Pipeline

- Pipeline is the process of arrangement of hardware elements of CPU such that its overall performance is increased.

- Existing CPU কে এমনভাবে arrange করা যাতে performance increase হয়।

- Simultaneous execution of more than one instruction takes place in pipelined process.
  ↳ একই time এ multiple instruction execute হবে।

- In pipelining multiple instructions are overlapped in execution.

Why pipeline is called enhancing process?

Ans:
Pipeline is called enhancing process because it improves performance by allowing multiple tasks to be processed simultaneously in different stages. This increases speed efficiency and resource utilization compared to doing tasks one at a time.

Example:

Instruction কে কিভাবে pipeline architecture এর মাধ্যমে execute করি :-

Suppose,
We have 8 Instructions. I_1 to I_8.

5 stage pipeline means we have 5 stages :-

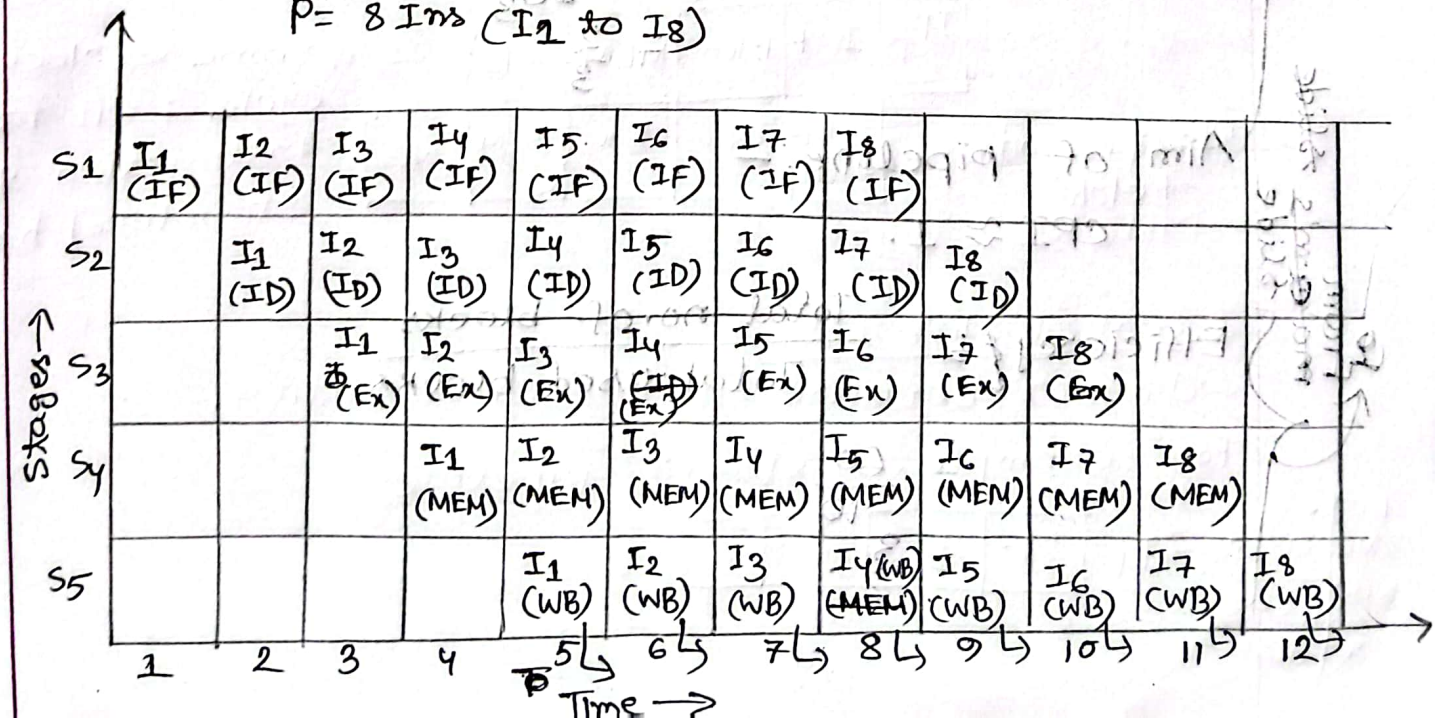
1. Instruction Fetch (IF)
2. Instruction Decode (ID)
3. Execution (Ex)
4. Memory (MEM)
5. Write Back (WB)

