Instruction Pipelining

Instruction Pipelining: Prefetching the next instruction in advance while execution of current instruction is going on is called as instruction pipelining.

The execution of any instruction is not carried out in a single stage, but requires no. of stages.

The various stages of instruction pipe are

1] Fetch the Instruction [FI]
2] Decode the Instruction [DI]
3] Fetch the Operand [FO] (It is optional)
4] Execute the Instruction [EX]
5] Write Back or Store the Result [WB]

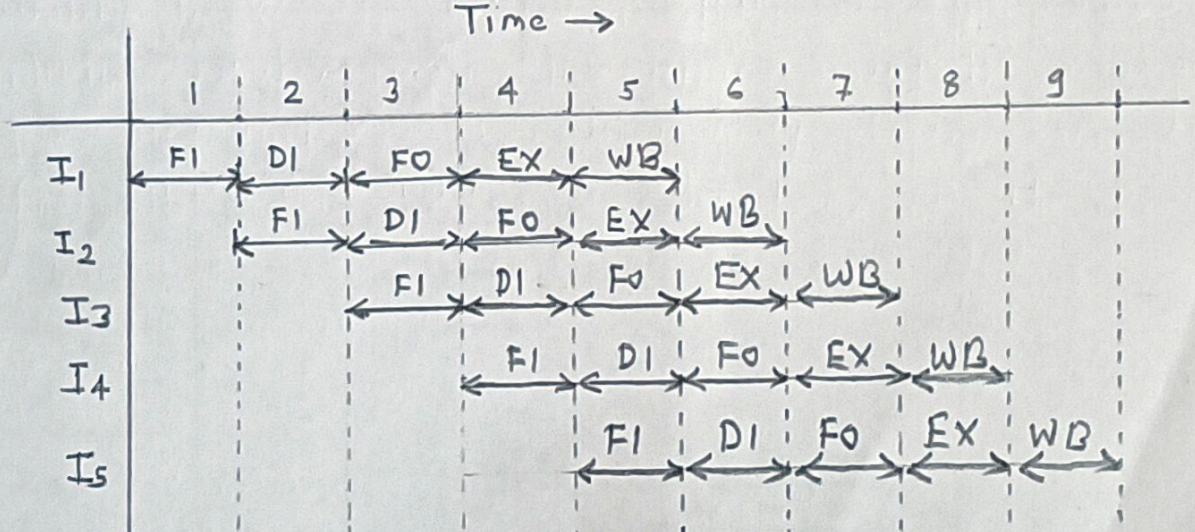
Effect of Instruction pipelining on performance of Computer

A] Performance Without Pipeline

Consider a program of 5 lines which is to be executed. Assume that each line requires equal amount of time i.e 5 seconds for execution. Thus for execution of this 5 line program without pipeline total time required for execution will be $5 \times 5 = 25$ seconds.

B] Performance with Pipeline:

Consider the same program of 5 lines & each line is requiring 5 seconds for execution. Each line requires 5 seconds & execution

