

COMPUTER SCIENCE

Computer Organization and Architecture

Instruction Pipelining

Lecture_01



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**

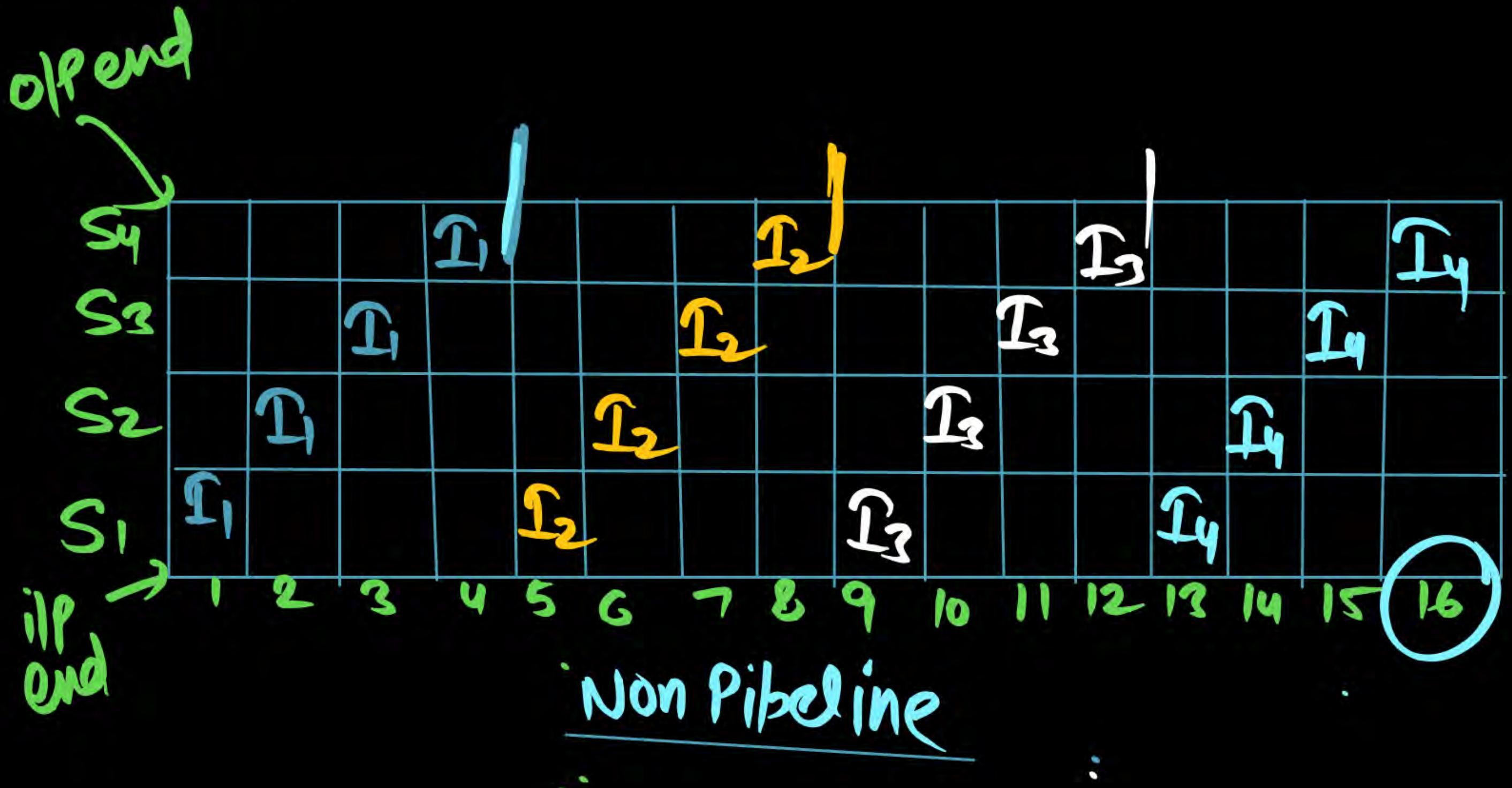
o1

Pipelining Concepts

- ① Introduction of CoA
- ② MLC Design & AM
- ③ Floating Point Representation
- ④ ALU Data Path & Control Unit
- ⑤ Pipelining