→ Exam এ যত stage বলবে সেটা নিয়ে কাজ করবো

Non-pipeline এ :-

1 stage complete করতে যদি 1টি clock cycle লাগে তবে 5টি stage complete করতে 5টি clock cycle লাগবে এবং 8টি instruction এ 5টি stage এর জন্য 5টি clock cycle করে $8 * 5 = 40$টি clock cycle লাগবে 8টি instruction complete হতে।

P= 8 Ins (I_1 to I_8)



In pipeline, we only need 12 clock cycle to complete the all the Instructions.

clock cycle = $k + (n-1)$; where, n = no. of instructions
k = no. of stages

For this problem,

clock cycle = $5 + (8-1)$
$= 5 + 7$
$= 12$

Speed up = $\frac{NP}{P} = \frac{40 \text{ cc}}{12 \text{ cc}}$
$= 3.33$

Pipeline is 3.33 times speeder than non-pipeline

From yacoob

Efficiency/ Utilization = $\frac{\text{No. of used blocks}}{\text{Total no. of blocks}}$
$= \frac{40}{60}$
$= \frac{2}{3}$

From mam's slide

Aim of pipeline:-
CPI ≈ 1.

Efficiency = $\frac{\text{Total no. of blocks}}{\text{Total used blocks}}$
$= \frac{60}{40}$
$= \frac{3}{2}$.

Ma

Consider a non-pipelined machine with 6 execution stages of lengths 20ns, 20ns, 30ns, 30ns, 20ns & 20ns.

(I) Find the instruction latency on this machine.

(II) How much time does it take to execute 77 instructions?

Ans:

(I) Instruction latency = (20+20+30+30+20+20)ns
= 140ns.

For N instructions = (140×N) ns.

(II) Time to complete 77 instructions = 140*70
= 10780ns

For pipeline,

Latency for pipeline = 30ns × 6 = 180ns

Time for 77 instructions in pipeline =

= (30×6) + (77-1) × 30 ns.

= 180 + (76 × 30) ns

= 2460 ns

There are 3 types of problem in pipeline:

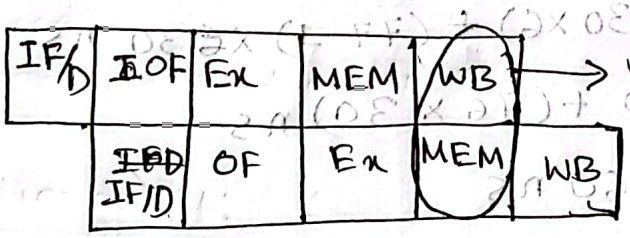
1. Structural Hazards
2. Control Hazards.
3. Data Hazards.

1. Structural Hazards:

- Resource related problem.
- Multiple instruction, same resource কে access করলো
Like:- Processor, Memory, Register, RAM, Cache etc.

For example,
Memory তে একই সময়ে ~~কোনো~~ ২টি instruction এর WB stage ~~কে~~ access ~~কে~~ করছে এবং ঐ একই সময়ে একই memory তে অন্য ২টি instruction এর কোনো ২টি stage access করছে-এর কারণে structural hazard হয়।

- Inadequate hardware to simultaneously support all instructions in pipeline in the same clock cycle.



OF=Operand Fetch

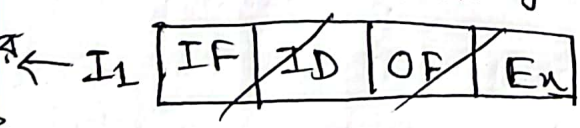
→ same clock cycle এ ২টি ~~ভিন্ন~~ instruction এর কারণে structural hazard.