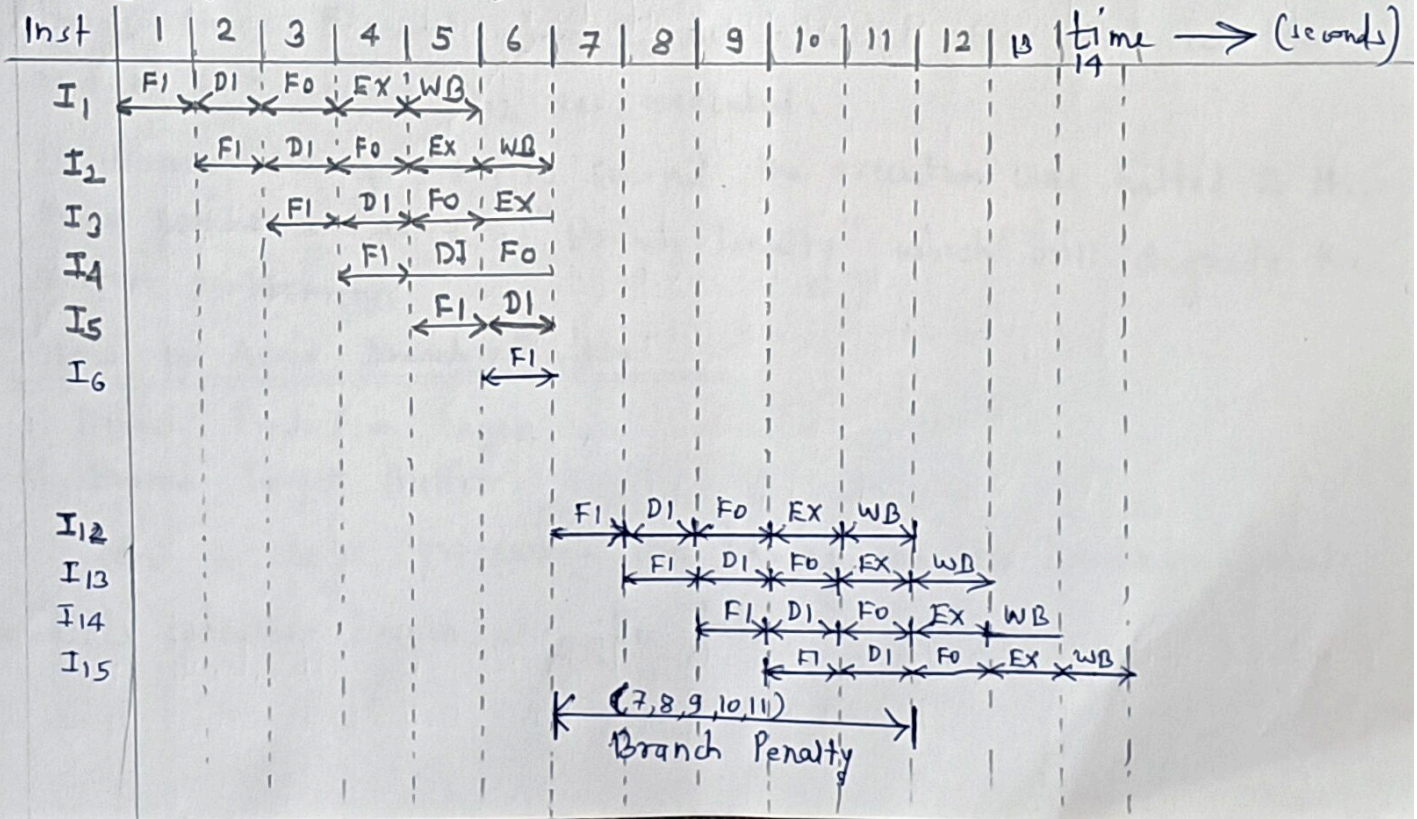


of Each line is carried out in 5 stages (FI, DI, FO, EX, WB) & it is assumed that each stage requires 1 seconds.

During 1st Second, Instruction 1 (I_1) will be taken inside pipeline
During 2nd Second Instruction 2 (I_2) will taken inside & I_1 will go to next stage i.e DI.
During 3rd Second I_1 will move FO (3) stage, I_2 will move to DI (2) stage, at the same time I_3 will be accepted in & it will continue
At the end of 5th second the first inst. (I_1) will get completely executed. At the end of 6th second I_2 will get completely executed & At the end of 9th second I_5 means last line of the program will get executed.
Thus the entire program of 5 lines will get executed in only 9 secs which was earlier taking 25 seconds to execute (without pipelining)
Thus by using instruction pipelining we are saving 25 − 9 = 16 Secs. which is huge amount of time & thus perfomance will improved tremendously by using concept of Instruction Pipelining.

Effect of Branch Instruction on Performance of Inst. Pipeline

No doubt perfomance is improved by use of instruction pipelining, but if there is a branch instruction in between in the program then the perfomance is degraded as shown below. Consider the similar situation. Each line requires 5 seconds. [within one line each stage require 1 seconds] Now the program is of 15 lines & I_5 line is Branch inst. (Jump at line No. 12)



As shown in the diagram

1) During 1st second I_1 will be taken inside
2) During 2nd second I_1 will go to DI stage & I_2 will enter inside
3) This process will continue and at the end of 5th second, the first line I_1 will get completely executed.
4) During 6th second (if we see vertically in colmn 6), I_2 is completely executed, I_3 is in EX stage, I_4 is in FO stage, I_5 is DI stage & I_6 is just taken inside.
5) I_5 is in DI stage & when I_5 decoded, it meaning is jump at instruction No. 12 (I_{12}) by bypassing instructions in between.
7) Thus I_3, I_4, I_5, I_6 will remain pending & processor will start executing instructions from I_{12} onwards.
8) I_{12} will get executed at the end of 11th second & I_{15} the last line I_{15} will get executed in 14th second.
9) Normally in pipeline system we have to just wait for first 5 seconds (pipeline gets full) & after that for every second there is execution of one instruction continuously.
10) But In this case the I_2 was the instruction which was completely executed at the end of 6th sec. & after that I_3, I_4, I_5, I_6 were kept pending & instead of that I_{12} was taken inside, then I_{13}, I_{14}, I_{15}
11) Finally I_{12} was completely executed at the end of 11th seconds.
12) Thus after 6th second where I_2 was executed, after that at the end of 11th second I_{12} was executed.