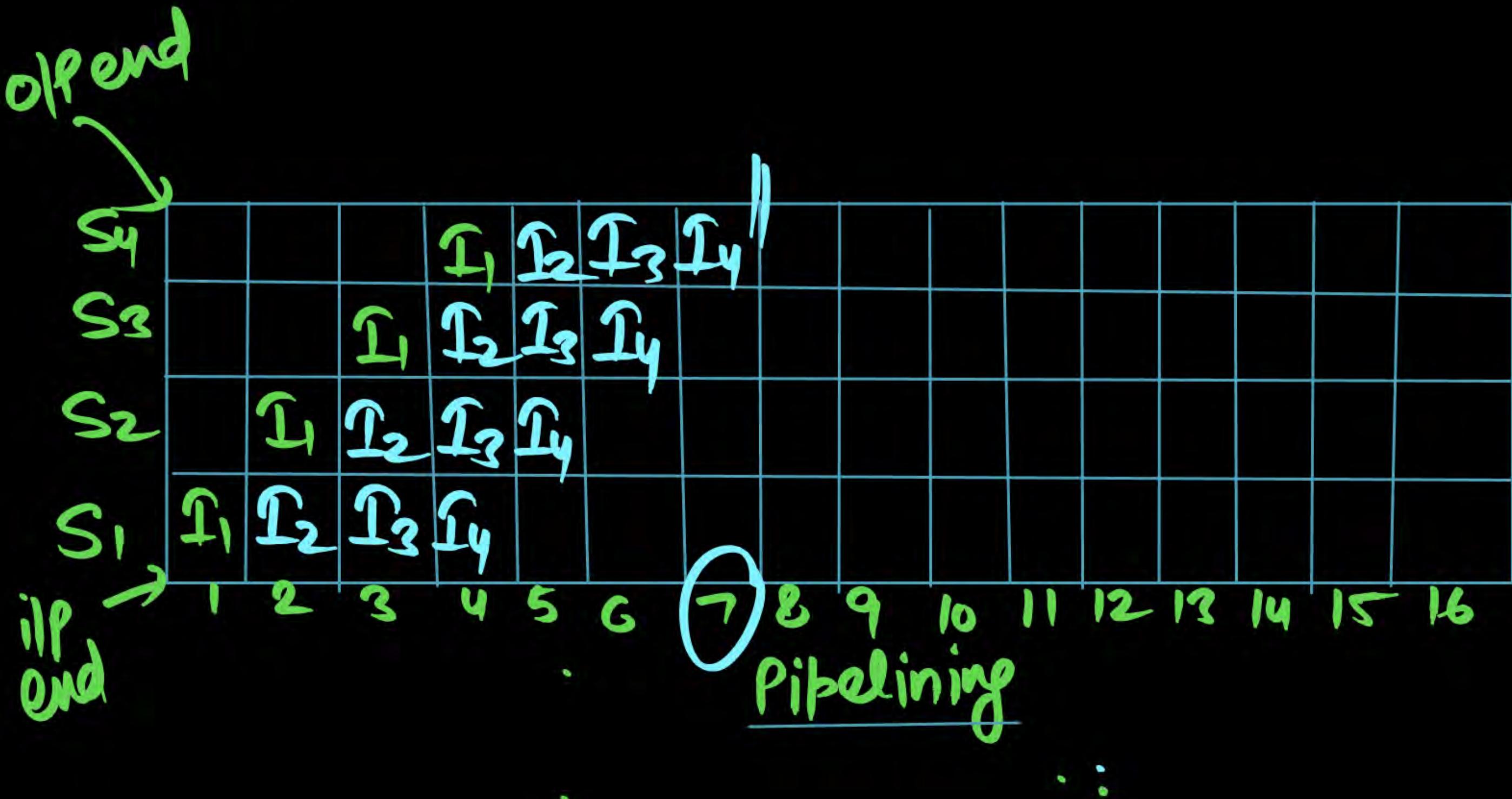
PIPELINE

P
W



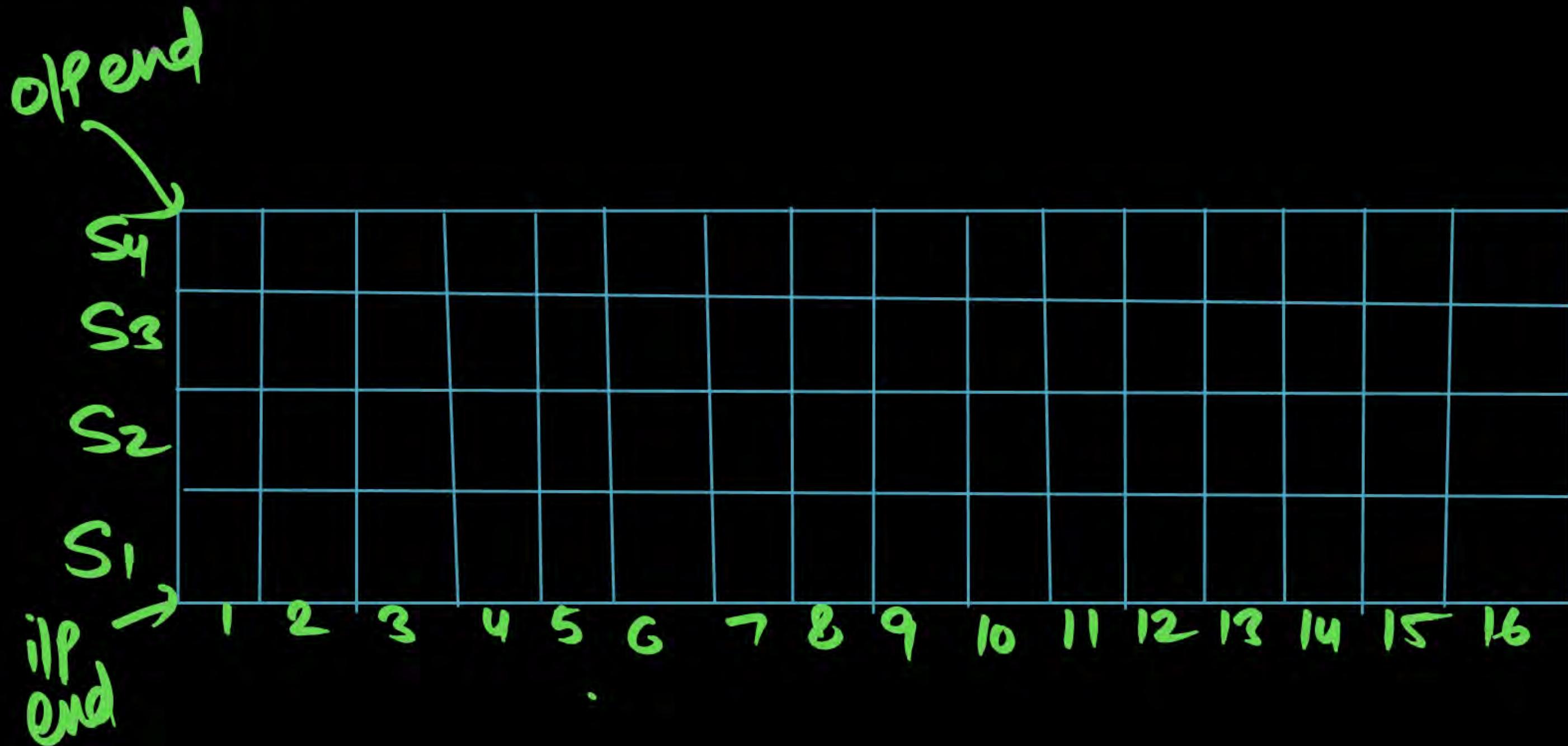
PIPELINE

PW



PIPELINE

P
W



PIPELINING

Pipeline is mechanism which are used to Improve the Performance of the System, in Which task are executed in a Overlapping Manner.

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

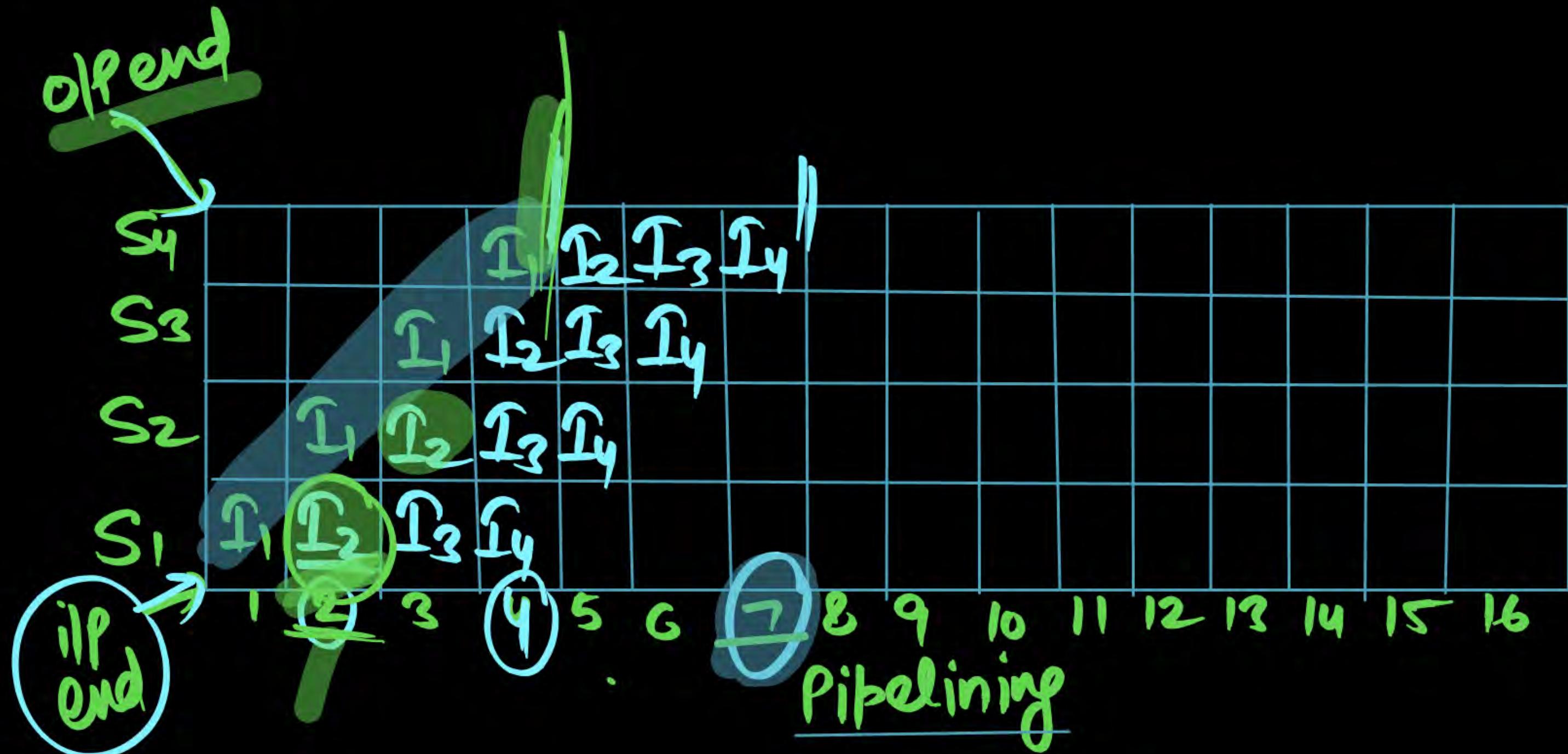


New inputs are
Accepted at one end
Before previously
Accepted inputs
Appear as output at
The other end

- 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.

PIPELINE

P
W



Pipeline



Accepting New input before the previously accepted input appears as a output at other end.

That means New input are executed along with Old input in a overlapping manner.