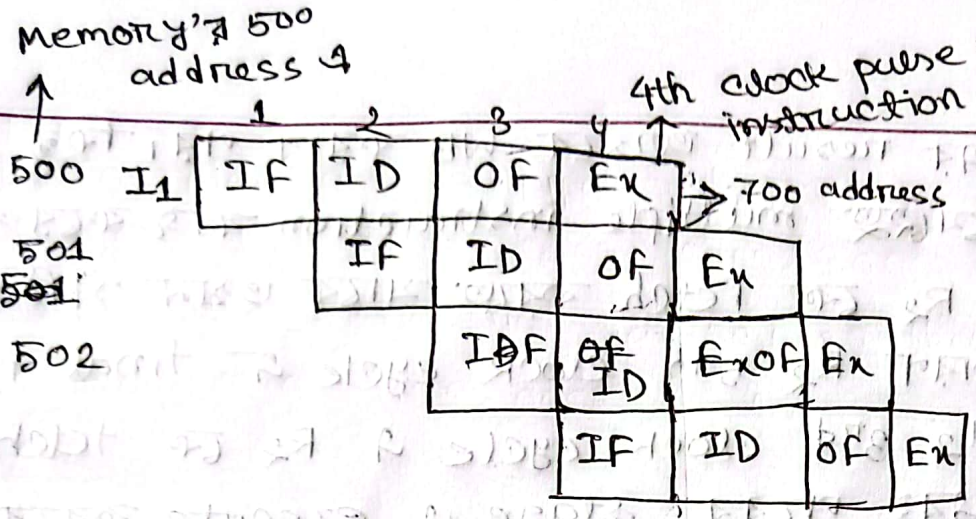
2. Control Hazards:

- Occures because of branch condition.

For example, 4 stages pipeline

Let, Memory র 500 no. address এ আছে





Instruction fetch করলে PC increment হয়ে যায়।

PC next available instruction এ point out করবে।

PC 501 এ automatically চলে যাবে।

Suppose, it's a branch instruction and we have to execute the instruction that is in 700 address. এখানে যেহেতু ২টি-২টি করে address execution করে 700 address এ যাওয়া হবে, মাঝের যত execution হয়েছে সেগুলো কোনো কাজে লাগছে না। যার কারণে বাকি সব delete করতে হচ্ছে। এটিই control hazard.

3. Data Hazards:

Data related problem.

Three Divide three parts:

(I) Read After Write (RAW) → True Dependency (TD)

(II) Write After Read (WAR) → Anti Dependency (AD).

(III) Write After Write (WAW) → Output Dependency (OD).

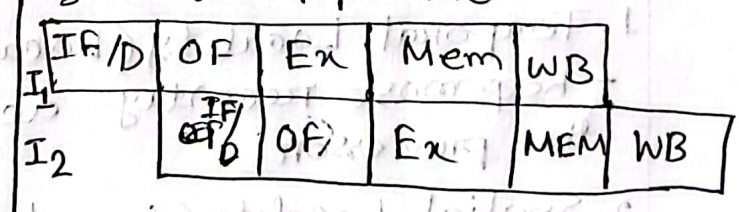
Example:

RAW →

$I_1 \rightarrow R_2 \leftarrow (R_2) + R_3$ (Write)

$I_2 \rightarrow R_5 \leftarrow (R_2) + R_4$ (Read)

5 stages pipeline



~~I1 এর result WB করার আগেই I2 তে~~ এসে

I_1 এর R_2 এর result আসবে WB করার পর। কিন্তু pipeline এ যেহেতু multiple instruction কাজ করে, তাই I_2 তে যখন R_2 কে Fetch করতে যাবে তখন সঠিক value পাবে না কারণ R_2 5th clock cycle এ final result দিবে। কিন্তু I_2 3rd clock cycle এ R_2 কে fetch করার try করবে এবং পরের stage এ execute করবে, যার কারণে ভুল result আসবে। এটিই data hazard এর RAW problem.