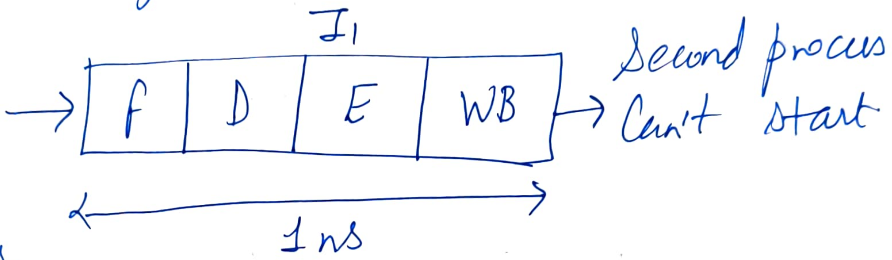
⇒ <u>Pipelining</u> :- A way of speeding up execution of Instructions. OR overlap execution of multiple instruction.

⇒ pipelining :- is a technique of decomposing a sequenctial process into suboperations, with each suboperation being executed in a dedicated segments that operates Concurrently with all other segments.

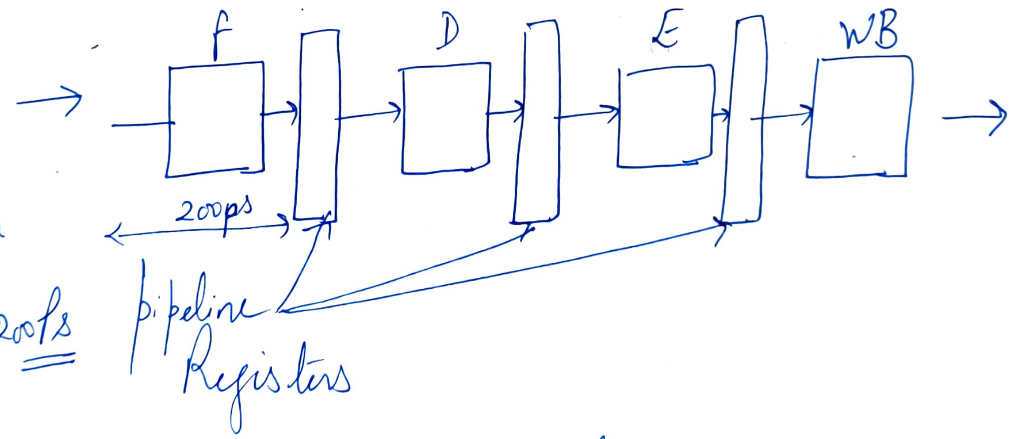
• Non-pipeling

1 operation finish every 1ns.



• Pipelining :

1 operation finished every 200ps



• Simultenerous execution of more than 1 instrucetion takes place in pipelining procedure.

Each instruction in a computer is processed wi
following sequence of step (phase)

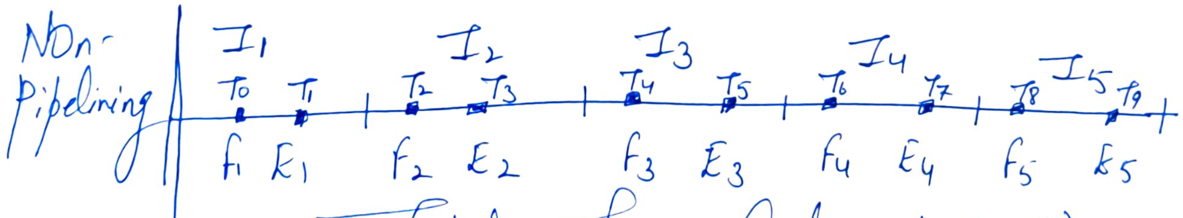
1. Fetch the instruction from memory (F1)
2. Decode the instruction (D1)
3. Fetch the operand from memory (Fo)
4. Execute the instruction (E1)
5. Store the result (Write Back) in the suitable place. (WB)

Instruction pipeline

2Stage	3Stage	4Stage	5Stage
1. Inst. fetch & Decode	1. Instruction fetch	1. Inst. fetch	1. Instruction fetch
2. Execute	2. Decode	2. Decode	2. Decode
	3. Execute	3. Execute	3. Fetch the operand
		4. Write Back.	4. Execute
			5. Store/Write Back