PIPELINE

P
W

opend



IP

out

IP

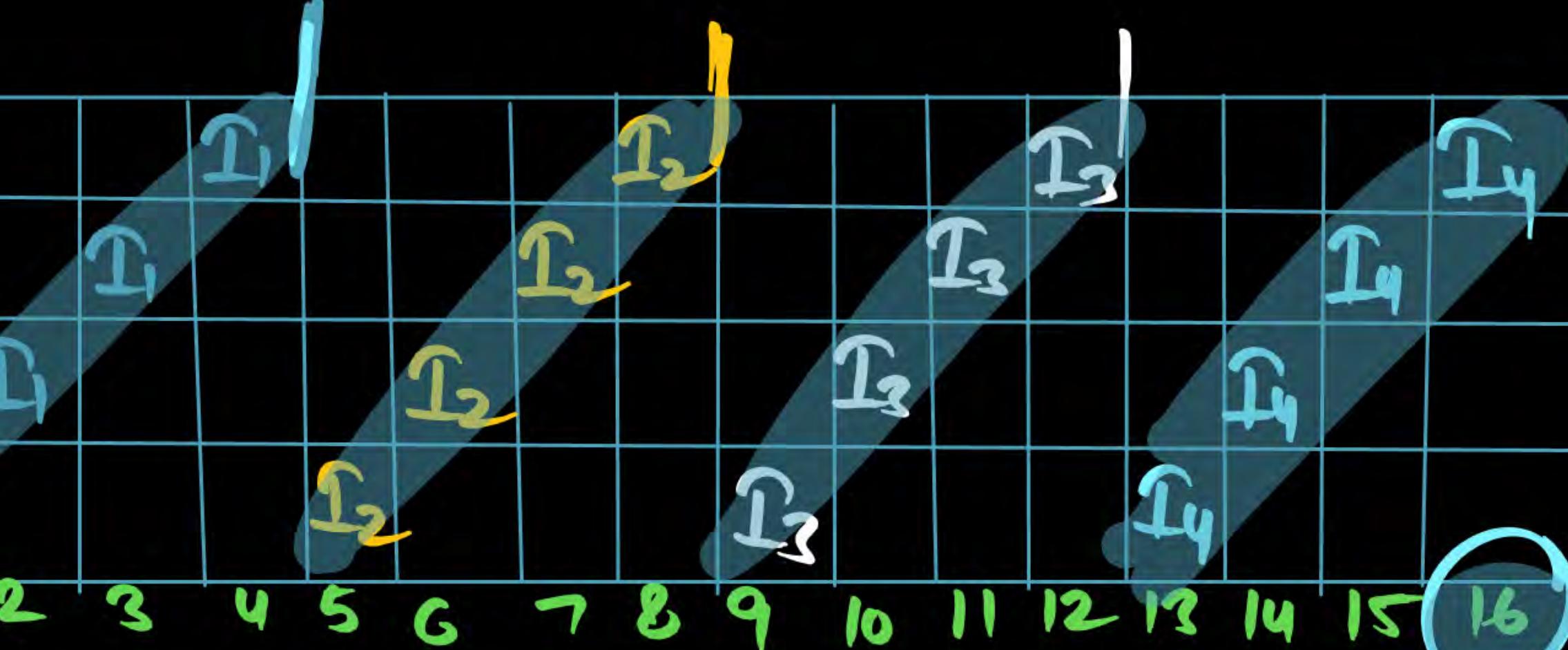
out

IP

out

Non Pipeline

Non Pipeline



Non pipeline :

Accepting New Input after the Completion of
OLD Input

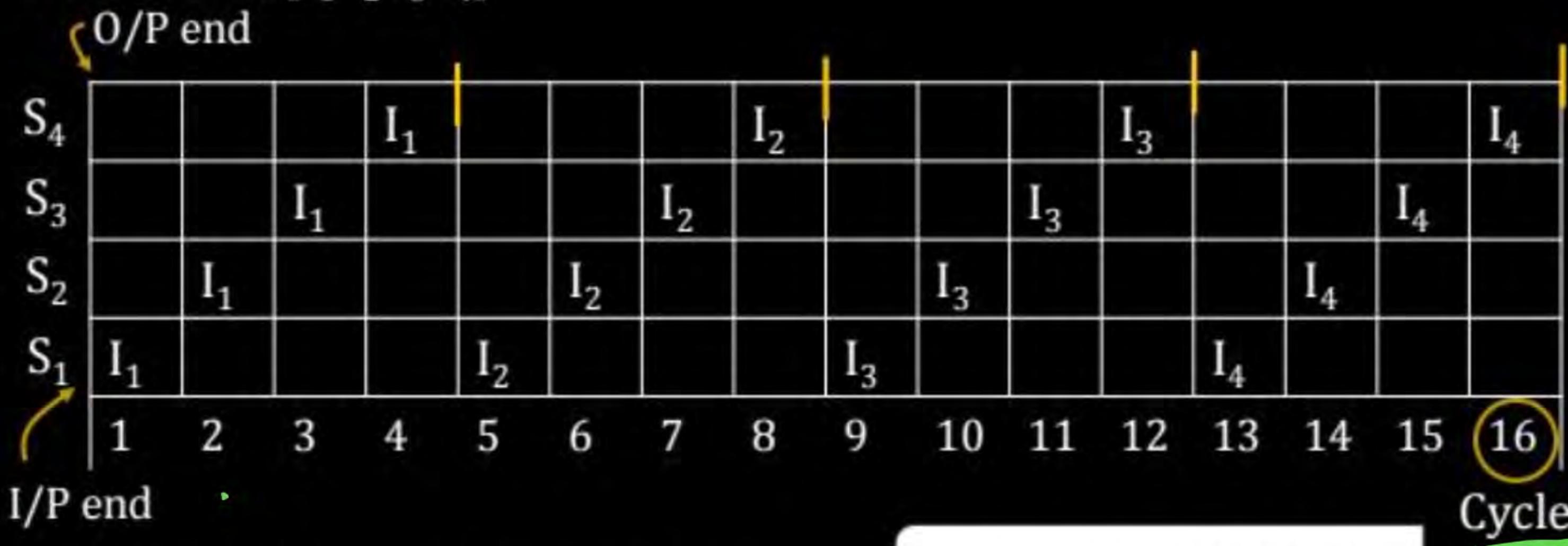
(OR)

Accepting New Input After, the Previously accepted
Input appear as a output of other end. In which
Non overlapping execution occur.

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

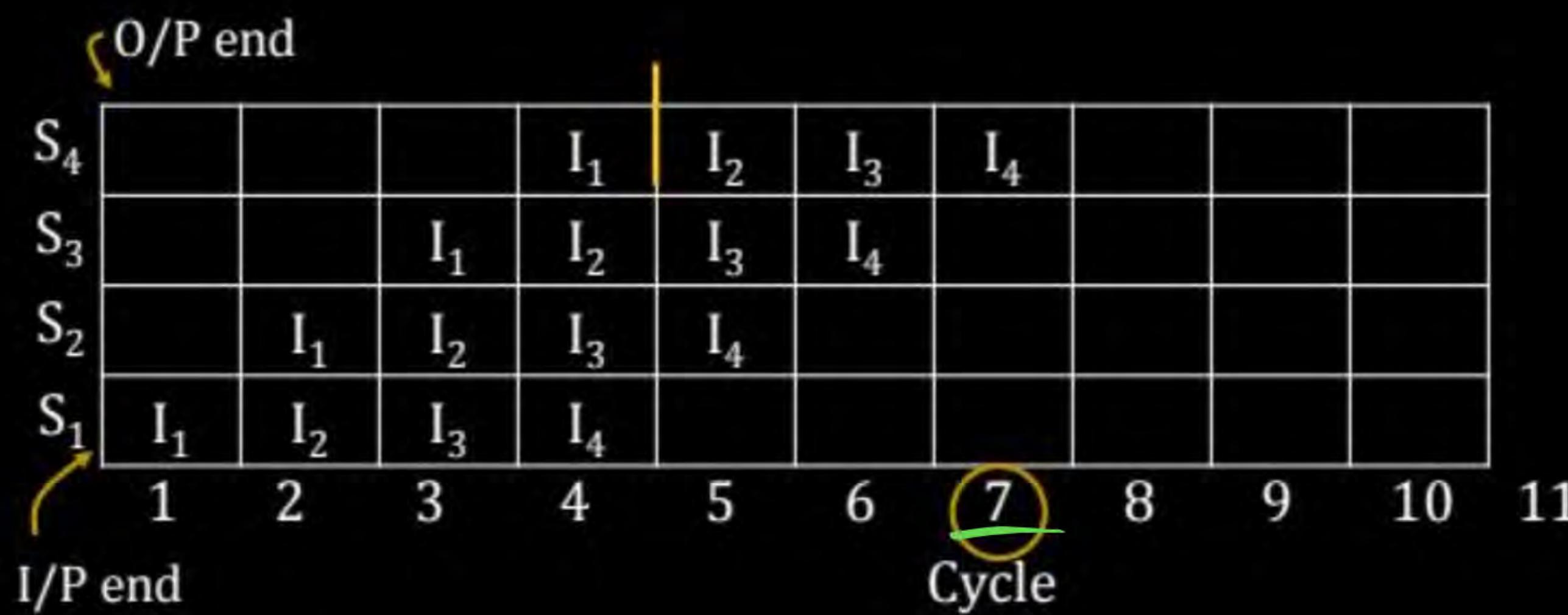
Segment/stages = $[S_1 \ S_2 \ S_3 \ S_4]$

