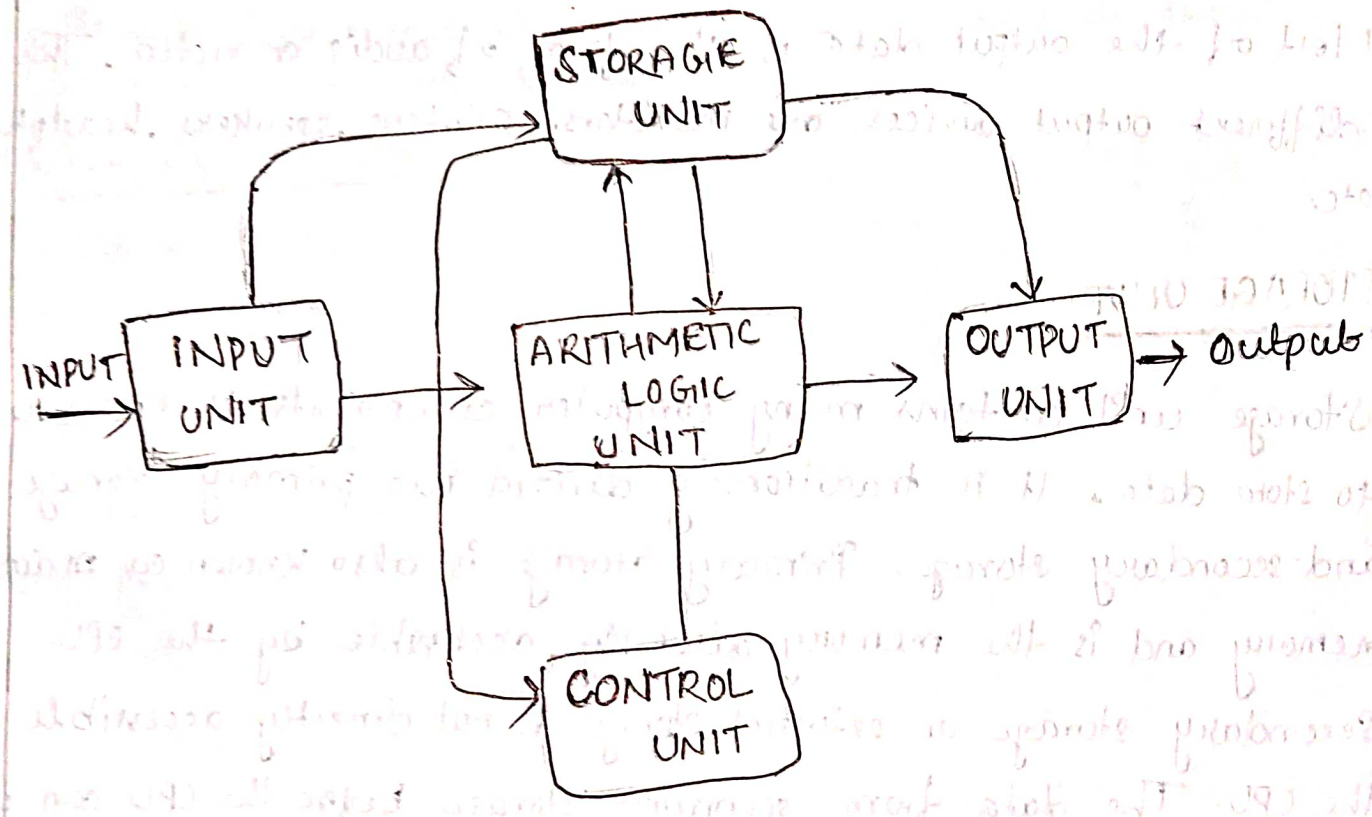
BASIC STRUCTURE OF COMPUTER HARDWARE

Dt-21.10.2021

A computer system is basically a machine that simplifies complicated tasks. It should maximize performance and reduce costs as well as power consumption. The different components in the computer system Arti Architecture are input unit, Output unit, storage unit, Asithmetic Logic Unit, control unit etc.

A diagram that shows the flow of data between these units is as follows:-



The input data travels from input to ALU. Similarly, the computed data travels from ALU to output unit. The data constantly moves from storage unit to ALU and back again. This is because stored data is computed on before being stored again. The control unit controls all the other unit as well as their data.

INPUT UNIT

The input unit provides data to the computer system from the outside. So, basically it links the external environment with the computer. It takes data from the input devices, converts it into machines language and then loads it into computer system. Keyboard mouse etc, are the most commonly used input devices.

OUTPUT UNIT

The output unit provides the results of computer process to the users is it links the computer with the computer environment. Most of the output data is the form of audio or video. The different output devices are monitors, printers, speakers, headphones etc.

STORAGE UNIT

Storage unit contains many computer components that are used to store data. It is traditionally divided into primary storage and secondary storage. Primary storage is also known as main memory and is the memory directly accessible by the CPU. Secondary storage or external storage is not directly accessible by the CPU. The data from secondary storage before the CPU can use it. Secondary storage contains a large amount of data permanently.

ARITHMETIC LOGIC UNIT

All the calculations related to the computer systems are performed by the arithmetic logic unit. It can perform operations like additions, substraction, multiplication, division etc. The control

unit transfers data from storage unit to arithmetic logic unit when calculations needs to be performed. The arithmetic logic unit and the control unit together form the central processing unit.

