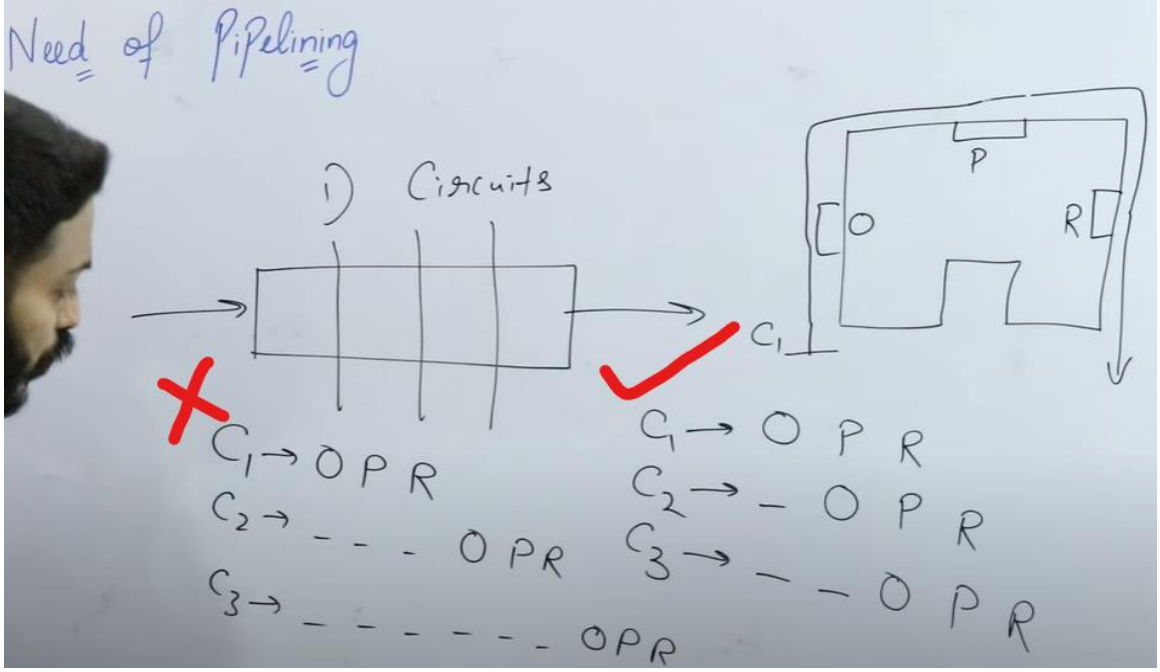
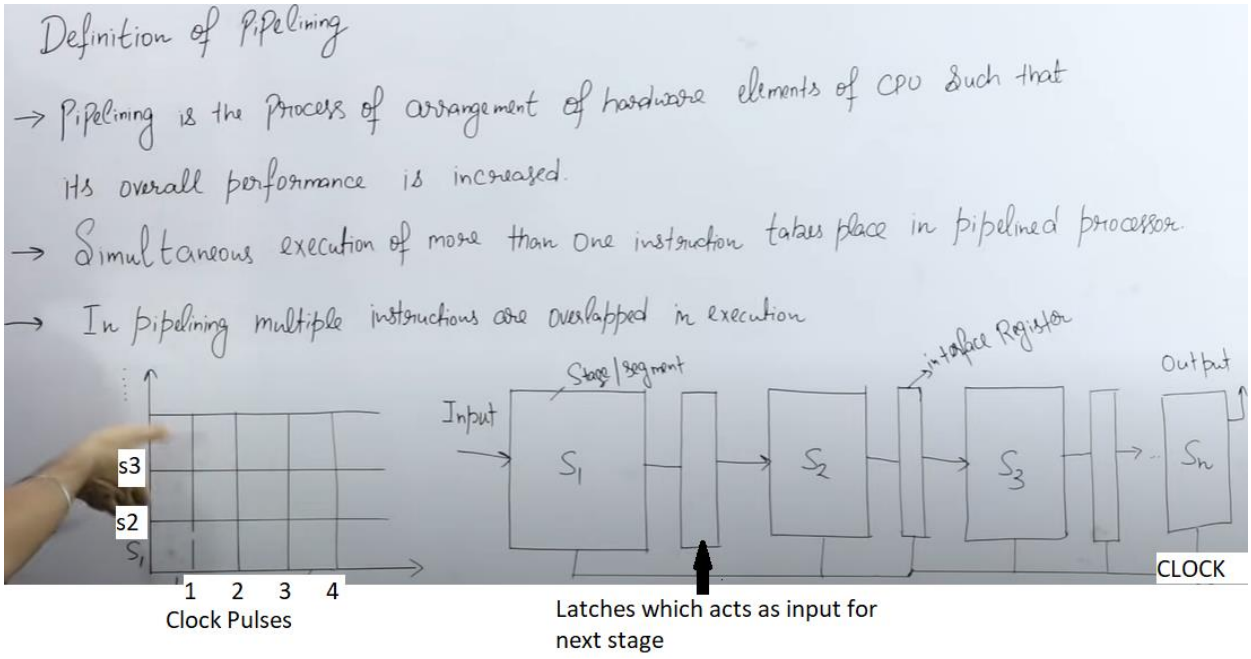
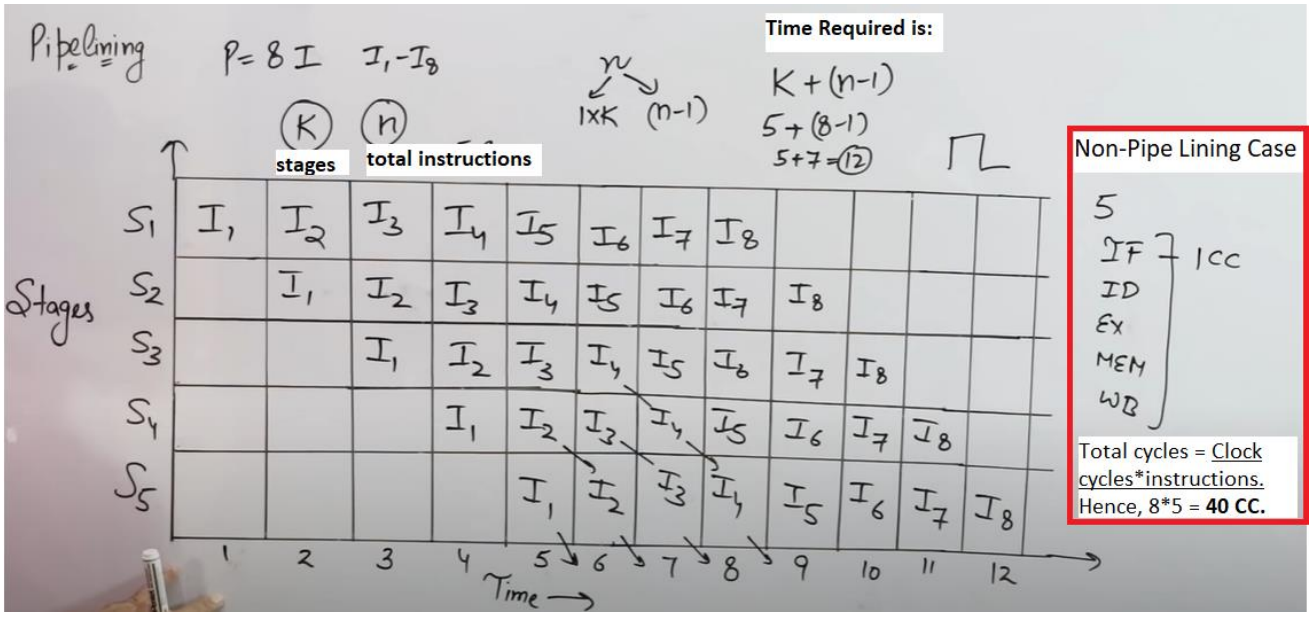