It means during 7, 8, 9, 10 second the execution was halted & this time period is called as "Branch Penalty" which will degrade the system performance.

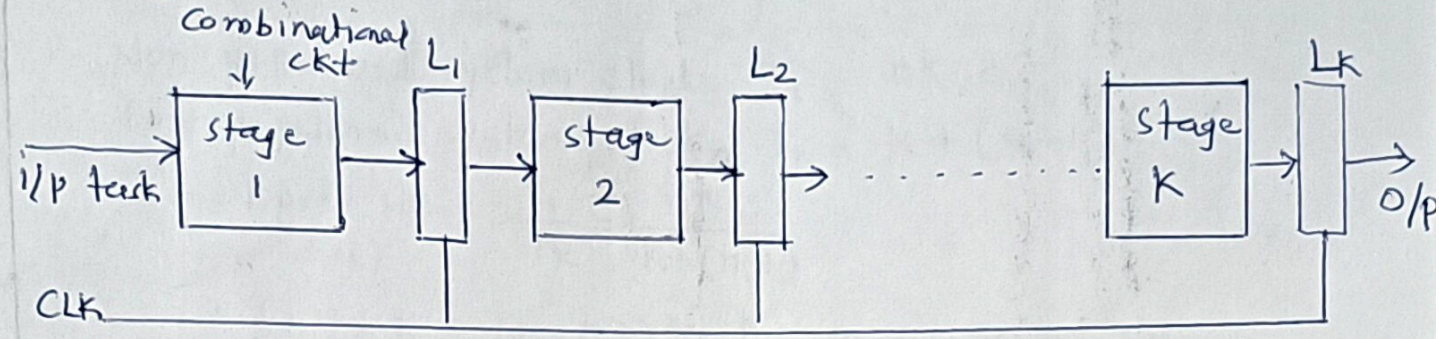
How to Avoid Branch Penalty

1] Branch Prediction Logic
2] Branch Target Buffer

Now a days processors are having inbuilt hardware that predicts wheather branch will be taken or not.

Performance of a linear pipeline

* Linear pipeline is designed by using a number of pipeline stages & each of these pipeline stage is purely a combinational circuit.
* Two successive stages in a pipeline are seperated by using a high speed latch.



* Result generated by stage 1 is copied to the Latch & it goes for processing the next job

* Performance of a linear pipeline is measured in terms of
  1) Clock frequency / clock period
  2) speed up
  3) Efficiency of pipeline
  4) Throughput of pipeline

Clock frequency / Clock Period:

The clock frequency of linear pipeline is

$$T = \max_{i=1}^{K} \{t_i\} + t_L$$

where t_L = Latch period.

While defining clock frequency of pipeline maximum time delay is taken into account as per the equation

Reciprocal of clock period gives clock frequency of pipeline

$$\text{Clock frequency of pipeline} = \frac{1}{\text{Time Period } (T) \text{ of pipeline}}$$

Speed up:

Ratio of time required for 'n' tasks on a non pipelined system to the time required on a pipelined system is called speed up.

No. of tasks = n
No of pipeline stages = K

∴ Non pipelined system clocks $T_1 = nK$
∴ for pipelined system clocks $T_K = K + (n-1)$

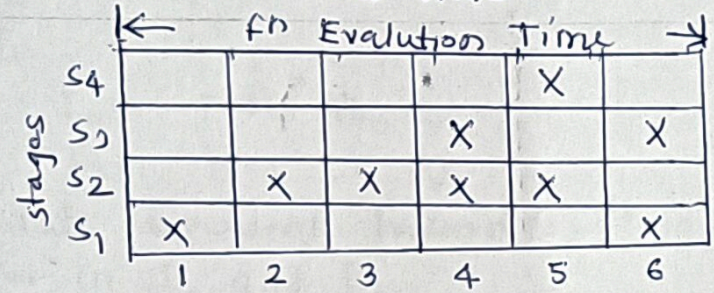
∴ Speed up $(S_K) = \frac{T_1}{T_K} = \frac{nK}{K+(n-1)}$

$S_K \to K$ for $n \gg K$ for maximum speed up.