CONTROL UNIT

The unit controls all the other unit of the computer system and so is known as its central nervous system. It transfers data throughout the computer as required including from storage unit to central processing unit and vice versa. The control unit also dedicates how the memory, input, output devices, arithmetic logic unit etc. should behave. (sequencly execution of instruction is done by control unit).

Size of Memory

Dt = 2.11.21

1 byte = 8 bit

1Kb = 1024 byte

1mb = 1024 kb

1mb = 1024 x 1024 byte

1gb = 1024 mb

1Gb = 1024 x 1024 ~~x 1024~~ kb

1Gb = 1024 x 1024 x 1024 byte

1Gb = 1024 x 1024 x 1024 x 8 bit

* There are more than one register present in C.P.U or processor

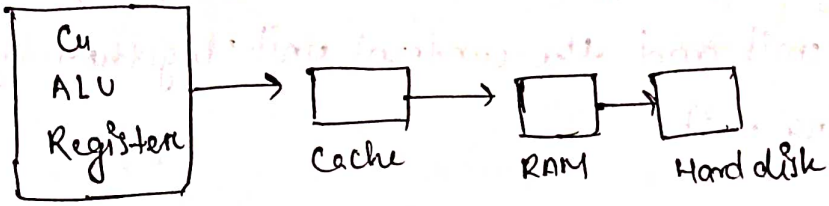
Register = It is smallest memory located in CPU

It is smallest capacity memory stored intermediate results.

Dt = 09.11.21

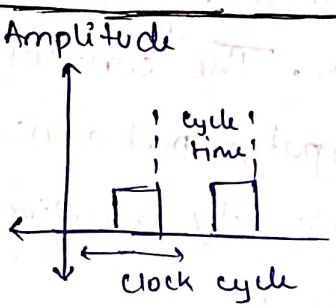
Intermediate Results

a = 5
c = a+b = (11)
d = c+a = (16)



To enhance the process of data is also known as intermediate results.

Performance Measure :-



1f = Frequency / clock rate
= No. of clock cycle per second

→ How many cycles are there within a particular time interval

CC = cycle count
= Time interval
CT = Time period / cycle Time
= $\frac{1}{F}$

CPI = Cycle per instruction
IC = Instruction count
MFLOP = Million Floating Point Instruction Per Second
MIPS = Million Instruction Per Second

Execution Time = CC x CT (cycle count x cycle time)
= IC x CPI (Instruction count x cycle per instruction)

$$IC \times CPI \times \frac{1}{F}$$

To find CPI: $\frac{\text{Total CPU clock cycle for the Program}}{\text{Instruction count}}$

$$= \boxed{\sum_{i=1}^{n} \frac{CPI_i \times I_i}{I_c}}$$

$$MFLOP = \frac{\text{no. of floating point operation in a program}}{\text{Execution time} \times 10^6}$$

* 10^6 = million (mega is known as 10^6)

$$MIPS = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6}$$

→ If a computer A runs a program in 10sec & computer B runs the same program in 5sec than, How much faster is B than A.

Sol, $\text{Performance} = \frac{1}{\text{Execution time}}$

or

performance $\propto$ execution time

A = 10ses B = 5sec

Execution time of A $\boxed{E_A > E_B}$ then

Performance of A $\boxed{P_A < P_B}$

$$= \frac{P_A}{P_B} = \frac{\frac{1}{10}}{\frac{1}{15}}$$

$$= \frac{1}{10_2} \times 5$$

$$\frac{P_A}{P_B} = \frac{1}{2} \qquad \boxed{P_B = 2P_A}$$

Dt: 11.11.21

RAM (Random Access Memory)

RAM is a hardware element where the data being currently used is stored. It is a volatile memory. The data on the random access memory can be read, written and erased any ~~times~~ number of times.

Types of RAM :-

(i) Static RAM :- Stores a bit of data using the state of a six transistor memory cell. (SRAM)

(ii) Dynamic RAM :- Stores a bit data using a pair of transistor and capacitor which constitute a DRAM memory cell.

ROM (Read Only Memory)

ROM is a type of memory where the data has been prerecorded. Data stored in ROM is retained even after the computer is turned off or also known as non-volatile memory

Types OF ROM

(i) PROM → (Programmable Read only memory)
(ii) EPROM → (Erasable Programmable Read only memory)
(iii) EEPROM → (Electrically erasable programmable ROM)
(iv) UVEPROM → (Ultra-violet EPROM)

Difference between RAM and ROM

RAM	ROM
→ RAM is a volatile memory	→ ROM is a non-volatile memory
→ It could store the data as long as power is supplied	→ It could retain the data even when power is turned off
→ RAM can be retrived and altered	→ Data stored in ROM can only be read
→ It is a high speed memory	→ It is much slower than RAM
→ Larger in size with higher capacity and costlier	→ Small in size with less capacity and cheaper than RAM

	SRAM (Static RAM)	DRAM (Dynamic RAM)
i)	Stores information as long as power is supplied	Stores information only for few mili-seconds even when power is supplied.
ii)	It is made up off Flip-Flop and many transistors	It is made up off capacitor and Few transistors.
iii)	Bit stored in the form of voltage	Bit stored in the form of electric charge
iv)	It has no leakage property so does not need to refresh	It has charge leakage property so, it needs to be refreshed
v)	Refreshing circuit is not implemented	Refreshing circuit is implemented
vi)	For a single memory cell, six transistor are used	For a single memory cell, one transistor is used
vii)	Faster than DRAM	Slower than SRAM
viii)	More expensive than DRAM	Cheaper than SRAM
ix)	Size of memory cell is larger	Cheap size of memory cell is smaller
x)	It has low density (memory cell per area)	It has high density
xi)	It is used is cache memory	It is used in main or primary memory that is DDR, DDR$_2$, DDR$_3$ (DDR = double data rate ram)

