t_n}{t_p}} \; ; \quad n \leftarrow \infty$$

$\boxed{t_n : \text{one task E.T. nonpipeline}}$

$t_n = k$ cycles → # of stages

$\therefore t_n = k \cdot t_p$

So $S_{max} = \frac{k \cdot t_p}{t_p}$

$S_{max} = k \; ; \quad \eta = 100\%$ → efficiency.

* when the Pipeline Stages are perfectly balanced then one task execution time in the Non-Pipeline is also equal to a Number of Stages in the pipeline i.e.

$$\boxed{t_n = k \text{ cycles} = k \cdot t_p}$$

In this Condition :

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

* $\boxed{S = k}$; $\eta = 100\%$

Stages in Pipeline [Pipeline depth].

★ when the System is operating with 100%. efficiency then maximum Speedup possible i.e. also equal to pipeline depth.

100% η — S_{max}

? η — S

$$\eta_{pipe} = \frac{S}{S_{max}} = \frac{S}{k}$$

$$\#\text{ of stage} * \text{efficiency} = \text{Speedup}$$

$$\text{Throughput}_{pipe} = \#\text{tasks Processed} \Big/ \text{Total time taken to Process the tasks}$$

$$TP_{pipe} = n / (k+n-1)\, t_p$$

Types of Pipeline :-

1.) Linear Pipeline :-

★ This pipeline always contain the 'feed-forward connection', so inputs are inserted into a pipeline with a latency of '1' therefore Reservation table is not required to process the inputs so all the Linear pipelines are Synchronous pipelines.

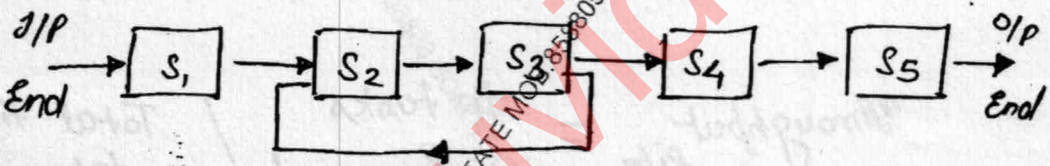
2.) Non-Linear Pipeline :-

* This Pipeline contain both forword and backword connections so Reservation table is used in the Pipeline to process the inputs.

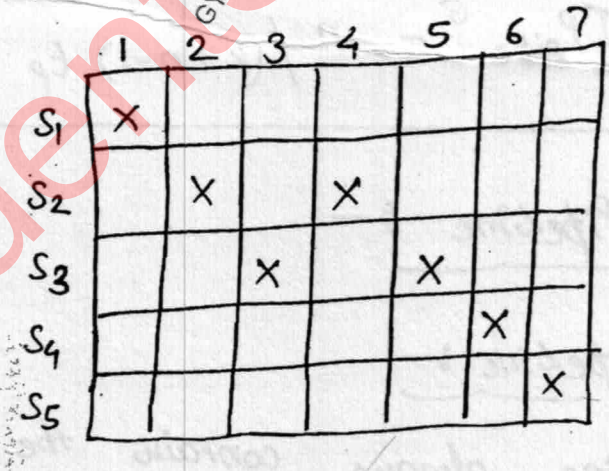
* In this pipeline Latency value will be varies, calculated based on the Reservation table.

* All the Non-Linear pipelines are Asynchronous pipelines.

Eg:

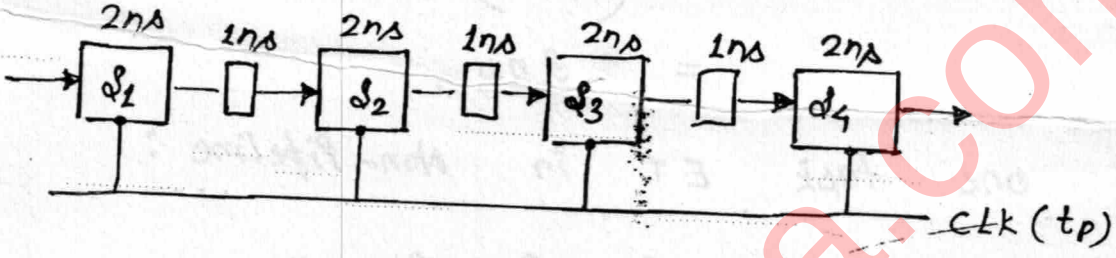


Reservation table :-



3.) Uniform Delay Pipeline Design :-

In this Pipeline all the stages are taking same amount of the time to complete the assigned operation.