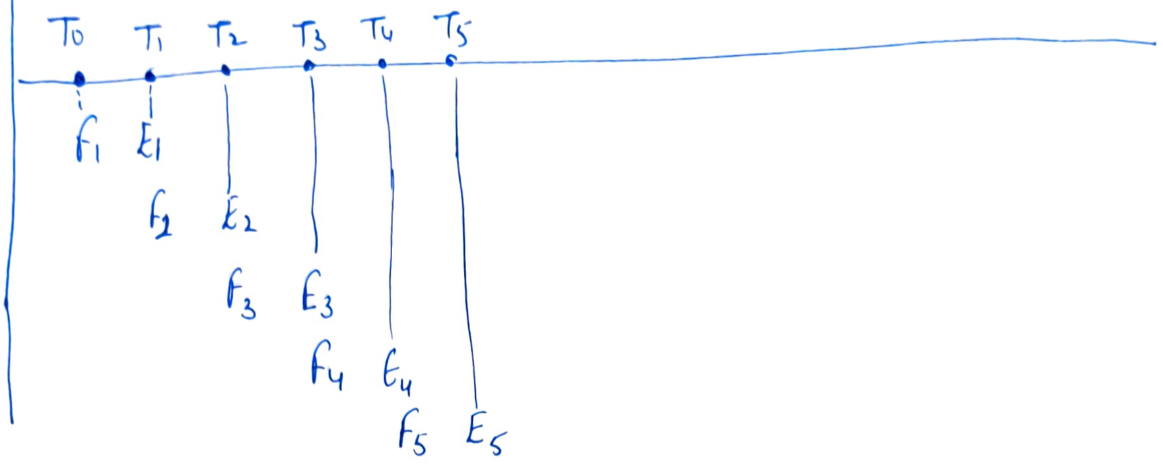
<u>2 Stage Pipelining</u>: The 2Stage pipeline CPU will break the process into <u>Instruction fetch</u> & <u>Instruction decode</u>.

Suppose in a program we have 5 instruction



Total timing Cycles = 10 (0-9)

ining



Total timing Cycle = 6

formula for Calcuting timing.

for Non-pipeline

$T = N \times K$

N: no. of instruction.

K: no. of Stages

T: timing Cycles Required

for 2 stage $K = 2$.

5 instruction $= N = 5$

$T = 2 \times 5 = 10$

for pipeline for 2 stages.

$T = K + (N-1)$

~~N~~:

$K = 2$

$N = 5$

$T = 2 + (4) = 6$

Pipeline Hazards:

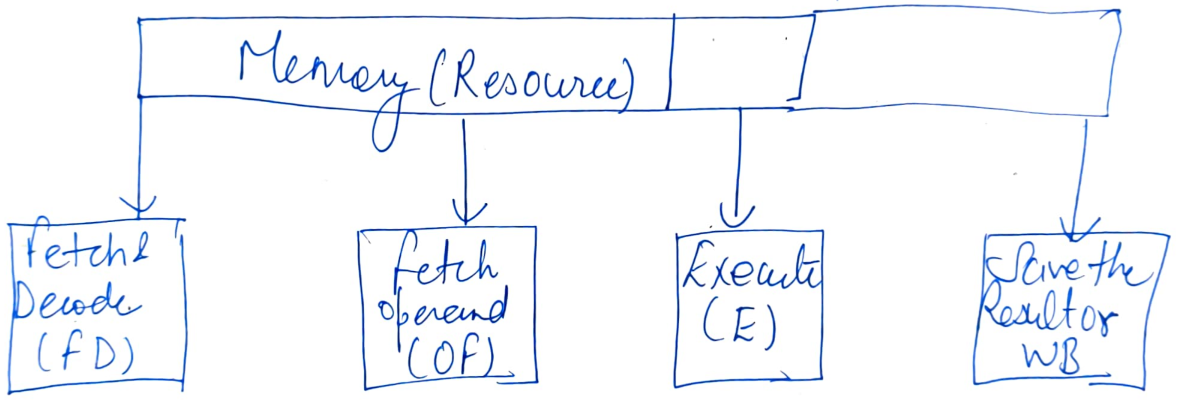
Types 1 Data
2. Structural
3. Control

Structural: Multiple Instructions share same resources: n-stage pipeline: n. Combinational ckt. but CPU Resource single: Memory, System bus

ADD R_1, R_2

ADD R_1, [2000]

eg.



	1	2	3	4	5	6	7	8	9	10	11	12
I_1	FD	OF	E	WB	↓	←--	--→					
I_2		NOP	FD	NOP	OF	E	WB	←--	--→			
I_3						FD	NOP	OF	E	WB		
I_4												

CPI = 3 (instead of CPI = 1)

olution: Resource Duplication (cost)
: NOP usage (efficiency).

Control Hazards: Branch o/come not known.