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



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

Non-PIPELINE



PIPELINE

$k = 4$

$n = 4$

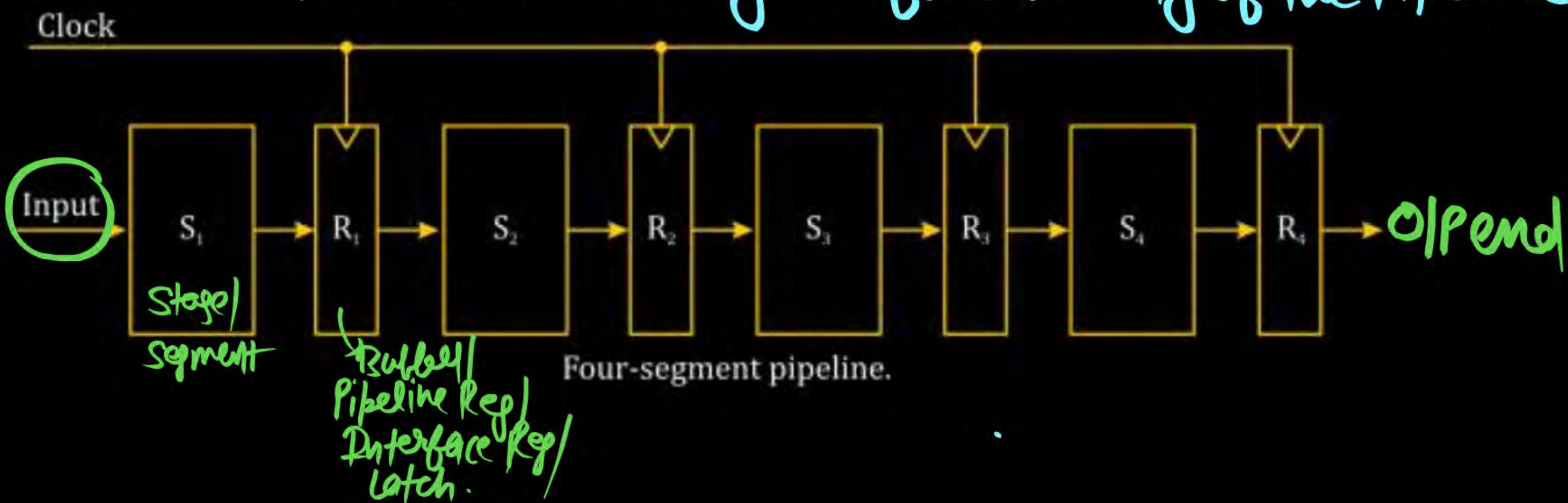
PIPELINE Design

Pipeline has 2 end

Input end

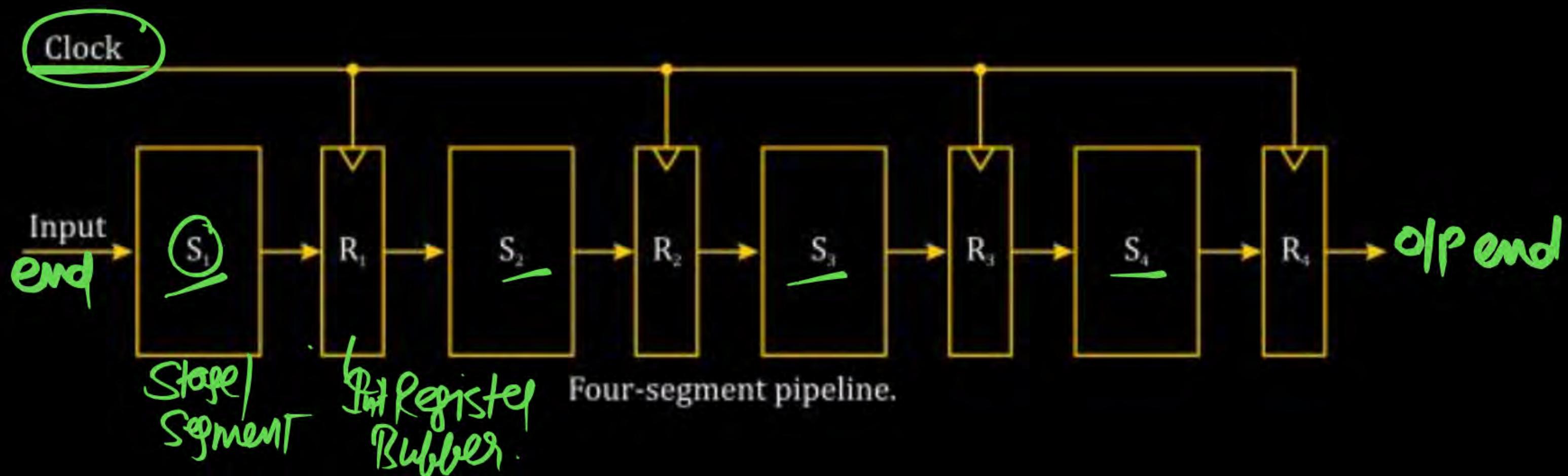
Output end

Between the Input & output end multiple pipes
are interconnected to satisfy the functionality of the Pipeline.



PIPELINE Design

Segment \circlearrowleft Stages [$S_1 S_2 S_3 S_4$]



PIPELINE Design

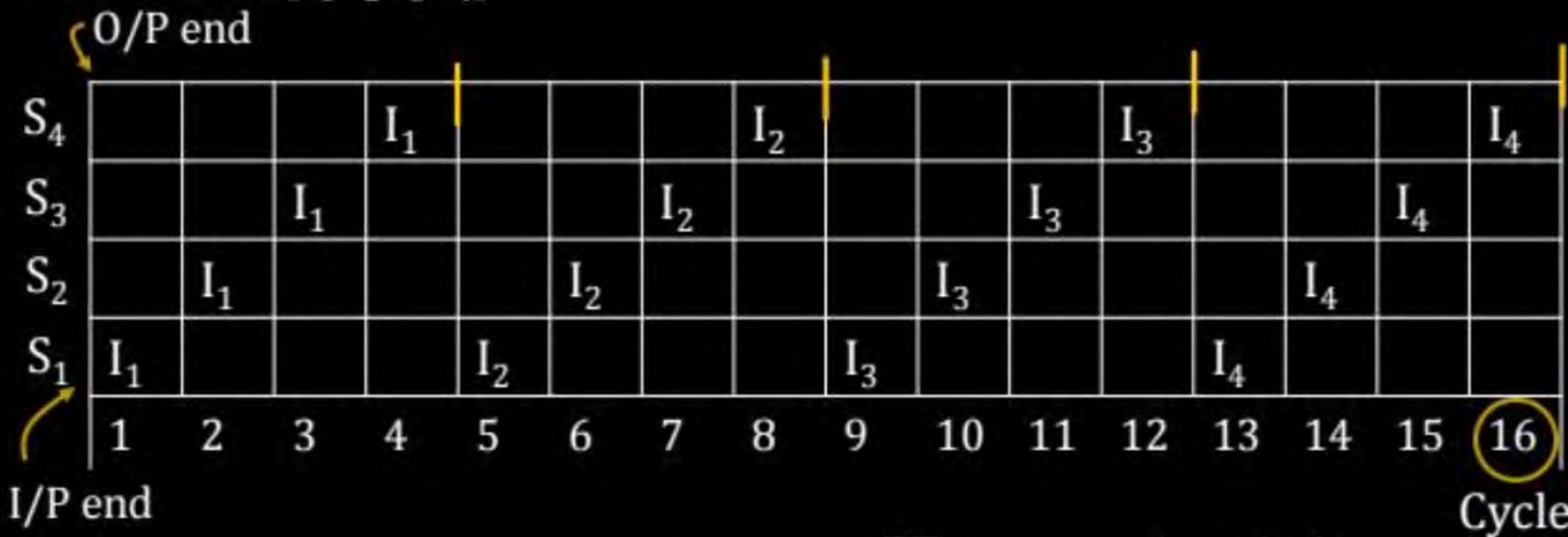


- These Pipes are called Stage or Segment.
- Between the Stages Buffer are Used to Store the Intermediate Result.
- These Buffer are also Called as Latch, Pipeline Register, Interface Register.
- All the Stages along With the Buffer are Controlled by Common clock.

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

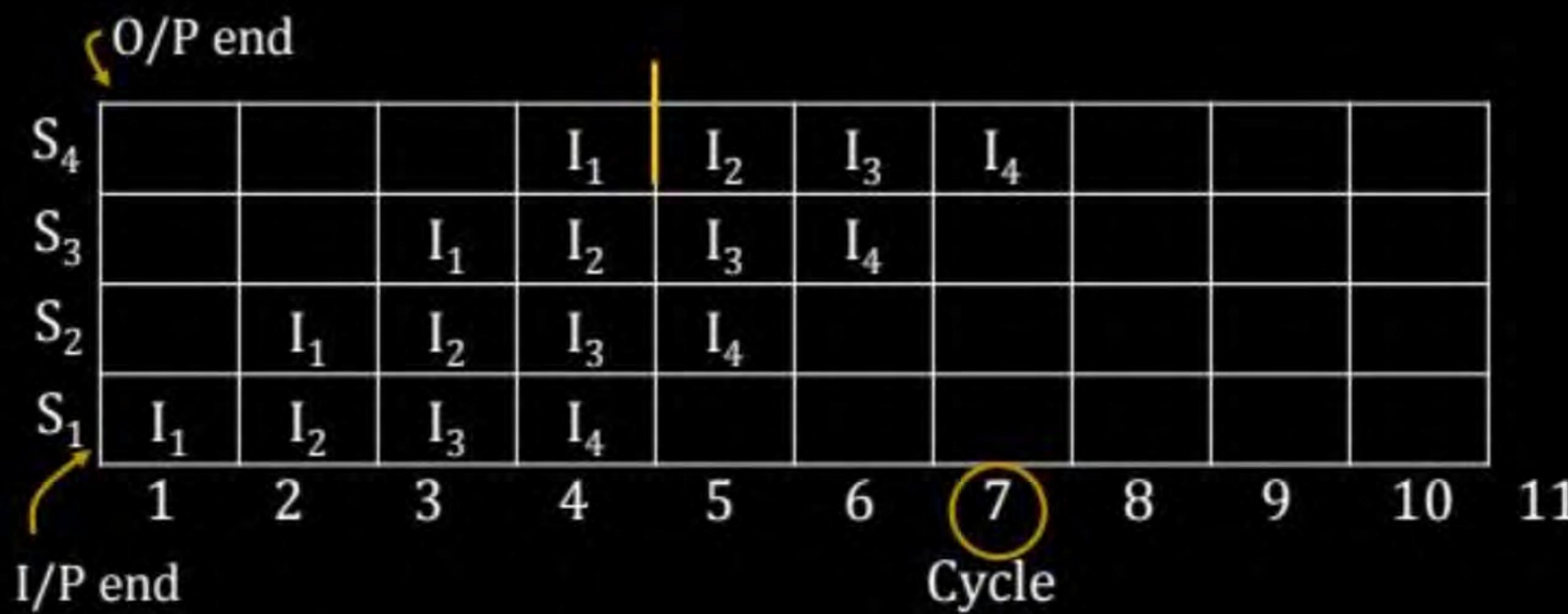
Segment/stages = $[S_1 \ S_2 \ S_3 \ S_4]$