MEMORY ADDRESSING

Two types

(i) Word addressing : Combination of several bit ~~several bytes~~ are known as word addressing

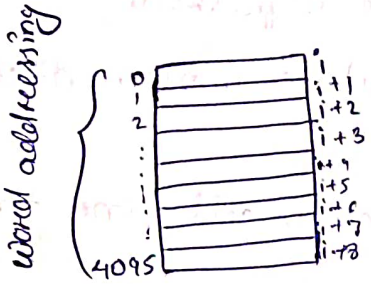
(ii) Byte addressing :

Word addressing

Ex → 4096 x 32

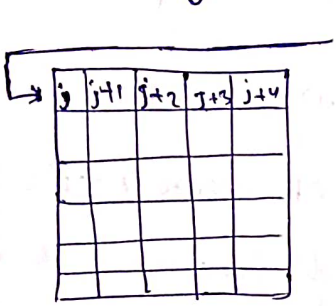
32 → No. of bytes/block

4096 → Size of block



Each cell contains one word

Byte addressing



i = j

i + 1 = j + 4

i + 2 = j + 8

4096 x 32 (4 byte, so, 4 block) word = 32 bit

4096 = 2^{12} → 12 bit needs to store) word addressing

(i) 4096 x 32

= 2^{12} x 4 byte

= 2^{12} x 2^2

= 2^{14}

(ii) 4096 x 16

= 2^{12} x 2 byte

= 2^{12} x 2 byte

= 2^{13}

(iii) 1024 x 32

= 2^{10} x 2^2

= 2^{12}

(iv) 256 x 16

= 2^8 x 2

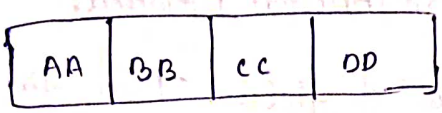
= 2^9 Ans

Dt = 17.11.21

ENDIAN

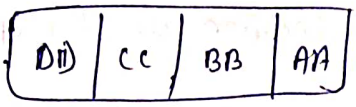
Two types

(i) Little Endian (INTEL PROCESSOR)



lower ——→ higher

(ii) Big Endian (MAC, AMD)



lower ——→ higher

Ex → start

int x = 0x AABBCCDD

least significant Byte (LSB) = DD

Maximum significat Byte (MSB) = AA

CHAPTER = 2 INSTRUCTIONS

Instruction :- Commands given by computer

Three types

(i) MODE → It specify whether the address is direct or indirect

(ii) OPCODE → Operation performed by ALU ~~for ex~~ or mathematica operation/expression performed known as OPCODE for ex- +, −, ÷, ×.

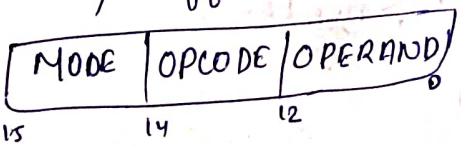
(iii) OPERAND → contains direct and indirect address.

→ direct address / indirect address

No fix location

Pointer : A variable which hold the address of another variable

Instruction contains - (i) Memory reference Instruction



COMPUTER INSTRUCTIONS

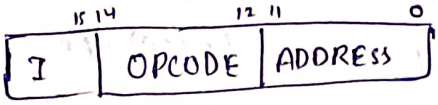
Computer instructions are a set of machine ~~langug~~ language instruction that a particular processor understands and executes. A computer performs tasks on the basis of the instruction provided.

An instruction comprises of groups called fields. These field include:

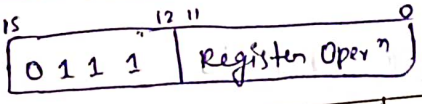
- The operation code (OPCODE) Field which specifies the operation to be performed
- The address file which contains the location of the operand, that is register or memory location.
- The mode field which specifies how the operand will be located.

→ A basic computer has three instruction code formats which are:-

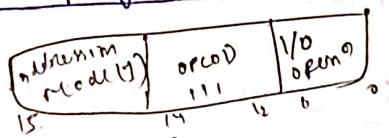
(i) Memory instruction →



(ii) Register instruction →



(iii) Input - Output instruction



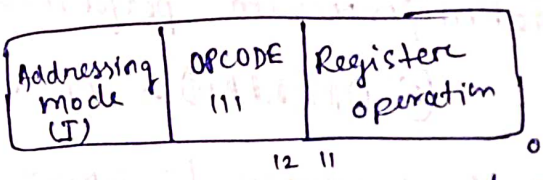
→ Memory - reference instruction :- In this instructions ~~are represented by the oper~~ 12 bits of memory is used to specify an address and one bit to specify the addressing mode 'I'.

→ Register - reference instruction

2 byte = 16 bit

3 bit - opcode = $2^3 = 8$

000 = AND
001 = ADD
010 = LOAD
011 = STA
101 = BUN (Branch)
110 = ISZ
111

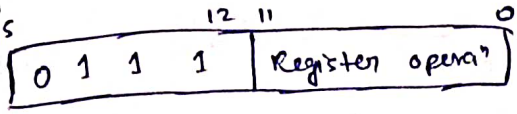


I=0 , opcode = 111 , code = 7400

CAL = Clear Accumulator (Register)
CLE = Clear E (Register)
CMA = complement Accumulator E and Accumulator is a register.