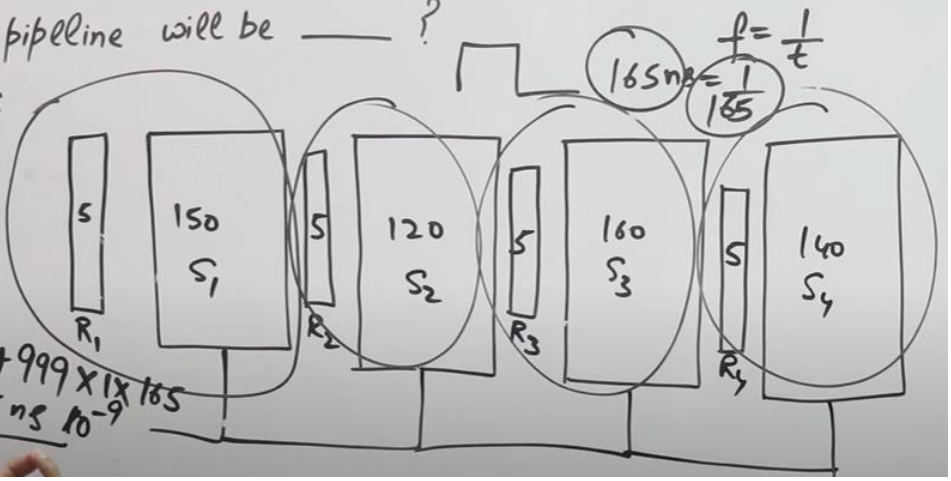
First Instruction's time = number of Stages (**K**). After that each instruction takes 1 unit.