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



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

$$\begin{aligned} ET_{\text{NONPIPE}} &= nt_n \Rightarrow 4 \times 4 \\ ET_{\text{NONPIPE}} &= 16 \text{ cycle} \end{aligned}$$



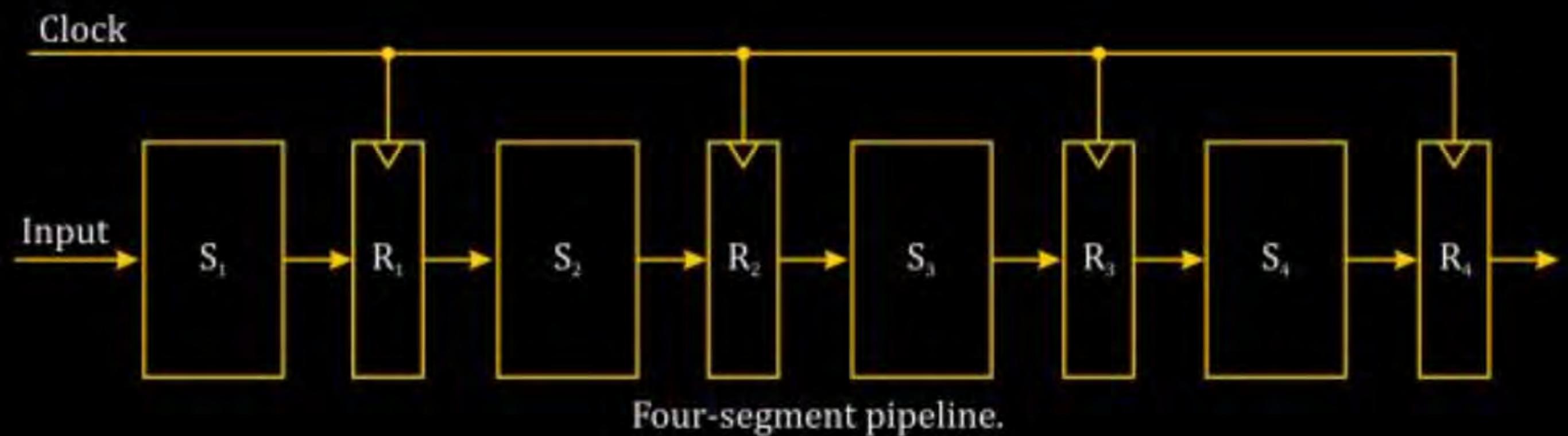
PIPELINE

$$k = 4$$

$$n = 4$$

$$\begin{aligned} ET_{PIPE} &= (k + (n - 1))tp \\ &= (4 + (4 - 1) \times 1 \text{ cycle}) \end{aligned}$$