Efficiency:

It is the fraction of busy time in the total time

$$\eta_{(E)} = \frac{\text{Busy Time}}{\text{Total Time}}$$



$$\eta_{(E)} = \frac{9}{24}$$

OR

Efficiency is speed up in the pipeline system due to 1 stage

$$\eta_{(E)} = \frac{S_K}{K} = \frac{nK}{K+(n-1)} \times \frac{1}{K} = \frac{n}{K+(n-1)}$$

$\eta_{(E)} \to 1$ as $n \to \infty$

Throughput:

It is the average number of results generated per unit time.

If No. of tasks are = n
No. of stage are = K

∴ No. of clocks required = K+(n-1)
∴ Total time = [K+(n-1)]T
∴ Throughput = $\frac{n}{[K+(n-1)T]}$

∴ Throughput = $\frac{\eta_{(E)}}{T}$

Pipeline Hazards

There are two types of pipeline hazards
1] Data dependant Hazards
2] Control Hazards.

Data Dependant Hazards

These hazards are dw to inter instruction data dependency Broadly, data dependent hazards are caused by resource usage conflicts among successive instructions

Normally such conflict is due to common register between two instructions.

In case of data dependent hazards unless the earlier instruction goes beyound the possible resource conflict stage, the next or subsequent instruction is not allowed in the processing stage.

It means 2nd instruction can be fetched as well as decoded but execution is delayed untill first instruction gets totally executed

In 8086 Processor

I_1 : MOV AX, BX
I_2 : MOV CX, AX

Here data dependent hazard is triggered due to use of AX register in I_1 and I_2

In case of data dependent hazards, linear operation of pipeline is sometimes suspended.

There are three types of data dependent hazards
1] Read After Write (RAW)
2] Write After Read (WAR)
3] Write After Write (WAW)

* RAW Hazard

In 8086 Processor

I_1 : ADD AX, BX : Write
I_2 : SUB CX, AX : Read

with respect to AX register data dependent hazard is triggered in RAW

A RAW Hazard may occur when instruction 'J' attempts to read some data object which has been modified by previous instruction 'I'

D(I) defines the domain of Instruction I i.e. a set of resource data objects that may affect the execution of Inst 'I'

R(I) defines the range of instruction I i.e a set of resource data objects that can be modified by execution of Instruction 'I'

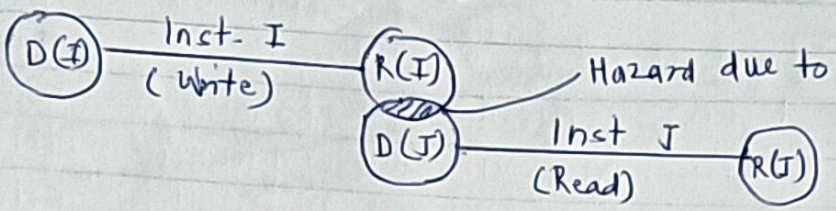


Illustration of RAW hazard condition

D(I) & D(J) define domain of instruction I & J respectively. (Domain is input)

R(I) & R(J) defines range of instruction I & J respectively (Range is output)

* WAR Hazard :-

In 8086 system

I_1 : ADD AX, BX : Read

I_2 : SUB BX, CX : Write

With respect to BX Register data dependency hazard is triggered in WAR

A WAR hazard can occur if instruction 'J' attempt to modify some data object which is read by instruction 'I'

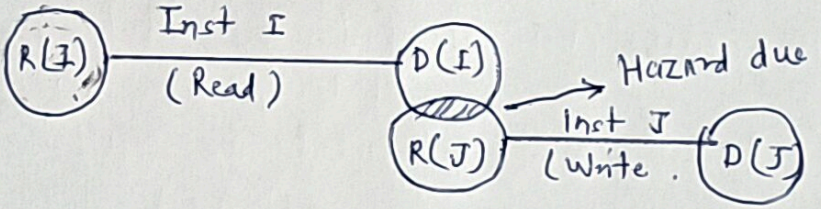


Illustration of WAR Hazard Condition

* WAW Hazard :-

In 8086 system

I_1 : MOV AX, BX : Write

MOV AX, CX : Write

With respect to AX register data dependent hazard is triggered in WAW

In WAW hazard can occur when both I & J instruction modify the same data object.