→ INPUT-OUTPUT INSTRUCTION :-

- Register - reference instruction :-

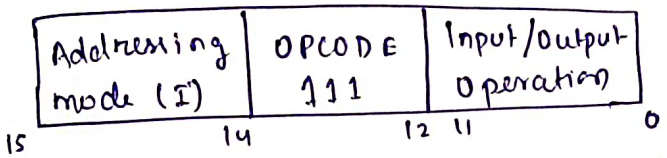


The register-reference instruction are represented by the opcode 111 with a 0 in the leftmost bit (bit 15) of the instruction.

A register-reference instruction specifies an operation on a test of the AC (Accumulator) register

- INPUT OUTPUT INSTRUCTION :-

An Input-output instruction does not need a reference to memory and is recognised by the operation code 111 with a 1 in the leftmost bit of the instruction. The remaining 12 bits are used to specify the type of input-output operation performed.



I = 1 , 111

INP = F800 - Input character to AC (Accumulator)

- Types of Instruction :-

(i) Data Transfer Instruction
(ii) Data Manipulation Instruction
(iii) Program control Instruction

(i) Data Transfer Instruction :- This instruction is used to transfer data from one location to another location without changing the binary information content

NAME	MNEMONIC
LOAD	LD
STORE	ST
MOVE	MOV
INPUT	IN
PUSH	PUSH
POP	POP

(ii) Data manipulation instruction:- These instruction performed arithmetic, logic and shift operation.

NAME	MNEMONIC
Clear	CLR
AND	AND
OR	OR - X-OR
Compliment	COM
EX-OR	X-OR

(iii) Program control instruction :- These instruction provide decision making capability and change the path taken by the program, when executed by computer

NAME	MNEMONIC
JUMP	JMP
SKIP	SKP
RETURN	RET

* TYPES OF REGISTER

Dt - 23.11.21

→ Memory Address Register →

→ Memory data register

→ Accumulator

→ Program counter

→ Memory Buffer Register (MBR)

→ INF Instruction register

Program Counter Register :- Program counter/register holds the address of next instruction to be

Computer Register : Register holds an important position in computer architecture. These are temporary storage area in the computer, where the newly fetch dat is stored.

OR

A register is sequential circuit. Registers are group of flip-flops where each flip-flop stores one bit of information

Types

Program counter register :- It holds the address of the next instruction to be fetched.

- Instruction is fetched from the address specified by the program counter
- once the instruction is fetched then the value with each operation program counter increments its value

Memory ~~Addr~~ Address Register :-

It is a combinational circuit that holds the ~~as~~ memory address
It tells the computer which byte of information to find in storage.
It is also stone the address where the data will be stoned in memory

Instruction Register :- It is a part of a CPU . that holds an instruction before executed.

Once the instn is fetch from the memory the fetched instn is store in instn register.

Decoder is connected with this instn register & decode the fetched instn.

Memory Buffer Register = This register is used for buffering of the memory so that instruction is not halted abruptly to the processor.

Memory Data Register = When an instruction is decoded, then we get the information about opcode, mode and address field → This address field provides the informat- either directly or indirectly about the address of the operand.

→ When the operand address is found out the operand is fetched from that address and this operands is stored in a register known as memory data register

Accumulator :- It is a type of register that are stored by CPU.
Accumulator acts as a temporary storage location which holds an intermediate value in mathematical & logical calculatn

Keywords :-

Fetch - An instruction is fetched from the memory or picking an instrc^n ~~to the~~ from the memory

Decode - The instrd is fetched from the memory after that instruction is decoded so that instruction can be interpreted. In this phase, CPU finds out the operation to be performed and this operation is denoted by an opcode.

Execute :- Once the operation is identified during the decoding phase then the arithmetic & logical unit perform that operatn & the result is stored in memory from where the result is displayed to user.

Dt - 25.11.21

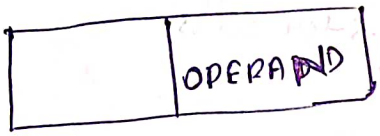
Addressing Mode :- [8086]

The term addressing mode refers to the way in which the operand of an instruction is specified.
The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually executed.

Types Of Addressing Mode :-

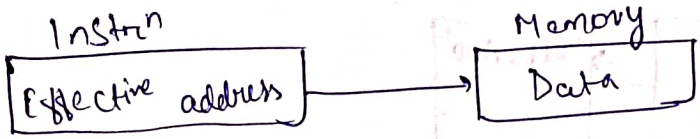
i) Immidiate addressing mode
ii) Direct addressing mode
iii) Indirect addressing mode.
iv) Register addressing mode
v) Displacement addressing mode (relative, index, based addressing)
vi) Stack addressing mode.

i) Immidiate addressing mode :- In this mode data is present in address field of instruction. Designed like one address instn format.



in this addressing mode data is directly stored here.

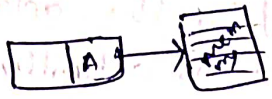
ii) Direct addressing mode / Absolute addressing Mode ( ) :- In this addressing mode the 16 bit effective address of data is the part of the instruction.
in this addressing mode only one memory reference operation is required to access the data.