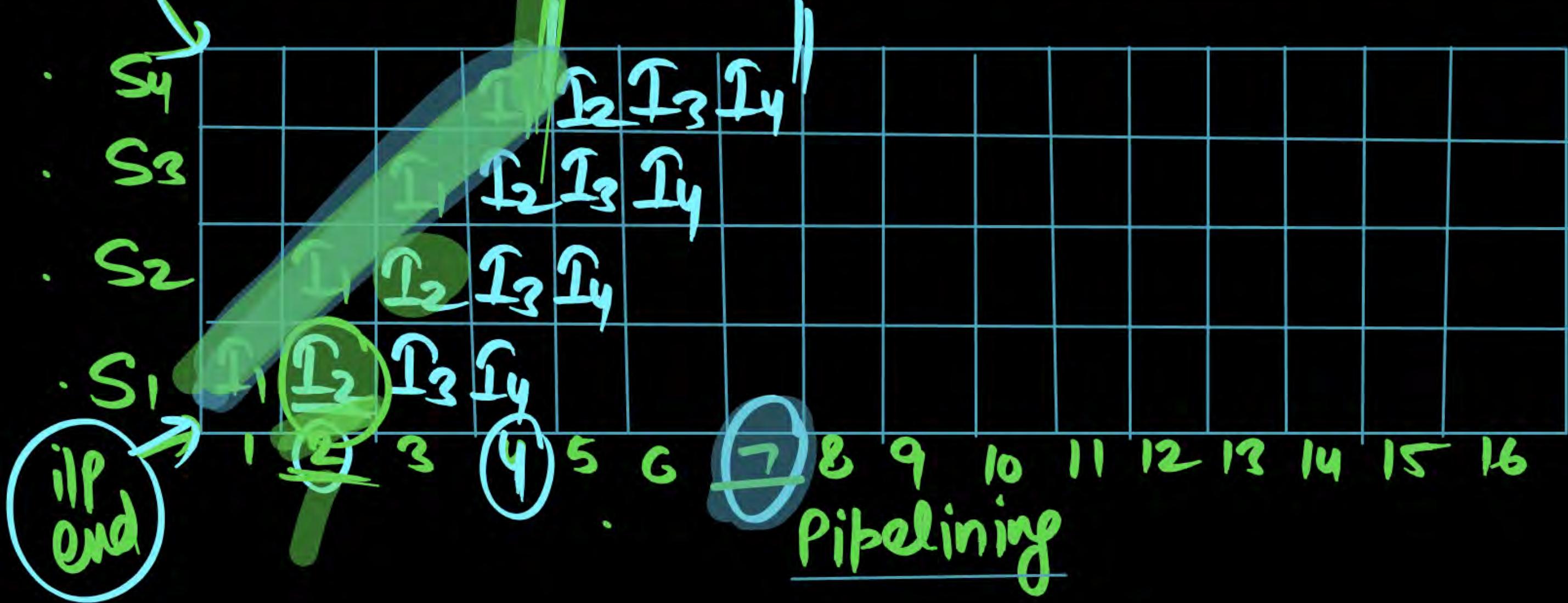
$$ET_{PIPE} = 7 \text{ cycle}$$



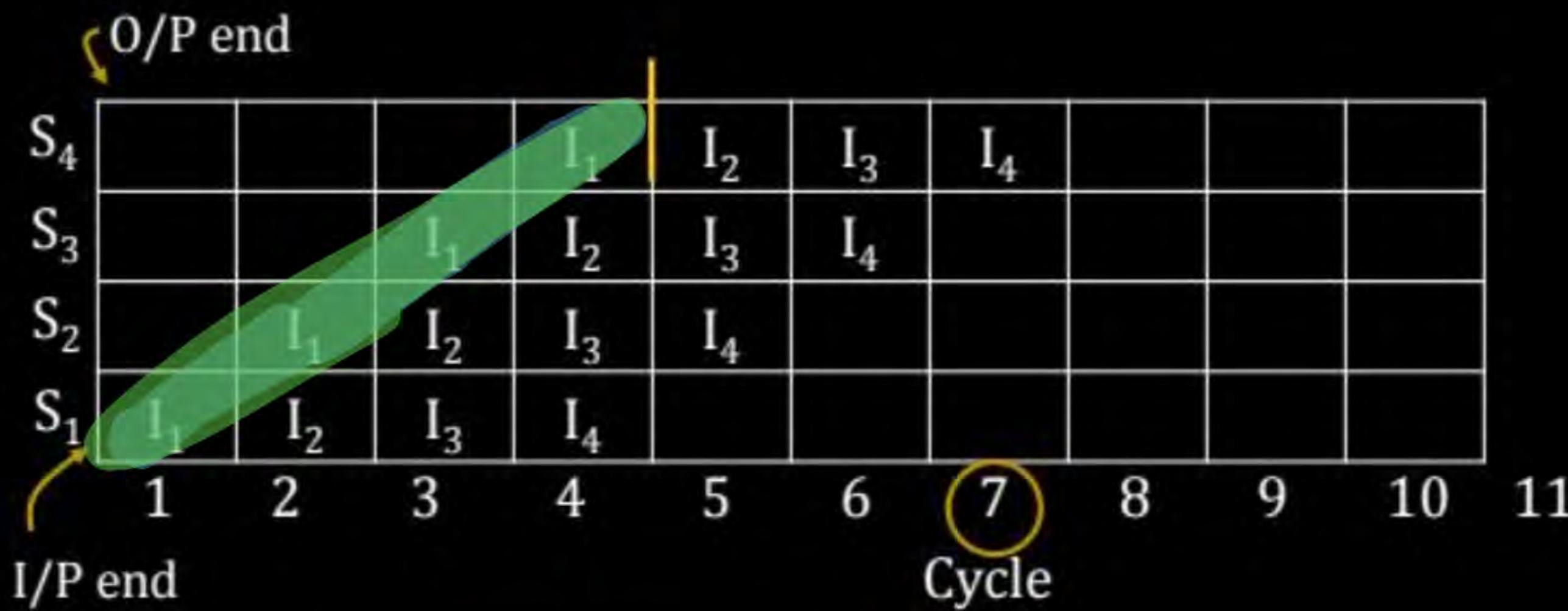
PIPELINE

P
W

Op end



$K = 4$ (S_1, S_2, S_3, S_4) # Stage / segment : K
 $n = 4$ (T_1, T_2, T_3, T_4)
 $t_p = 1 \text{ Cycle}$. Cycle time = t_p .



PIPELINE

$$\checkmark k = 4$$

$$\checkmark n = 4$$

$$\begin{aligned} \text{ET}_{\text{PIPE}} &= (k + (n - 1))tp \\ &= (4 + (4 - 1) \times 1 \text{ cycle}) \end{aligned}$$

$$\boxed{\text{ET}_{\text{PIPE}} = 7 \text{ cycle}}$$

K : Stage / Segment

tp: cycle time .

n: Number of Instruction .

PIPELINE Performance Evaluation

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.

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

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

k: # Stages (Segment)

n: # Instruction (# Task)

t_p: Each Stage Delay in Pipeline.

Q3) 4 stage, & 4 Instⁿ then ETPIPE

$$ET_{PIPE} = [k + (n-1)] \text{cycle}$$

$$\Rightarrow [4 + (4-1)]$$

$$ET_{PIPE} = 7 \text{cycle}$$

Q) $k=4, n=6$ then $ET_{PIPE} = ?$

$$ET_{PIPE} = [k + (n - 1)] \text{ cycle}$$

$$\Rightarrow [4 + (6 - 1)]$$

$$ET_{PIPE} = 9 \text{ cycle}$$

$$\begin{array}{c} n=6 \\ \hline k=4 \end{array}$$

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	

$$\begin{aligned} & [k + (n-1)] \text{ cycle} \\ & (4 + (6-1)) \Rightarrow 9 \text{ cycle} \end{aligned}$$

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.

$$ET_{PIPE} = [k + (n-1)] \text{ cycle}$$

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

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

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

$$\text{ETPIPE} = [k + (n - 1)] t_p.$$

k : # Stage (Segment)

n : # Inst (Task)

t_p : Each Stage Delay
in Pipeline .

Execution Time in Non pipeline

1 Instruction take t_n to execute in Non pipeline

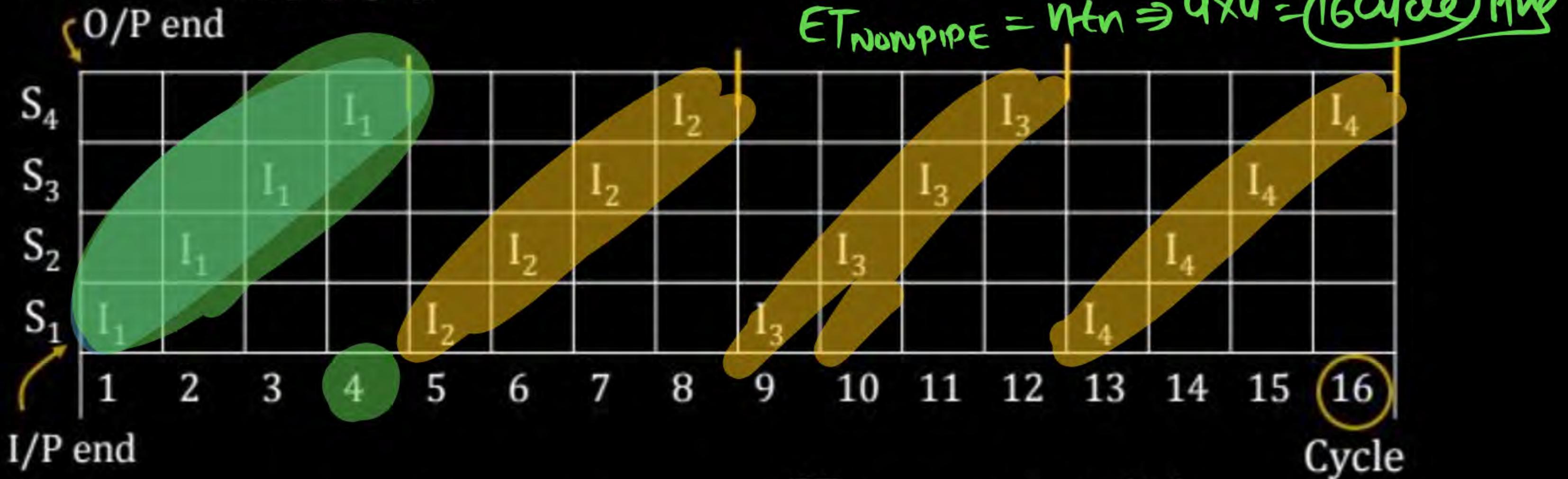
n Instruction

$$ET_{NONPIPE} = n t_n$$

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



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

$$ET_{NONPIPE} = n \cdot t_n \Rightarrow 4 \times 4$$

$$ET_{NONPIPE} = 16 \text{ cycle}$$

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

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

t_n : Each Instruction Execution time
in Non pipeline

n : # Instruction (# Tasks)

Performance Gain of Pipeline Processor.

Performance Gain
@
SPEED UP Factor [S] = $\frac{\text{Performance of Pipeline}}{\text{Performance of Non Pipeline}}$

$$\Rightarrow \frac{1}{ET_{PIPE}} / \frac{1}{ET_{NONPIPE}}$$

Performance of $\frac{1}{ET}$

$$\text{SPEED UP FACTOR } [S] = \frac{ET_{NONPIPE}}{ET_{PIPELINE}}$$

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

CASE I
 $k=4, n=4$

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

$$S \Rightarrow \frac{4 \cdot tn}{(4 + (4-1)) \cdot tp}$$

$$S = \frac{4tn}{7tp}$$

$$S = 0.56 \frac{tn}{tp}$$

CASE II
 $k=4, n=100$

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

$$S = \frac{100 \cdot tn}{(4 + (100-1)) \cdot tp}$$

$$S \Rightarrow \frac{100 \cdot tn}{103 \cdot tp}$$

When n is very large then
Value of $(k + (n-1)) \approx n$

CASE III
 $k=4, n=1000$

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

$$S = \frac{1000 \cdot tn}{1003 \cdot tp}$$

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

CASE IV
 $k=4, n=10,000$

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

$$S = \frac{10000 \cdot tn}{10003 \cdot tp}$$

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

Performance Gain of Pipeline Processor.

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

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

When very large Number of Task/Inst Executed \textcircled{or} Ideal Case
 \textcircled{or} When No. of Task is Not given
(when n value is Not given)