Here, **CPI is nearly 1** (Clock Per Cycle) and Clock Cycle = **k + (n-1).**
Speed Up = non-Pipeline time /Pipeline time.



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

a) 120.4 μsec
b) 160.5 μsec
c) 165.5 μsec
d) 590 μsec.



$$= \frac{1 \times 4 \times 165 + 999 \times 1 \times 165}{165495 \text{ ns}} \; 10^{-9}$$

Consider a non pipelined processor with a clock rate of 2.5 gigahertz and avg. cycles / Instruction of four. The same processor is upgraded to a pipelined processor with five stages, but due to internal pipeline delay, the clock speed is reduced to 2 gigahertz. Assume that there is no stall in pipeline. The speedup achieved in pipeline processor is ___ ?

a) 3.2
b) 3.0
c) 2.2
d) 2.0

$$\text{Speedup} = \frac{T_{wp}}{T_p} \qquad f = 2.5 \text{ GHz} \qquad T = \frac{1}{f}$$

$$T_{wp} = 4 \times \frac{1}{2.5 \times 10^9} \text{ sec} = \frac{4 \times 1}{2.5 \times 10^9} \text{ sec}$$

$$T_p = \frac{1}{2 \times 10^9} \text{ sec}$$

$$\frac{\frac{4 \times 1}{2.5 \times 10^9} \text{ sec}}{\frac{1}{2 \times 10^9} \text{ sec}} = \frac{4 \times 2}{2.5} = \frac{8}{2.5}$$

Hazards in Pipelining

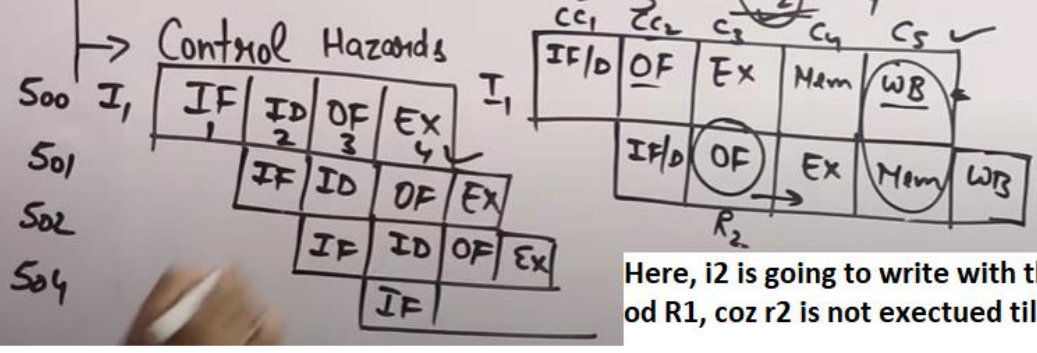
R = Read
W = Write

- Data Hazards
  - RAW (TD)
  - WAR (AD)
  - WAW (OD)
- Structural Hazards
- Control Hazards