a.) $\boxed{\text{Cycle time } (t_p) = \text{Stage delay}}$

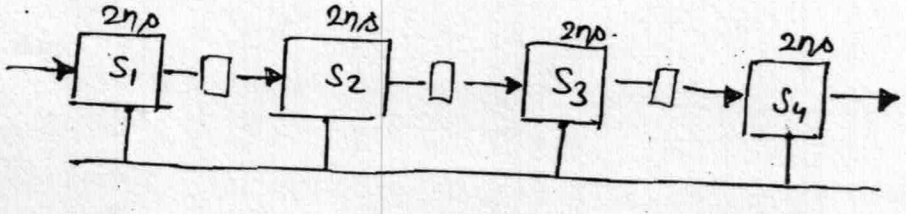
b.) If Buffer delay is included then

$$\boxed{t_p = \text{Stage delay} + \text{Buffer delay}}$$

c) If Skew time / Setup time overhead is included then $\boxed{t_p = t_p + \text{overhead}}$

★ Note :-

When the Pipeline stages are perfectly balanced (uniform delay) then one task execution time in the pipeline is equal to a one time execution time in the Non-Pipeline.



(a) one task E.T. in Pipeline :

$$\left\{ \begin{array}{l} k = 4 \\ n = 1 \\ t_p = 2ns \end{array} \right\}$$

$$E.T_{pipe} = (k + n - 1) * t_p$$

$$= (4 + 1 - 1) * 2ns$$

$$= \underline{8ns}.$$

(b) one task E.T. in Non-Pipeline :

$$t_n = s_1 + s_2 + s_3 + s_4$$

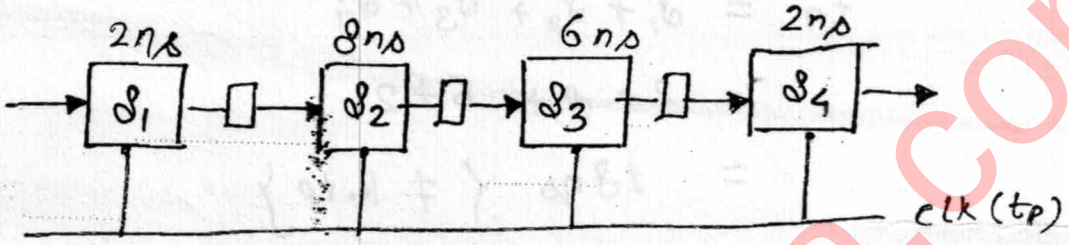
$$= 2 + 2 + 2 + 2$$

$$= 8ns \quad \{ = k\, t_p \}$$

$$\left\{ \begin{array}{l} s = \dfrac{t_n}{t_p} \; ; \; n \leftarrow \infty \\ s = \dfrac{8}{2} \\ s = 4 \quad \{ = k \} \\ \eta = \dfrac{4}{4} = \dfrac{s}{k} \\ \eta = 1 \; (= 100\%) \end{array} \right\}$$

4.) Non- uniform Delay Pipeline Design :-

* In this Pipeline different Stages consumes different amount of the time to complete the assigned operation.



a) $\boxed{\text{cycle time } (t_p) = \max(\text{Stage Delay})}$

b.) If Buffer delay is included then $\boxed{t_p = \max\left(\begin{matrix}\text{Stage} \\ \text{Delay}\end{matrix} + \begin{matrix}\text{Buffer} \\ \text{Delay}\end{matrix}\right)}$

** Note :- when the Pipeline Stages are not perfectly balanced (Non- uniform delay) then one task Execution time in the Pipeline is always greater than one task execution time in the 'Non- pipeline'.

a.) one - task E.T. in Pipeline :

$k = 4$
$n = 1$

$t_p = \max(2, 8, 6, 2)$
$= 8$ ns.

$$E.T._{pipe} = (k+n-1)\,t_p$$
$$= (4+1-1) * 8\,ns$$
$$= 32\,ns.$$

b.) one task E.T. in Non-Pipeline (t_n)

$$t_n = \delta_1 + \delta_2 + \delta_3 + \delta_4$$
$$= 2 + 8 + 6 + 2$$
$$= 18\,ns \quad \{ \neq k.t_p \}$$

$$\left\{ \begin{array}{l} S = \dfrac{t_n}{t_p} \; ; \; n \leftarrow \infty \\ \\ S = \dfrac{18}{8} = 2.25 \; (\neq k) \\ \\ \eta = \dfrac{S}{k} = \dfrac{2.25}{4} \\ \\ \eta = 56.25\% \; (\neq 100\%) \end{array} \right\}$$

* Note :- Let t_1 is one task execution time in Pipeline, t_2 is one task execution time in Non-pipeline. Relationship between the T_1 & T_2 is

$$\boxed{T_1 \geq T_2}$$

Que:1 :- Consider an instruction Pipeline which has speedup factor 13 while operating with 70% efficiency. what could be the # of stages in the pipeline ?

$S = 13$

$\eta = 70\%$

$k = ?$

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

$k = \frac{S}{\eta} = \frac{13}{0.7} = \lceil 18.57 \rceil \approx 19$

$\boxed{k = 19}$

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

★

Que ②:- Consider 2 Pipelines A and B where Pipeline A is having 8 stages of uniform delay of 500 MHz clock (clk). Pipeline B is having 5 stages with a respective delays of 2ns, 4ns, 6ns, 3ns and 1ns. How much time is saved when the 100 tasks are pipelined using A. instead of B.