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

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

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

SUMMARY

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

$$S = \frac{n t_n}{[k + (n-1)] t_p}$$

$$ET_{NONPIPE} = n t_n$$

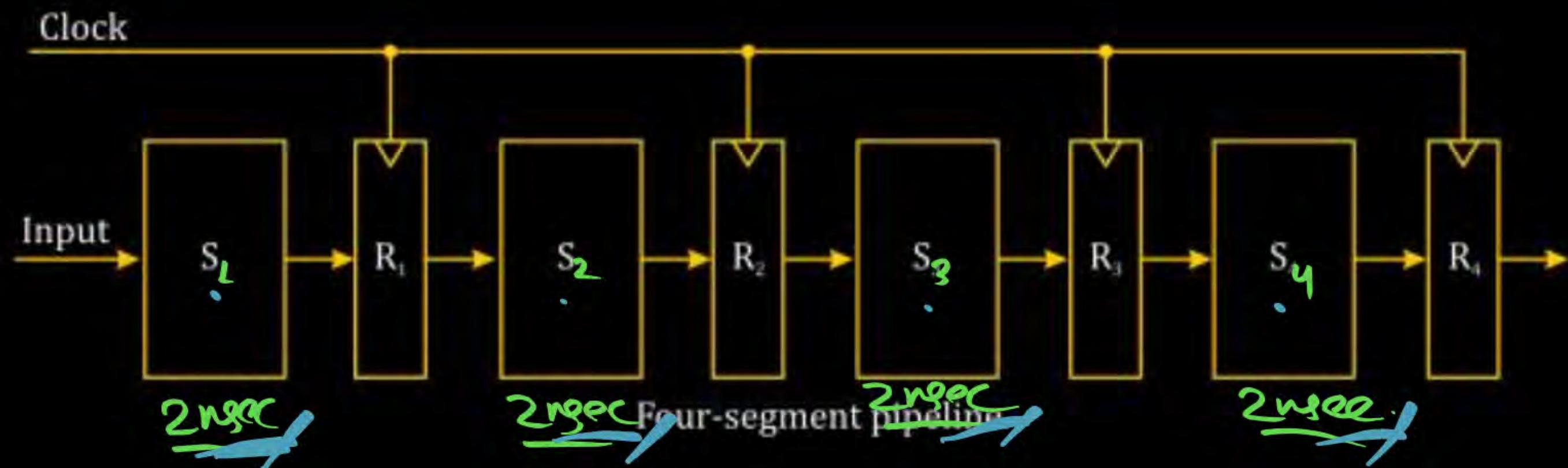
$$\text{Speed up factor (S)} = \frac{ET_{NONPIPE}}{ET_{PIPE}}$$

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

When n is Not given
Very Large
Or Ideal Case

⑧ 1 Instruction ET PIPELINE

⑨ 1 Instruction in ET NONPIPE



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

$$k=4, \quad n=1$$

$$t_p = (\text{Stage Delay}) = 2 \text{nsec}$$

$$ET_{PIPE} = [4 + (1-1)] \times 2 \text{nsec}$$

$$ET_{PIPE} = 8 \text{nsec}$$

PIPELINE

$$ET_{NONPIPE} = n t_n$$

t_n : Each Instr ET in Non pipeline

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

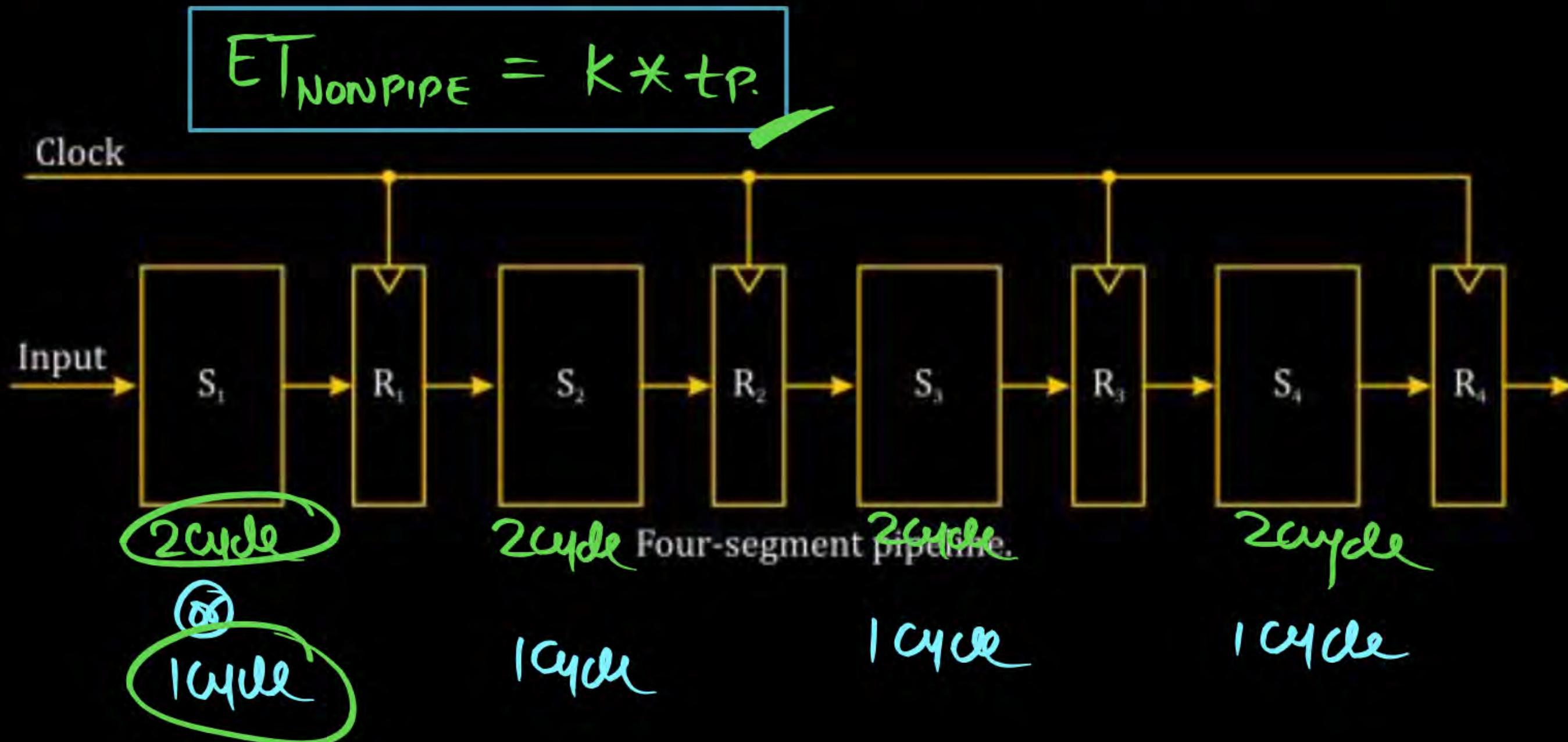
$$t_n = 8 \text{nsec}$$

$$ET_{NONPIPE} = n t_n$$
$$\Rightarrow 1 \times 8 \text{nsec}$$

$$ET_{NONPIPE} = 8 \text{nsec}$$

Non Pipeline

(Note) In Ideal Case When Pipeline Stage are Perfectly balanced then One Task ET in Non Pipeline.



$K=4$ 1 Task ET in Non Pipeline
 $tp=2\text{cycle}$ [2 cycle in each stage]

$$ET_{NONPIPE} = 2+2+2+2$$

$$ET_{NP} = 8\text{cycle}$$

$K=4$ $ET_{NP} = K * tp$

$tp=2\text{cycle}$ $= 4 \times 2\text{cycle}$

Note

$$ET_{NP} = 8\text{cycle}$$

But this is Only in the Condition When Pipeline are Perfectly Balanced.

1 Task ET in Non Pipeline
[1 cycle in each Stage]

$$ET_{NP} = 1+1+1+1$$

$$ET_{NP} = 4\text{cycle}$$

$K=4$
 $tp=1\text{cycle}$

$$ET_{NONPIPE} = K * tp$$

$$ET_{NP} = 4 \times 1$$

$K:4$
 $tp:1\text{cycle}$

$$ET_{NP} = 4\text{cycle}$$

$$S = \frac{ET_{NONPIPE}}{ET_{PIPE}}$$

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

(When n is
Not given)

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

Ideal Case,
when perfectly
Balanced

$$S = k$$

Perfectly Balanced :

Each Stage takes same
amount of time/cycle.

Through put

Number of Task completed in Per time Unit

OS: # Process Completed in Per time Unit

CN: Eff X B.W

Throughput = Rate of Output.

Through put

$[k + (n-1)]t_p$ time Unit n Task Executed.

$$\text{In 1 Time Unit} = \frac{n}{[k + (n-1)] t_p}$$

[Throughput]

(Note) In Pipeline When Number of Task are Not given

Number of task completed
Per time Unit.

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

SUMMARY Till Now :

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

$$ET_{NONPIPE} = ntn$$

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

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

[when n is Not given
@ Ideal Case]

$$\text{Speed up factor (s)} = \frac{ET_{NP}}{ET_{PIPE}}$$

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

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

η : efficiency

s: Speed up factor

K: Number of Stage (Segment)

n: Number of Instruction (Task)

tp: Each Stage Delay in Pipeline

tn: Each Instruction ET in Non pipeline.