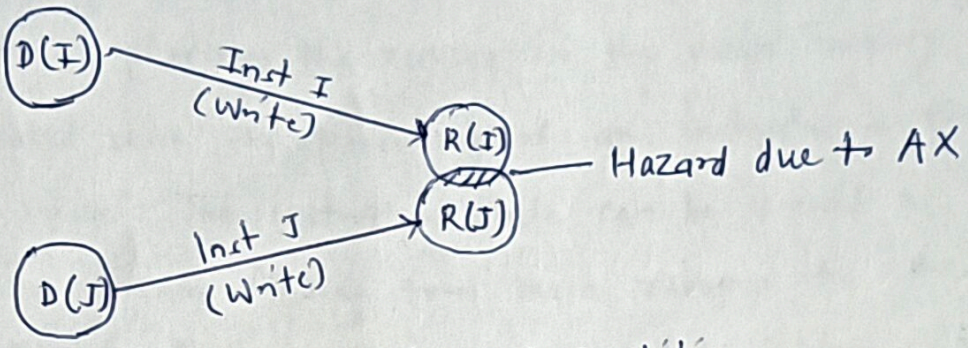


Illustration of WAW Hazard Condition

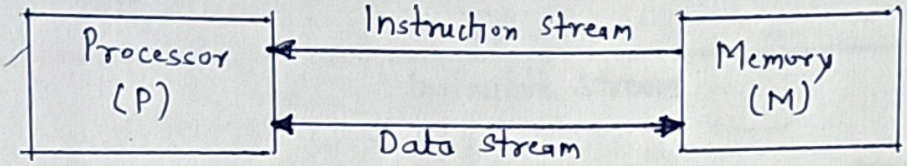
Control Hazards :

Control hazards are caused due to execution of

i] Branch Instruction
ii] Subroutine CALL instruction or
iii] In processing of Interrupts.

FLYNN's Classification of Parallel Processing Systems.

A typical central processing unit (CPU) operates by fetching instructions & operands from main memory, executing the the instructions & placing the results in the main memory. The steps associated with the processing of an instructiom form an "Instruction cycle". The instructim cycle can be viewed as forming an instructim stream flowing from main memory to the processor while the operands form another stream i.e data stream flowing to or from the processor as shown.



Instruction & Data Streams in a simple computer.

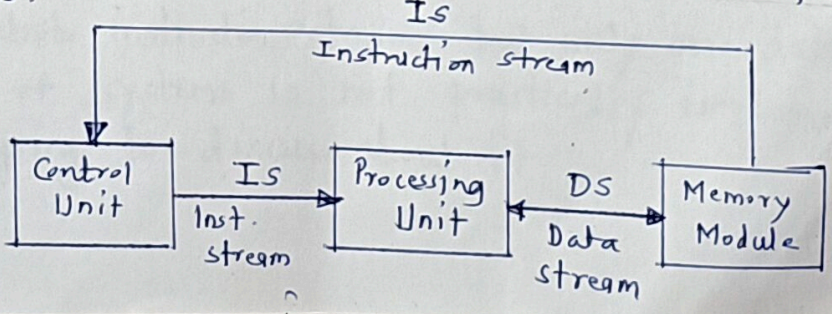
Based on the number of instructions streams & number of data data streams present Michael Flynn's has divided computers into 4 groups.

1] Single Instruction Stream - single data stream [SISD]
2] Single Instructim Stream - Multiple Data Streams [SIMD]
3] Multiple Instruction Stream - Single Data Stream [MISD]
4] Multiple Instruction Stream - Multiple Data Stream [MIMD]

(1) SISD

In these systems there is only one instructim stream & only one data stream.

Most conventional machines with only one CPU containing a single arithmatic & Logic unit (ALU) capable of only of scalar arithmatic fall into this category. SISD computers & sequential computers are thus synonymous.