CPI = 1

I_1 $R_2 \leftarrow R_2 + R_3$

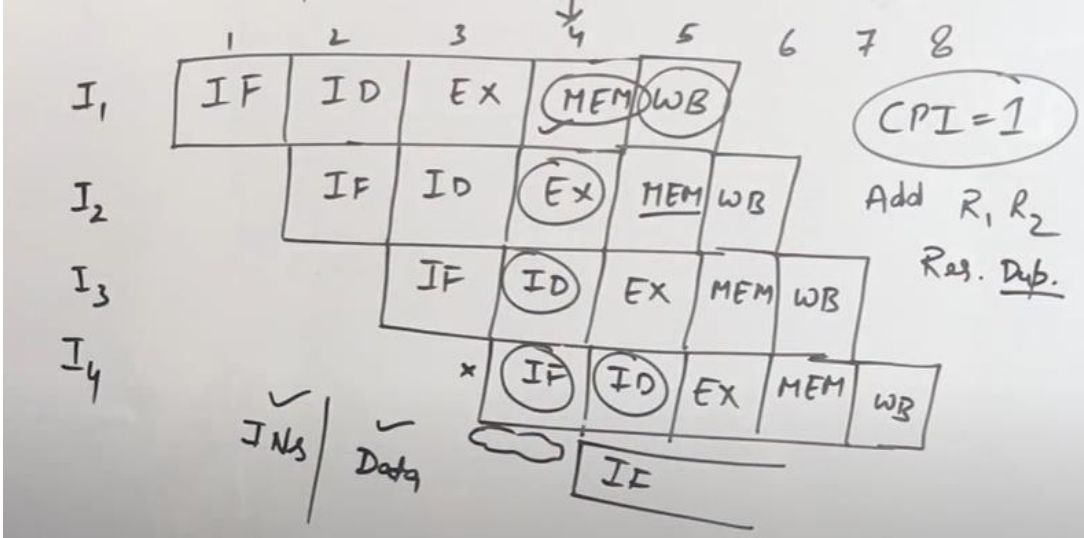
I_2 $R_5 \leftarrow R_2 + R_4$



Here, i2 is going to write with the old data od R1, coz r2 is not exectued till now.

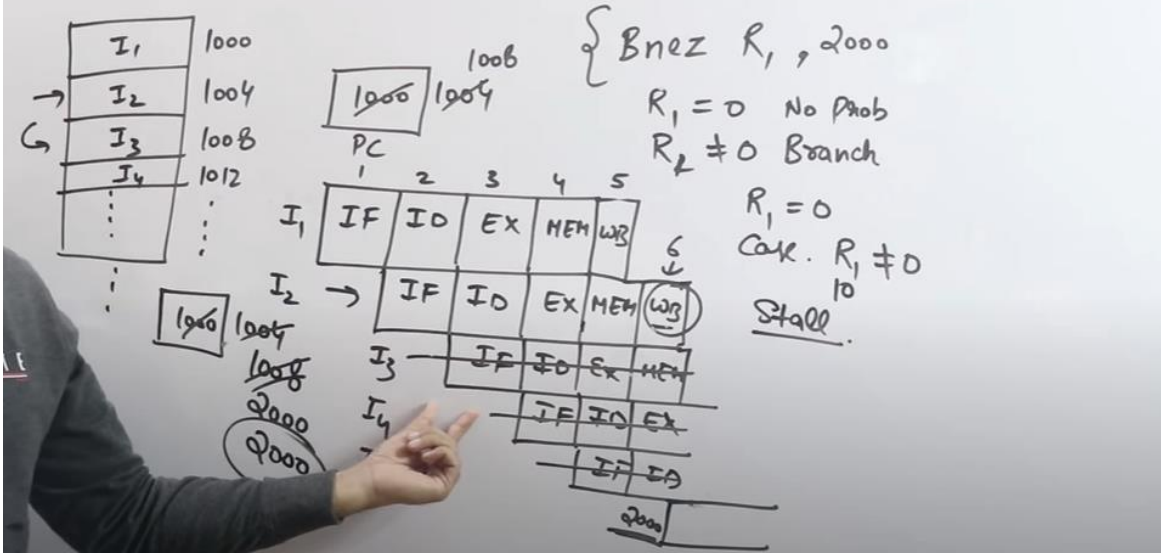
Structural Hazard

→ When multiple instructions needs same resource.