A Pipeline :-

$k = 8$

$\text{delay} = \frac{500\,MHz}{} = \frac{1}{500 * 10^{-6}} = \frac{10^6}{500} = \frac{1000}{500 \times 10^9}$

$= 2ns$

$(k + n - 1) * = (8 + 99) = 107 * t_p$

$= 214\,ns$

B Pipeline :-

$k = 5 \quad n = 100$

Cycle time : $\max(2, 4, 6, 3, 1) = 6ns$

Eq B = $(k + n - 1) t_p = (5 + 100 - 1)\, t_p$

$= 104 * 6 = 624ns$

Time save = $(ET_B - ET_A)$

= $(624 - 214$

= 410 ns

Que:- Consider 4 stage pipeline with a respective stage delay of (20, 40, 50 and 30) ns. what is the performance gain and efficiency of a Pipeline ?

:- Cycle time = max (20, 40, 50, 30)

= 50 ns

k = 4

E.T. Pipeline = $(k + n - 1) * t_p$

= ~~(4) * 50~~

→ n not given so n = ∞ (Let)

So $S = \frac{t_n}{t_p}$; $n \to \infty$

$S = \frac{40 + 20 + 50 + 30}{50}$

$S = \frac{140}{50} = 2.8$

$\boxed{S = 2.8}$

Efficiency $(\eta) = \frac{S}{k} = \frac{2.8}{4} = 70\%$

** Que:- Consider 4-stage pipeline with a respective delays of 30 ns, 60 ns, 40 ns, & 20 ns. Interface Register used between the stages have a delay of 5 ns. What is the Performance gain of a pipeline when the very large number of tasks are executed.

Soln:-

$k = 4$

$tp = \max(30, 60, 40, 20) + 5$

$= \max(35, 65, 45, 25)$

$= 60 + 5 = 65$ ns

"last Stage No Buffer"

$n = \infty$

So

$S = \frac{tn}{tp} = \frac{30+60+40+20}{65}$ So Not adding '5' Buffer delay

$= \frac{150}{65}$ ns $= 2.30$

** Note:- Buffer not required in Non-Pipeline design because Data dependency problem doesn't occured in the Non-Pipeline.

Que:- Consider a CPU with a clk cycle of 2 ns uses the 4 stage pipeline design. In this design different stages are taking different cycles to complete the assigned task i.e. S_1 take 2 cycle, S_2 take 1 cycle. S_3 takes 2 cycles and S_4 takes 4 cycles. In the Pipeline design 1 cycle delay buffer

is used b/w the stages. How much time is required to execute the 100 tasks in the ∞ pipeline.

1 cycle = 2ns

S_1 = 2 cycles = 4ns

S_2 = 1 cycles = 2ns

S_3 = 2 cycles = 4ns

S_4 = 4 cycles = 8ns

Buffer delay = 1 cycle = 2ns

t_p = max (6 *, 4 *, 6 *, 8) ← last stage so No Buffer Connected.

= 8 ns

$E.T_{pipe}$ = (k + n - 1) * t_p = (4 + 99) * 8 ns

= 103 x 8 ns

= 824 ns

Que:- Consider 4 stage Pipeline with a respective stage delays of 2ns, 4ns, 8ns, 1ns. In the Pipeline enhancement Process longer delay stage is further divided into two substages of equal delays. what is the clock frequency in the Enhanced pipeline.

4ns ⇒ 4×10^{-9} ⇒ 250 MHz

$f = \frac{1}{4 \times 10^{-9}}$ ↗

Sol^n:- 2ns, 4ns, 8ns, 1ns

Old_{pipe} = δ_1 δ_2 δ_3 δ_4
↓ ↓ ↓ ↓
2ns 4ns 8ns 1ns

New_{pipe} = 2ns 4ns δ_3 δ_4
↓ ↓ 1ns
4ns 4ns

Cycle time = tp = Max (2, 4, 4, 4, 1)
= 4ns

frequency = $\frac{1}{4ns}$ = 250 MHz

★ Que:- Consider a Non Pipeline Processor with a clk cycle time of 4ns used to execute 200 instructions with some CPI & set of instructions are executed on a Pipelined Processor which contain 5 stages with a cycle time of 1.8 ns. what is the Performance gain.

ET_{pipe} = $(k + n - 1) * tp$
= $(5 + 199) * 1.8$
= $204 * 1.8$ = 367.2ns

$ET_{Non\ pipe}$ = $200 \times \underbrace{4 \times 8}_{CPI} = 800ns \times 8 = 6400ns$

δ = $\frac{ET_{Non\ pipe}}{ET_{pipe}} = \frac{800 \times 8}{367.2}$ = 17.4