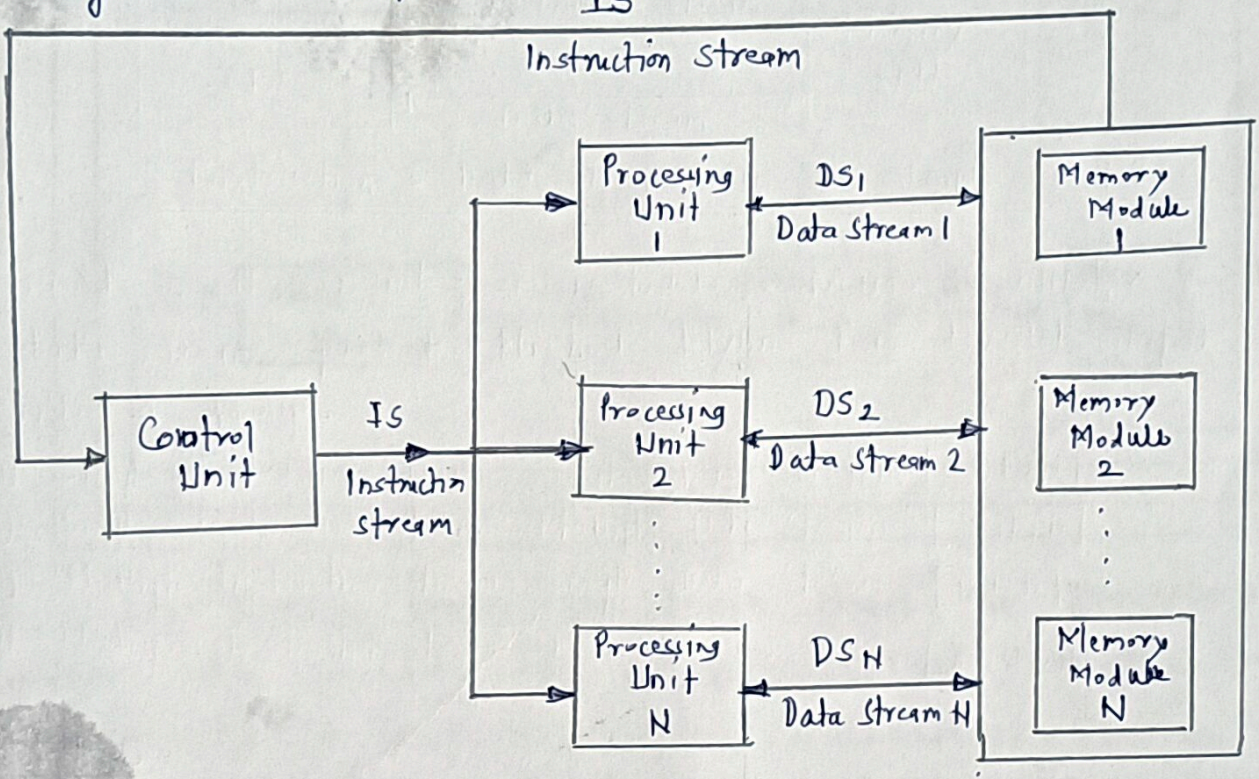


In SISD computers instructions are executed sequentially but may overlap in their execution stages (Pipelining). They may have more than one functional unit but all the functional units are controlled by a single Control Unit.

SIMD

They have single Instruction stream but Multiple Data streams. This category of computers corresponds to "Array Processors" They have multiple processing / execution units but only one control unit.

Therefore, all processing / execution units are supervised by the single control unit.



Here, all the processing units are receiving the same instruction from the control unit.