Branch) I_1	FD	OF	E	WB	<-.. Branch instruction if true -->		
I_2		~~FD~~	~~OF~~	~~E~~	~~WB~~		
I_3			~~FD~~	~~OF~~	~~E~~	~~WB~~	
I_4				~~FD~~	~~OF~~	~~E~~	~~WB~~

all in seq. will get discarded.

Sol: predictive Algo.
During decode phase / execute phase.

DataHazards: dependency of 1 instruction in pipeline on data in previous Instruction.

INC B
MOV A, B

	T_1	T_2	T_3	T_4	
INC B	IF	OF	E	WB	
MOV A, B		IF	OF	E	WB

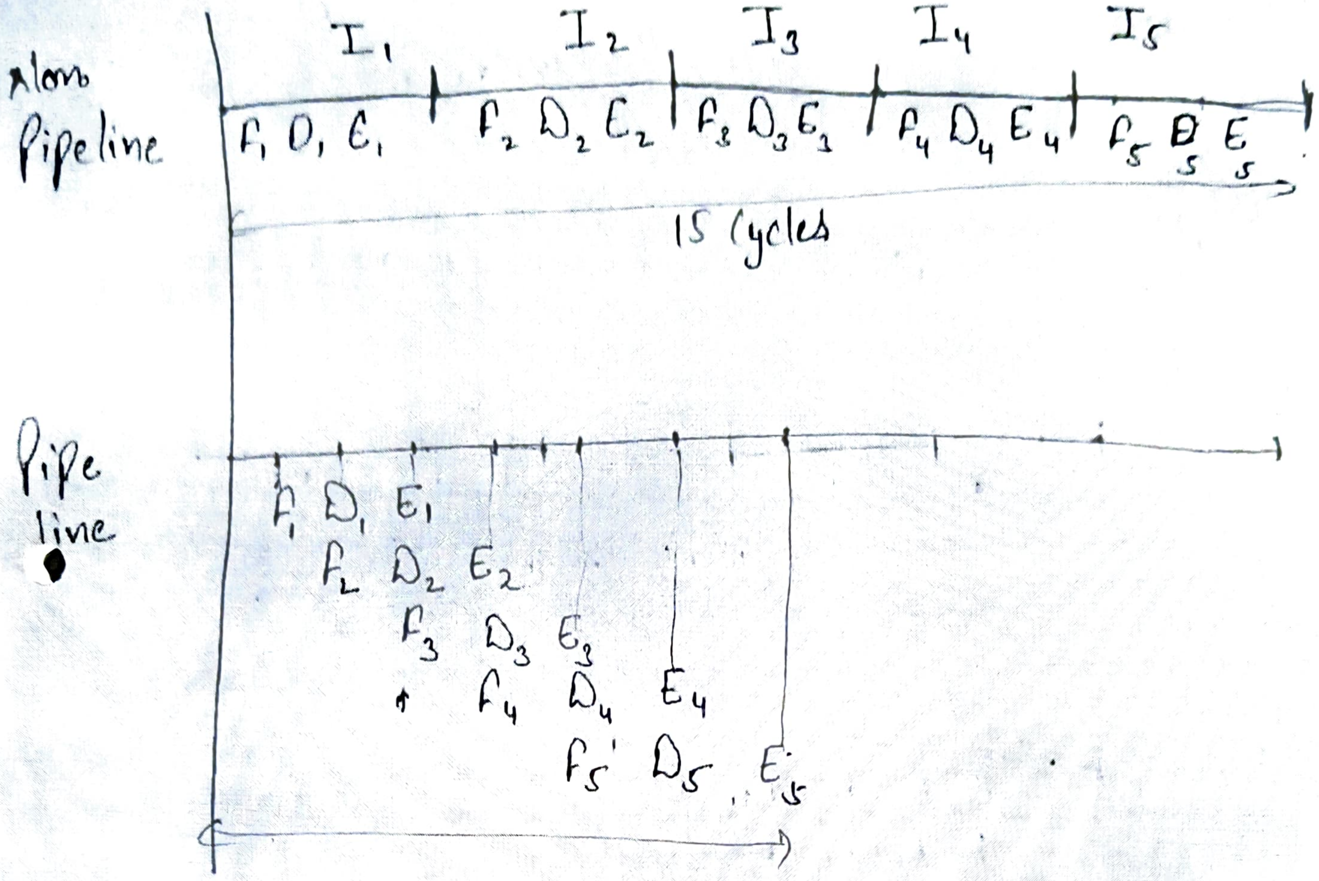
it ~~h~~ output of MOV A, B is wrong since it has not used the updated value of B obtained in INC B

Solution

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	
INC B	If	OF	E	WB				
MOV A,B		NOP	NOP	NOP	If	Of	E	WB

CPI = 3 ?