* Effective address refers to the location of operand

iii) Indirect addressing mode (@,I) :- In this mode address field of instruction contains the address field of affective address. Here two references are required.

i) Register Indirect :- In this effective address is in the register, and corresponding register name will be maintained in the address field of an instruction.



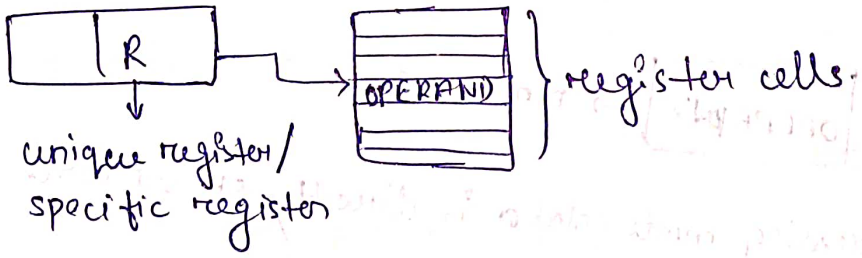
* Here only one memory reference is required to access the data

ii) Memory Indirect :- In this effective address is in the memory, and corresponding register name will be maintained in the address field of an instruction.

* Here two memory reference is required to access the data

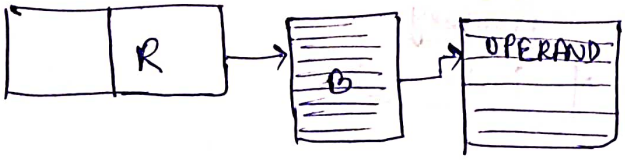
iv) Register addressing mode :- In this addressing the operand is placed in one of 8 bit or 16 bit general purpose registers. The data is in the register that is specified by the instructi

* Here one register reference is required to access the data



v) Register indirect mode: In this addressing the operand's offset is placed in any one of the registers (BX, BP, SI, DI) as specified in the instruction. The effective address of the data is in the base register or an index register that is specified by the instruction

* Here two register reference is required to access the data



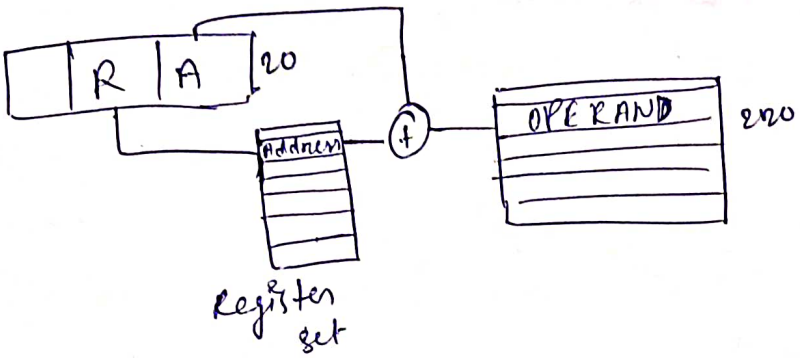
• It holds the address of another address or operand.

vi) Displacement addressing mode: It is an 8 bit or 16 bit immidiate value given in the instruction.

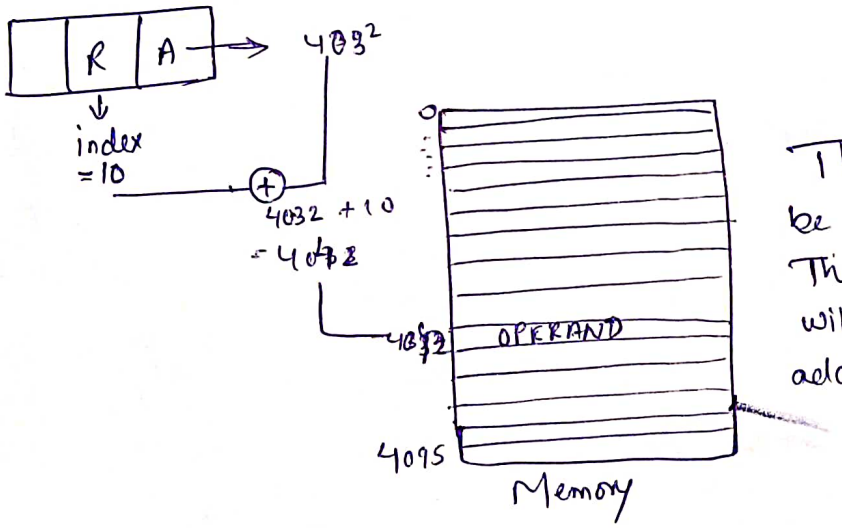
3 types :-

i) Relative
ii) Index
iii) Base

i) Relative:

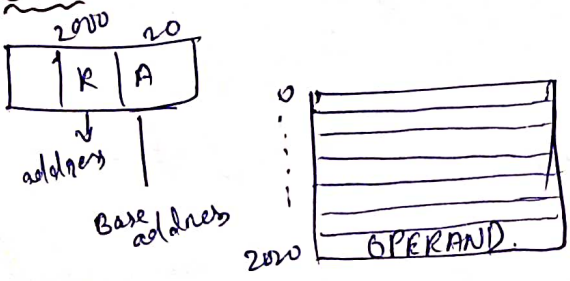


ii) Index : sum of the content register and of an register

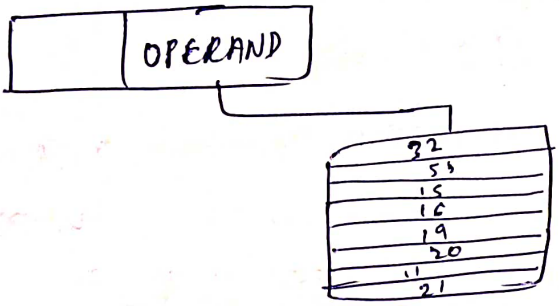


The value of index(R) will be add the adderess of A Then the resultant address will be the so known as index address

iii) Base:



vii) Stack addressing mode:

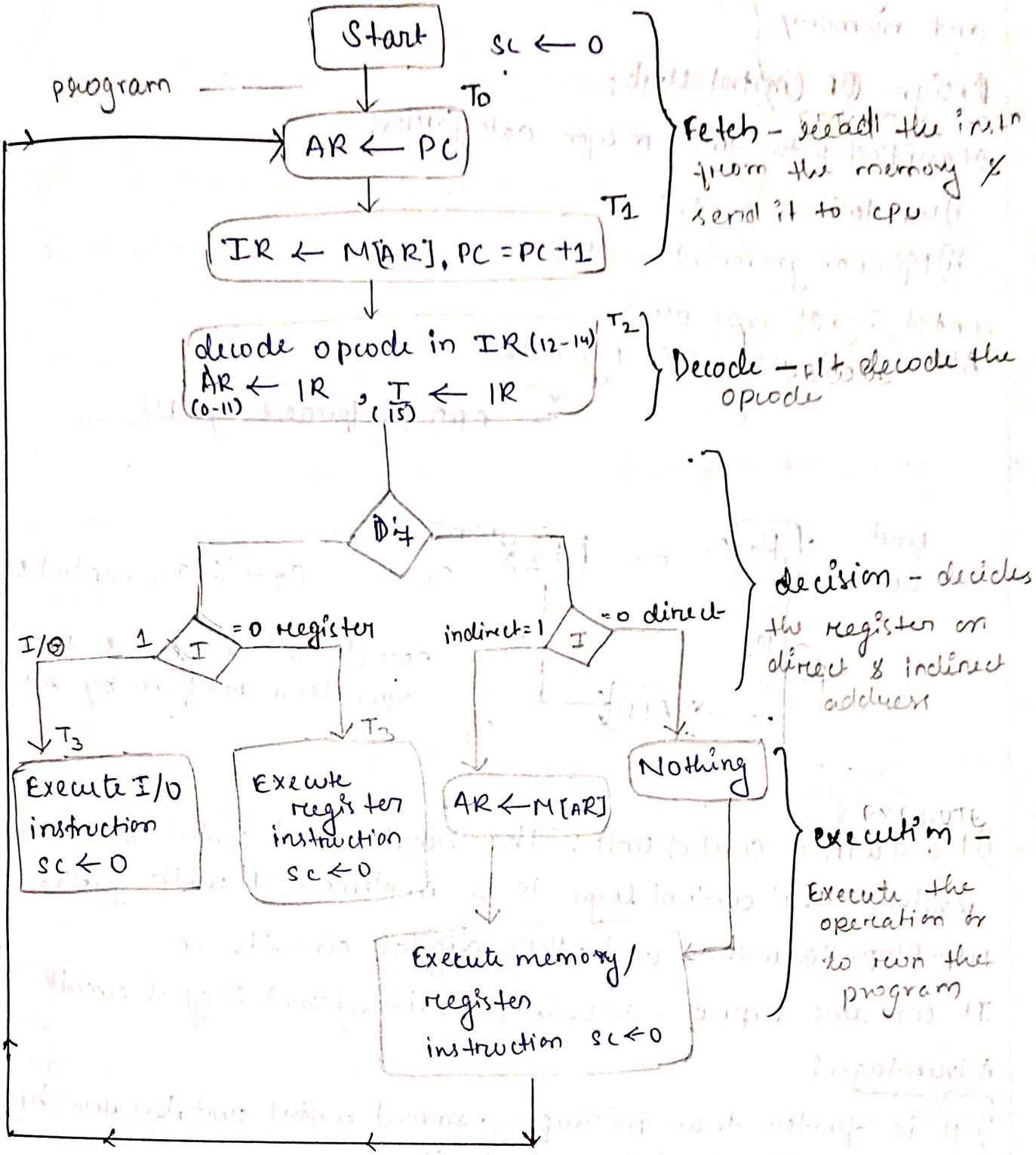


Always holds the address of first two memory.

CHAPTER-3 INSTRUCTION CYCLE

Instruction cycle:-

Fetch → decode → ~~execution decision~~ → execution



- SC - Sequence counter - It is an instruction which holds value to perform sequencly.
- AR or M[AR] - Memory address register - It stores the address where data will be stored in memory.
- PC - Program counter - It holds the address of the next instruction to be fetched.
- IR - Instruction register - holds an instruction before executed.