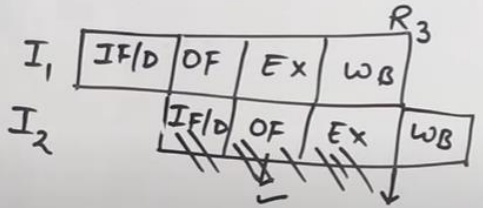
Control Hazard

→ All Instructions who change the program counter leads to control Hazards



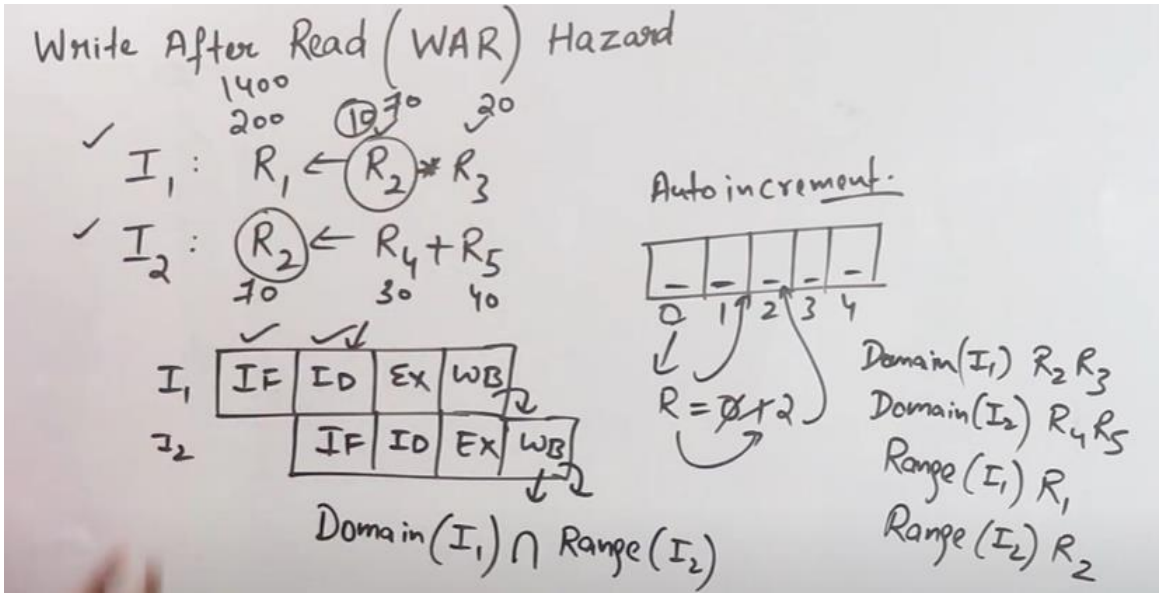
RAW (Read After Write) Hazard

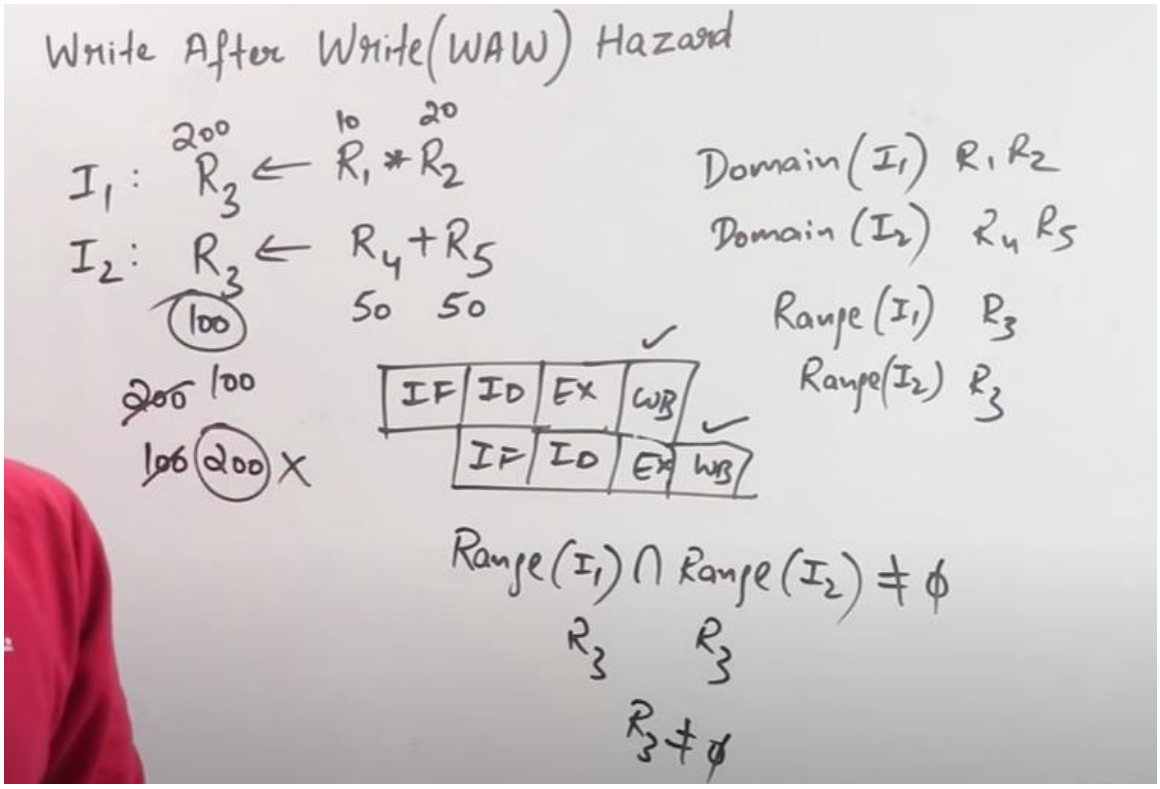
		Domain	Range
I_1	$R_3 \leftarrow R_1 + R_2$	R_1, R_2	R_3
I_2	$R_5 \leftarrow R_3 + R_4$	R_3, R_4	R_5



$I_1(\text{Range}) \cap I_2(\text{Domain}) \neq \phi$

RAW





Why we need interface?

1. To cope up with the speed of CPU and memory, coz we use I/O devices which is very slow comparatively.
2. For the conversion of signals b/w I/O and CPU.
3. For translator or interpreter b/w I/O and CPU.
4. For the conversion b/w data format (e.g. I/O is 8 bits and CPU is 64 bit).

Daisy Chaining Interrupt and Parallel Priority Interrupt:
https://upscfever.com/upsc-fever/en/gatecse/en-gatecse-chp165.html

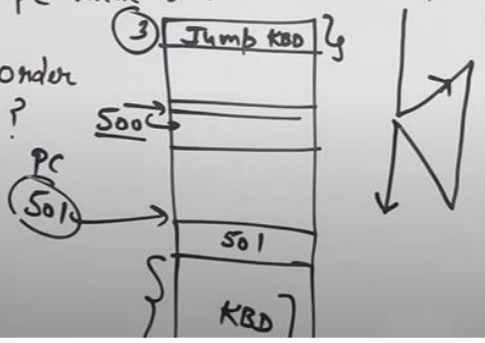
The following are some events that occurs after a device Controller issues an interrupt while process L is under execution.

P) The processor pushes the process status of L onto control stack.
Q) The processor finishes the execution of current instruction.
R) The processor executes the interrupt service routine.
S) The processor pops the process status of L from control stack.
T) The processor loads the new PC value based on interrupt.

Which of the following is correct order in which the above events occur ?

A) QPTRS
B) PTRSQ
C) TRPQS
D) QTPRS

Hence,
A) QPTRS
is the correct order of execution



"CISC"

1) Complex Instruction Set Computer
2) Large Number of Instructions
3) Variable length Instruction format
4) Large No. of addressing modes
5) Cost is High
6) More Powerful
7) Several Cycle Instructions
8) Manipulation directly in Memory
9) Microprogrammed Control Unit
10) Examples: Mainframes, Motorola 6800, Intel 8080
MULT 2:2, 3:3

"RISC"

1) Reduced Instruction Set Computer
2) Less No. of Instructions
3) Fixed length Instruction format
4) Few no. of AM
5) Less cost
6) Less Powerful
7) Single Cycle Instructions
8) Only in Registers
9) Hard wired Control Unit
10) MIPS, ARM, SPARC, Fugaku
LOAD A, 2:2
LOAD B, 3:3
PROD A, B