3 Stage Pipeline (5 Instruction)



Calculating Timing Cycle

For NonPipeline

or Three stage

$T = K \times N$

$= 3 \times 5 = 15$

15 Cycles

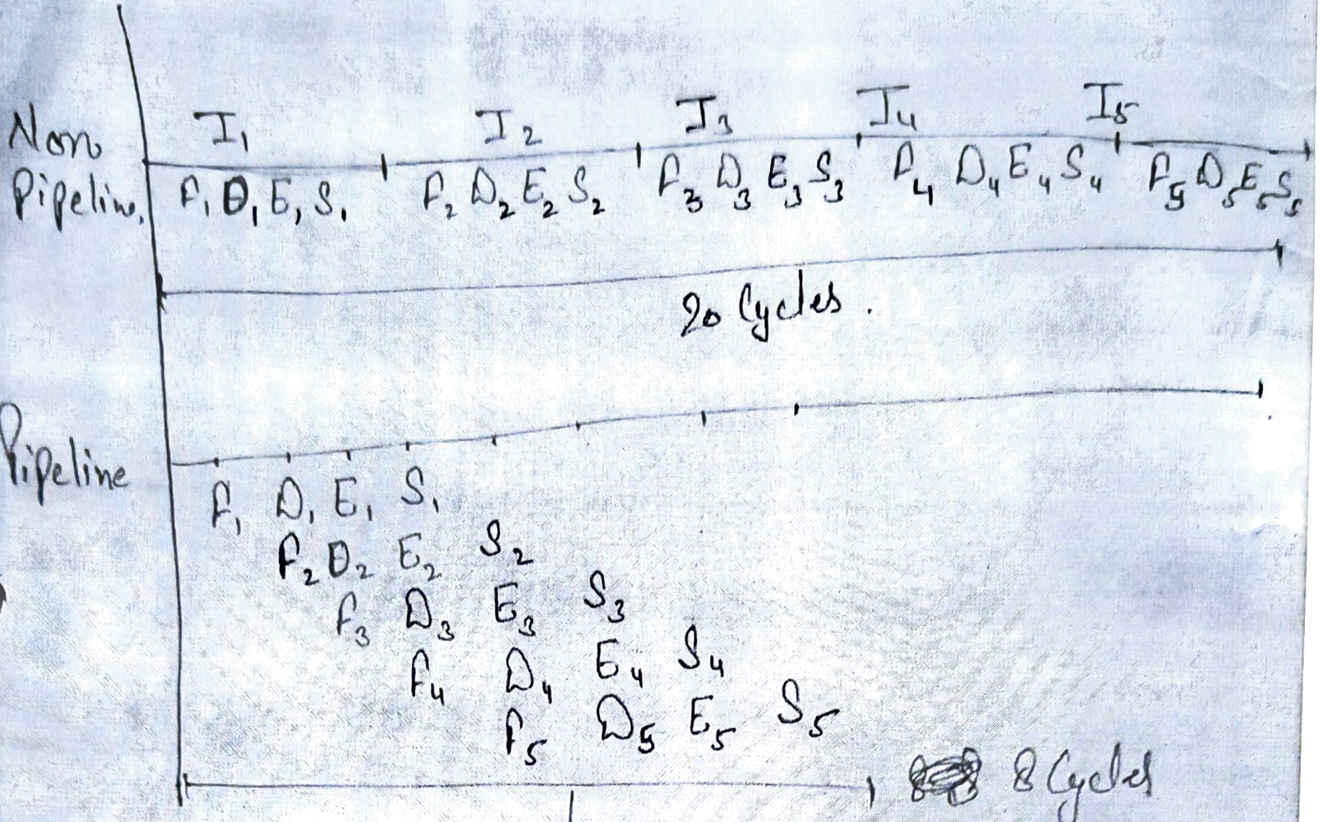
For Pipeline

$T = K + (N-1)$

$= 3 + (5-1) = 3 + 4 = 7$

7 Cycles.

4 Stage Pipeline (5 Instruction)



For Non Pipeline.

$T = K \times N$
$= 4 \times 5$
$= 20$ Cycles

For Pipeline

$T = K + (N-1)$
$= 4 + (5-1)$
$= 4 + 4 = 8$

Where

K = No. of Stages
N = No. of Instructions
T = Timing Cycles
F_i = Fetch Instruction
D_i = Decode
E_i = Execute
S_i = Store result.