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

(when n is very large)
(n is Not given @
in Ideal Case)

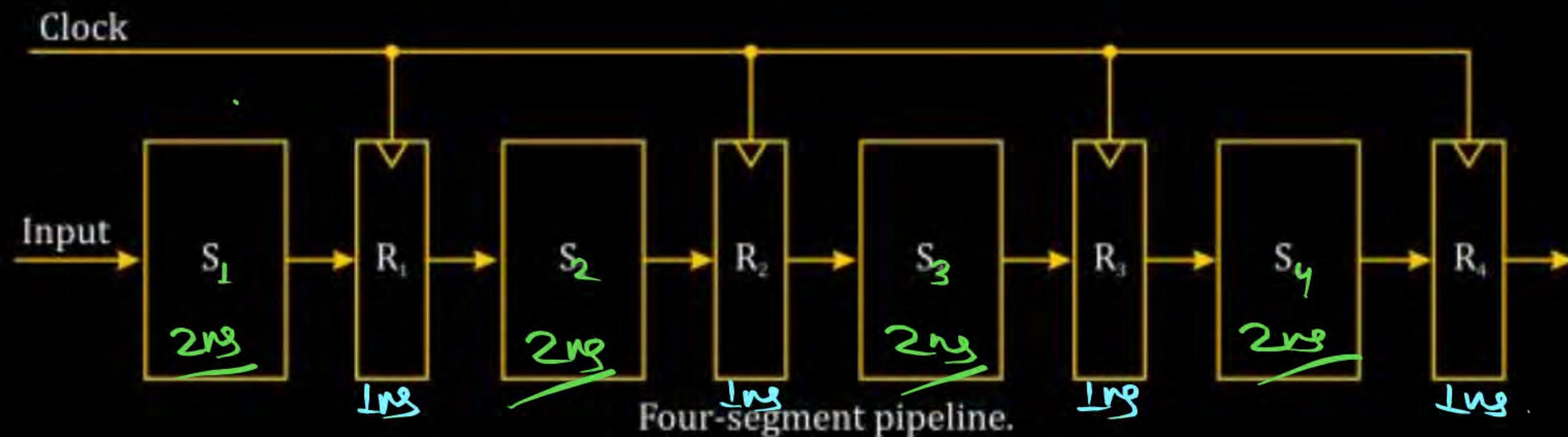
Type of Pipeline

- ① UNIFORM Delay Pipeline. Linear Pipeline
- ② Non Uniform Delay Pipeline. Non Linear Pipeline.

① UNIFORM Delay Pipeline

P
Perfectly
Balanced

In Uniform pipeline each Stage taking Same amount of time | Delay to execute the assigned operation | Task.



UNIFORM Delay Pipeline.

1 Task ET in pipeline

$$K = 4 \text{ (Stage)}$$

$$t_p: \text{Stage Delay } t_p = 2$$

$$n = L (\# \text{Inst})$$

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

$$\Rightarrow [4 + (1-1)] \times 2 \text{nsec}$$

$$ET_{PIPE} = 8 \text{nsec}$$

1 Task ET in Pipeline = 1 Task ET in Non Pipeline

1 Task ET in Non Pipeline

$$ET_{NONPIPE} = n t_n$$

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

$$t_n = 8 \text{nsec} \quad n = 1$$

$$ET_{NONPIPE} = 1 \times 8 \text{nsec}$$

$$ET_{NONPIPE} = 8 \text{nsec}$$

DB Buffer Delay is Included

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

$$t_p = 2 + 1$$

$$t_p = 3 \text{nsec}$$

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

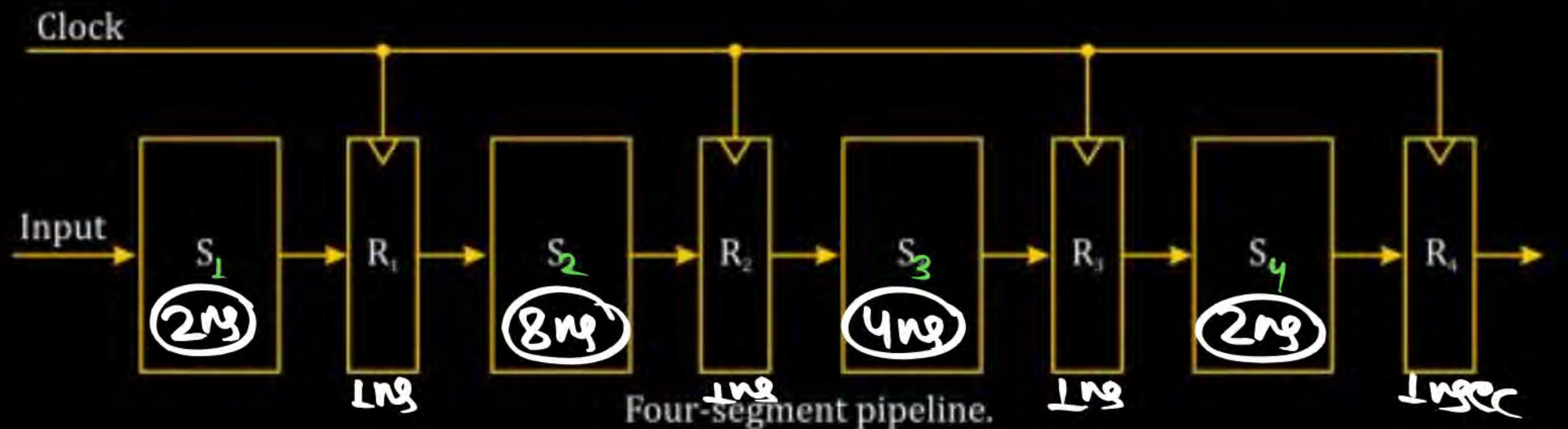
$$\Rightarrow [4 + (1-1)] \times 3 \text{nsec}$$

$$ET_{PIPE} = 12 \text{nsec}$$

$$ET_{NONPIPE} = 8 \text{nsec}$$

Non uniform Delay Pipeline -

In Non Uniform Delay Pipeline Each Stages taking Different Amount of time (Delay) to Complete the assigned Task / operation.



Non uniform Delay Pipeline

1 Task ET in Pipeline

$$K = 4, \quad n = 1$$

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

$$t_p = \max(2, 8, 4, 2)$$

$$t_p = 8 \text{nsec}$$

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

$$\Rightarrow [4 + (1-1)] 8 \text{nsec}$$

$$ET_{PIPE} = 32 \text{nsec}$$

1 Task ET in Non pipeline

$$ET_{NONPIPE} = n t_n$$

$$t_n = 2 + 8 + 4 + 2$$

$$t_n = 16 \text{nsec}$$

$$ET_{NONPIPE} = n \cdot t_n$$
$$\Rightarrow 1 \times 16 \text{nsec}$$

$$ET_{NONPIPE} = 16 \text{nsec}$$

If Buffer Delay is Included

$$t_p = \max \left[\begin{array}{l} \text{Stage Delay} + \text{Buffer Delay} \\ \dots \end{array} \right]$$

$$\Rightarrow \max [(2+1), (8+1), (4+1), (2+1)]$$

$$t_p = 9 \text{nsec}$$

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

$$\Rightarrow [4 + (1-1)] \times 9$$

$$ET_{Pipeline} = 36 \text{nsec}$$

Non uniform Delay Pipeline

When Number of Task/Instruction Increase.

if $n = 1000$ in Previous Question

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

$$\Rightarrow (4 + (1000-1)) \times 8 \text{ nsec}$$

$$ET_{PIPE} = 1003 \times 8 \text{ nsec}$$

$$ET_{PIPE} = 8024 \text{ nsec}$$

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

$$t_n = 2 + 8 + 4 + 2$$

$$t_n = 16 \text{ nsec} \quad n = 1000$$

$$ET_{NONPIPE} = n \cdot t_n$$
$$\Rightarrow 1000 \times 16$$

$$ET_{NONPIPE} = 16000 \text{ nsec.}$$

Important Point

Note

When Pipeline are Perfectly Balanced (Uniform Delay)

then 1 Task [Instruction] Execution Time in pipelined
is same as 1 Task ET in Non pipeline.

Note

When Pipeline are Not Perfectly Balance (Non Uniform Delay)

then 1 Task Execution in Pipeline is greater/equal to
1 Task ET in Non Pipeline.

T_1 : One Task ET in pipeline

T_2 : One Task ET in Nonpipeline.

$$T_1 \geq T_2$$

Important Point

(Note)

But When the Number of Task (Instruction) Increased
then Pipeline Performance is Best Compare
to Non pipeline.

(Note)

Buffer Delay is Not Include in Non pipeline, Because
In Non pipeline we are Not Storing the (non overlapping execution) Intermediate Result.
So Buffer Delay is include Only in pipeline When it is given in Question.

UNIFORM Delay

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

Non Uniform Delay Pipeline

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

SUMMARY Till Now :

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

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

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

$$\text{Throughput} = \frac{1}{tp} \quad \begin{array}{l} \text{[when } n \text{ is Not given]} \\ \text{@ Ideal Case} \end{array}$$

$$\text{Speed up factor [S]} = \frac{ET_{NP}}{ET_{PIPE}}$$

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

$$S = \frac{tn}{tp} \quad \begin{array}{l} \text{[when } n \text{ is very large]} \\ \text{n is Not given in Ideal Case} \end{array}$$

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

η : efficiency

S: Speed up factor

K: Number of Stage (Segment)

n: Number of Instruction (Task)

tp: Each Stage Delay in Pipeline.

tn: Each Instruction ET in Non pipeline.

**THANK
YOU!**