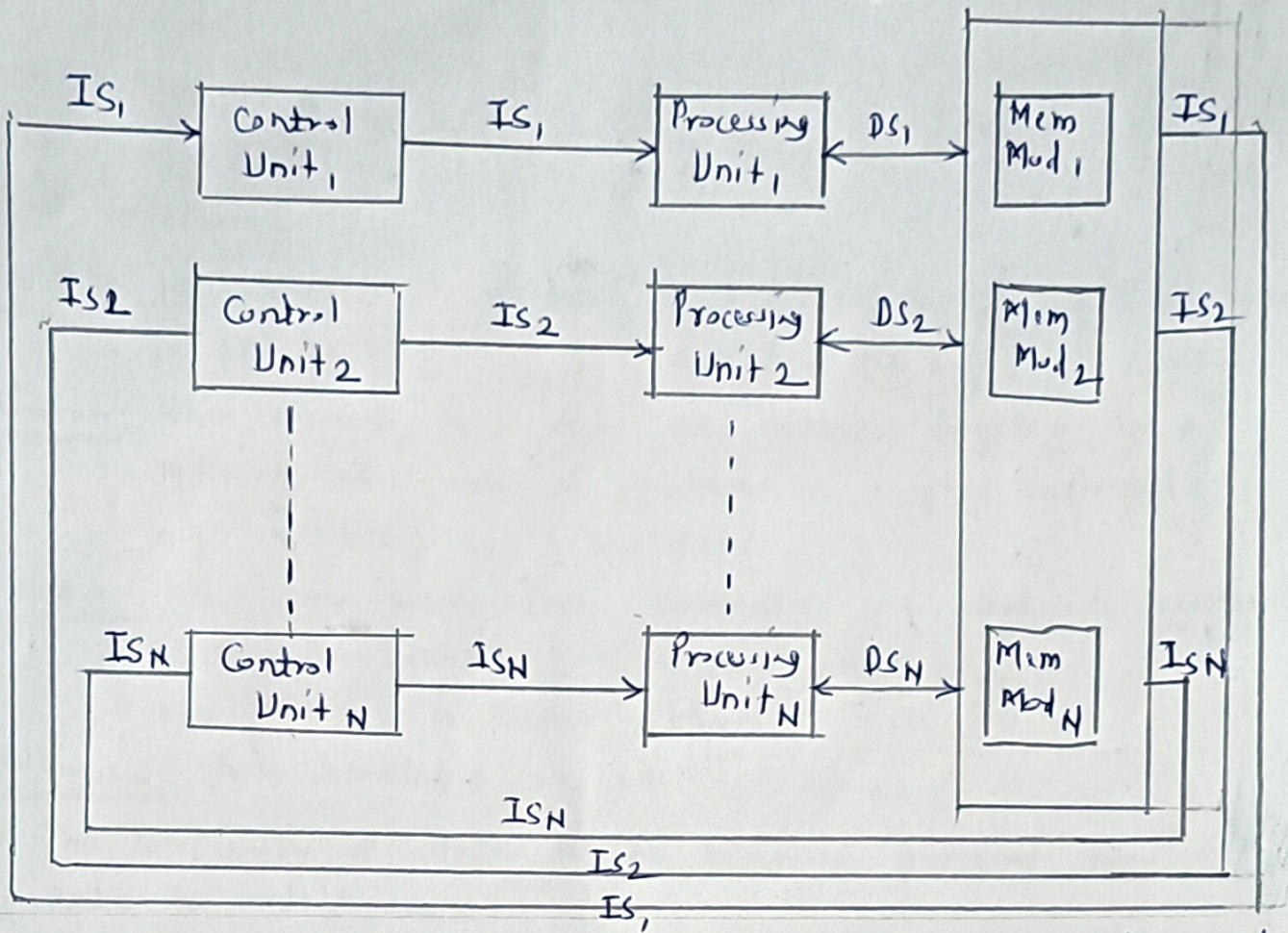
But each processing unit will operate on different data sets from distinct data streams.

MISD

They have multiple instruction streams but only one data streams This category of systems is not practically implemented & we are not going to discuss about it.

MIMD :

They have multiple instruction streams & multiple data streams. Most multiprocessor system & multi-computer system can be classified in this category.



In MIMD, there are more than one processing unit having the ability to execute several programs simultaneously. MIMD Computer implies interactions among the multiple processing units because all memory streams are derived from the same data space shared by all the processors.

Parallel Processing & its Applications

Parallel Processing: It is an efficient form of information processing which emplies exploitation of concurrent events in computing process.

Parrellel Processing is applicable to –

i) Data
ii) Information
iii) Knowledge
iv) Intelligence.

Data: It is collection of alphabets, newmericals or special characters
e.g. { P, N, E } { S, 4, H, 2, R, *, O, I }

Information: When elements from 'data' are arranged together by applying some rules of grammer it becomes information.
e.g. { PEN } or { SHRL420 }

Knowledge: When some words from information are combind together to make a seentence, it becomes information.
e.g. PEN is of BLACK COlour.

Intedligence: from knowledge we get intelligence.

The complexity of data to be processed increases from data to intelligence.

Applications of Parallel Processing:

(A) Modelling & Simulation

i) Weather Modelling
ii) Ocenography & Astrophysics
iii) Socioeconomics & Goverment Use

1) Weather modelling:
It is used for i) Short range forecast of weather
ii) Long Range hazard predictions (floods / T-sunami / Earth quake)

In India weather modelling is carried out by using 'PARAM' series of supercomputers designed at C-DAC (Center for Development of Advanced Computing). A series of super computers models like PARAM 8000, PARAM 9000, PARAM 10000 is developed by C-DAC.

[II] Oceanography & Astrophysics

It is used for climate Prediction over ocean & more specifically for 4 application listed below.

i] Climate Predictive Analysis

ii] Fishery Management

iii] Ocean Resource Exploitation

iv] Coastal Dynamics & Tides

The computers or systems which are used for implementing these applications are "Cyber-205" & "ILLIAC-IV" array processors.

[III] Socioeconomics & Govt. Use

It is used for

i] Econometrics

ii] Social Engineering

iii] Govt. Census / Public Opinin Polls

iv] Crime Control & Investigation

v] Modelling of National / World Economy.

These applications are implemented using CDC Scientific computers.

CDC → Control Data Corporation.