Control Unit: It directs the components in a computer by sending control signal & control signals flows between CPU, I/O and memory

Design Of Control Unit:

Classified into two major categories:

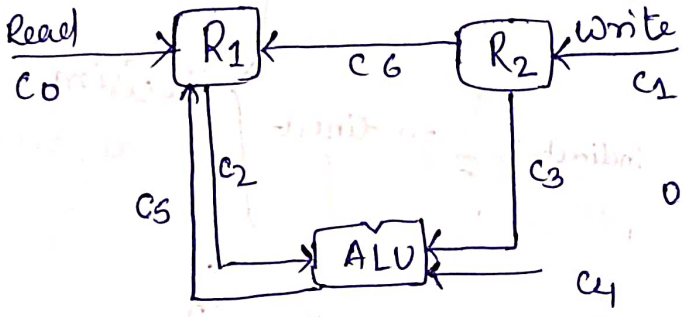
i) Hardwired control

ii) Microprogrammed control

Control Signal example:

ADD, R1, R2 is R1 ← R1 + R2

← ADD performed by ALU



$C_0 - C_6$ are control signals

opcode will decode & then operation perform by ALU

i) Hardwired control unit: The hard-wired control organization involves the control logic to be implemented with gates, flip-flops, decoders, and other digital circuits. or

It contains flipflop, decoder, combinational logical circuit

Advantages:

i) It is faster than microprogrammed control unit, because here we use combinational logical circuit.

ii) Due to combinational circuit, performance is high

Disadvantages:

i) Complex

ii) Modification in combinational circuit is very difficult.

Dt→14-12-21

ii) Microprogrammed control unit :-

It contains } micro-instructions
↓
micro-operations

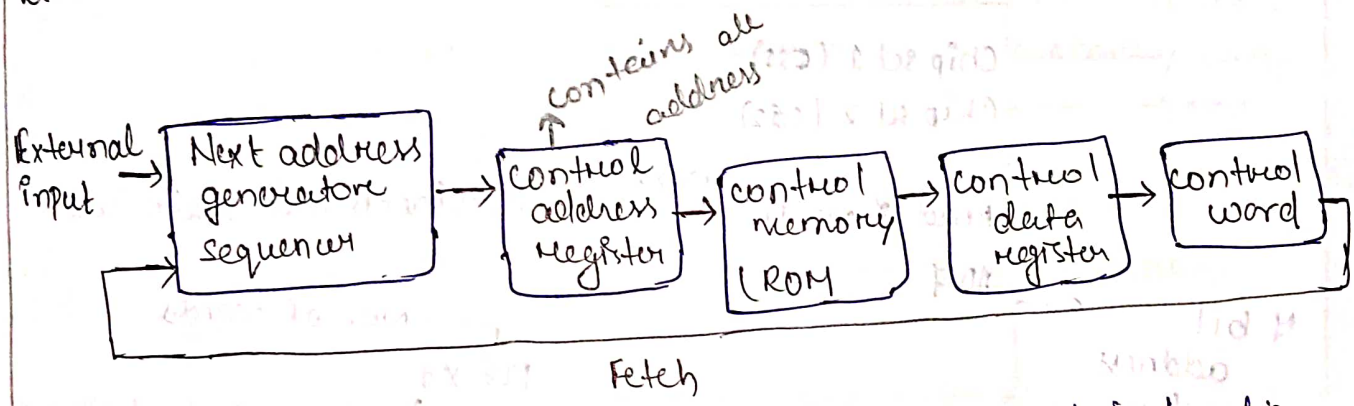
i) It is implemented using programming approach.

ii) A sequence of micro-operations are carried out by executing a program consisting of micro-instructions.

iii) Micro-programmed are stored in a memory of control unit as control memory.

iv) Execution of micro-instruction is responsible for generation of control signal.

* Architecture of microprogrammed control unit:-



* Code written in binary to perform instructions, each instructions has their own & different code, these code are not constant, it changes according to code size & memory.

* The address of micro-instruction is that is to be executed, it stored in control address register.

* This micro-instruction contains control word to executed one or more micro-operations.

* After the execution of all micro-operation of micro-instruction the address of next micro-instruction is located.

* Advanteges of microprogrammed instruction:

* It is less complex in desing because microprogram is implemented using software routine.

* It is more flexible because desing, modification, construction & enhancement is easily possible.

* less error to implemented

* Disadvanteges

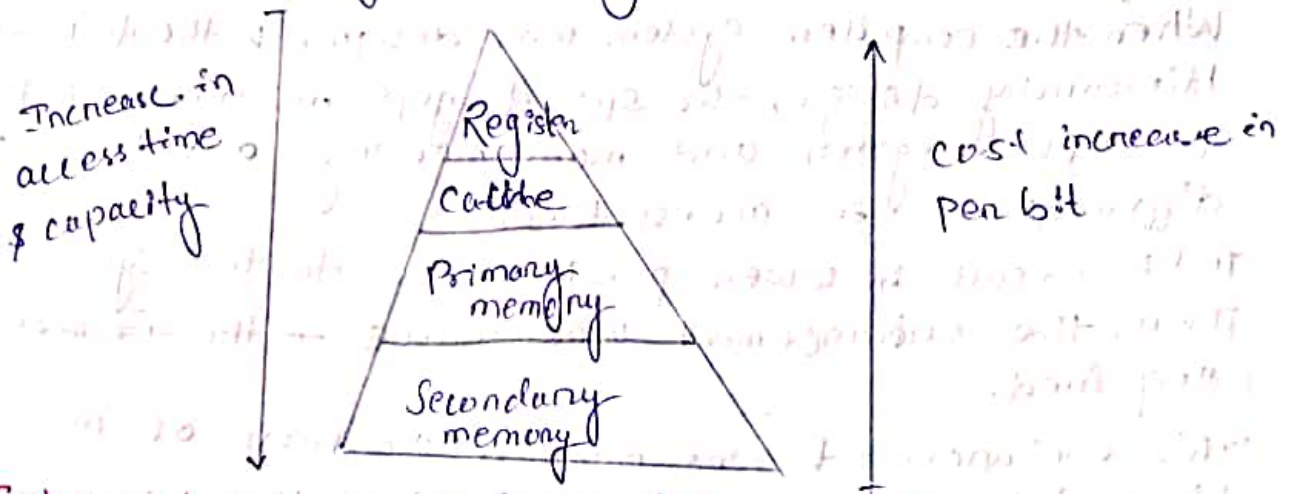
* It is slower than hardwired control unit, because it takes more time to generate control signals

Memory hierarchy and characteristics :-

In the computer system design, memory hierarchy is an enhancement to organize the memory such that it can minimize the access time.