(B) Engineering Design & Automation

It is used in

① Finite Element Analysis: FEA is used for the design of huge structures like dams, bridges, ships, supersonic jet engins etc.

* The designing of such things needs a sequence of differential equations to be solved
* On a traditional / conventional machine to solve those differential equations it takes longer time. Hence parallel structures like array processor or Vector processor are used to reduce the processing time.
- systems which are implementing such type of applications are
  CDC Star-100 Vector Processor
  Cyber-205 Vector Processor.

② Computational Aerodynamics

It is used for designing of aerodynamic structures like JET engines.

System Design for Computational Aerodynamics i.e (NASF) Numerical Aerodynamic Simulation Facility is used.

It is developed by Burrough's Corporation & CDC

It is used for simulation of complete aircraft design.

③ AI & Automation:

It is used for

1] Image Processing
2] Pattern Recognition
3] Computer Vision
4] Speech Understanding
5] Machine Inference
6] CAD - Computer Aided Design
7] CAM - Computer Aided Manufacturing
8] office automation
9] Design of intelligent Robots
10] Design of Expert Systems.

In all the applications stated above a system should be developed with maximum LIPS (Logical Inferences Per Second)

One such system is CRAY-1 from CRAY Research Laboratory. To carry out these applications the processing power required is 100 MLIPS to 1 GLIPS

④ Remote Sensing Applications

* Remotely Sensed Earth resource data (via Satellite) is analysed by using parallel structures like an array processor MPP (Massively Parallel Processor)
* This analysis is helpful for forestry, Agricultural use & for water resource exploitation.

⑤ Medical, Military & Research Applications

It is used in

1] Computer Assisted Tomography
2] Genetic Engineering
3] Weapon Research & Defence

CISC & RISC Computers

* CISC Architecture:

* The Intel 80X86 and Motorola's 680X0 processors are the most commonly used CISC processors.
* They provide complex instruction sets having powerful instructions for direct implementation of high level language operations and program sequencing control features
* Execution of such instructions is quite complex and it is usually implemented by microprogrammed control.
* CISC architecture provides only few on chip registers & therefore most of the instructions have to refer main memory.
* This increases the time required for execution of instructions.
* As CISC architecture provides large number of instructions, they also provide many complex addressing modes & hence resulting into complex control unit.

* RISC Architecture:

* RISC is an acronym for "Reduced Instruction Set Computer"
* Today's processor such as Intel i860, SPARC, MIPS R3000 are examples of RISC architecture
* In most of RISC processors they use hardwired control unit.
* In RISC architecture it provides very few instructions & most of the instructions are 32 bit instructims.
* As there are less number of instructions, there are only few addressing modes.
* As the instructims are less & and they are simple in nature, this architecture requires simple decoding circuitary.
* It provides large number of on chip registers & hence the processor need not refer main memory for most of the instructions.

Thus because of simple instructions & less number of inst. & large number of internal registers RISC architecture is much more popular now a days.

Difference between CISC & RISC Computers

RISC	CISC
i) Simple instructions requiring only one cycle for executions	i) Complex instructions requiring multiple cycles for execution
ii) Very few Instructions	ii) Large number of instructions
iii) Fixed instruction formats	iii) Variable format instructions
iv) Verf few instructions refer main memory.	iv) Most of the instructions refer main memory
v) Instructions are executed by hardware	v) Instructions are executed by microprogram.
vi) Multiple register sets are available	vi) A single register set is used (only few registers)
vii) Very few addressing modes are used.	vii) Many addressing modes are used.
viii) Decoding circuitary is simple	viii) Decoding unit is more complex
ix) Highly pipelined.	ix) Not pipelined or less pipelined.
x) Complexity is in the compilier	x) Complexity is in microprogram

Concurrent Accesses to Memory

Memory Contention

A memory module can handle only one access request at a time. Hence when several processors request the same memory module, it causes memory contention.

This contention is resolved by an arbiter permitting the requesters to proceed one at a time. However the processors that wait to complete their memory cycle waste their time in busy waiting. This results in loss of performance

Communication Contention:

When several requesting processors can not complete their access due to the limitation of the interconnection network, the contention occurs and such contention is called communication contention.

Hot Spot Contention:

When several processors repeatedly access the same memory location, it gives rise to hot spot contention.

This is because it creats hot spot in either the memory module or the interconnection network that get overloaded with the requests and creat a bottleneck for the system performance.

Techniques for Reducing Contention

1] Local Memories
2] Better Interconnection network
3] Cache memory
4] Memory allocation.