The memory hierarchy was developed based on a Program behavior known as Locality of references.

The figure below clearly demonstrates the different level of memory hierarchy.



External memory or Secondary memory :

Comprising of magnetic ~~disk~~ Disk, magnetic ~~Tapie~~ Tape i.e peripheral storage devices which are accessible by the processor via I/O module.

Internal memory or primary memory :

Comprising of main memory, cache memory and CPU register. This is directly accessible by the processor.

We can infer the following characteristics of memory hierarchy design from above figure.

Cache memory

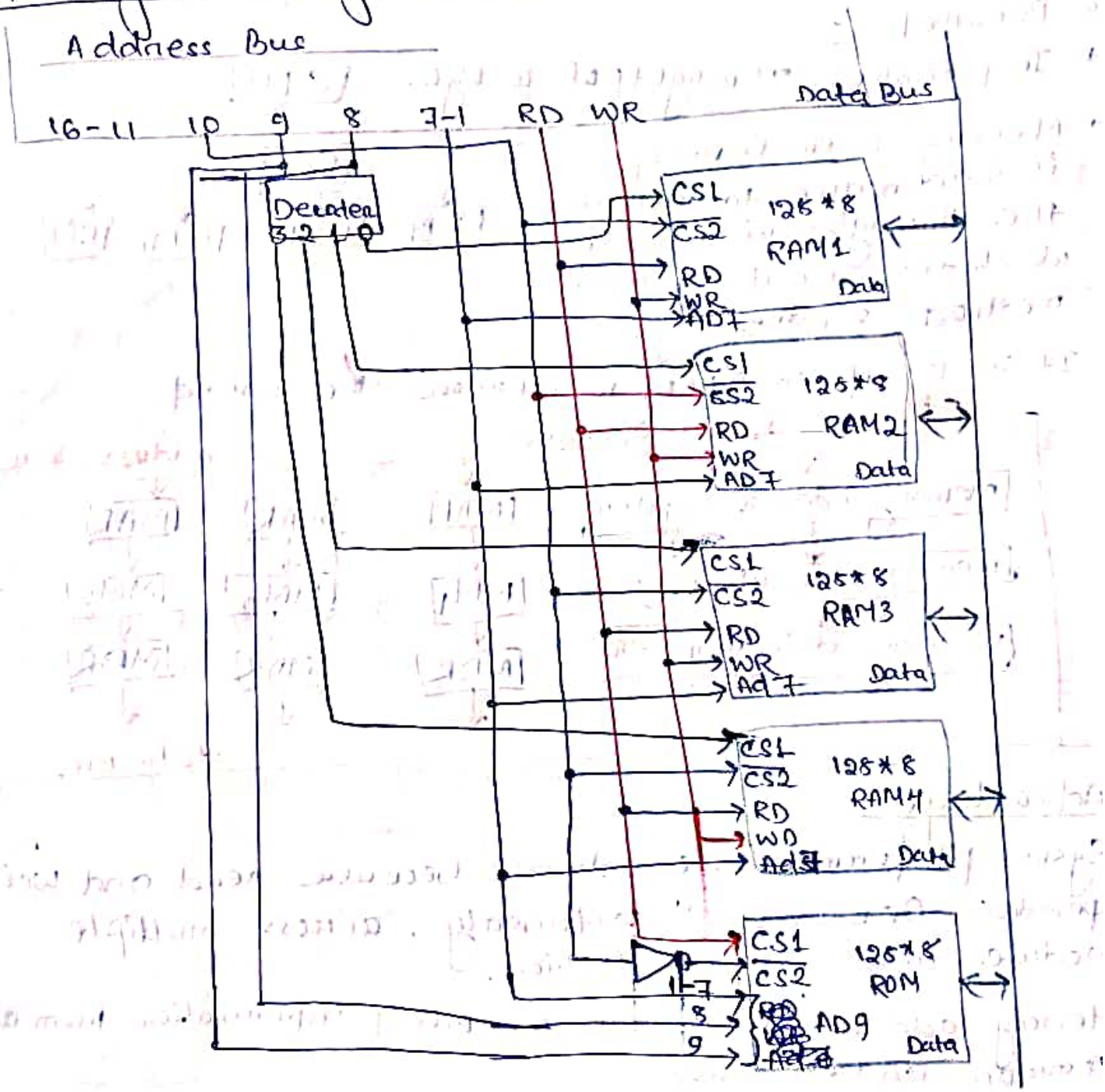
It is the memory which is very nearest to CPU, all the recent instruction are stored into cache memory.

Processor $\rightleftarrows$ Cache $\rightleftarrows$ main memory.

Register :

Component	Address Bus									
	10	9	8	7	6	5	4	3	2	1
RAM 1	0	0	0	x	x	x	x	x	x	x
RAM 2	0	0	1	x	x	x	x	x	x	x
RAM 3	0	1	0	x	x	x	x	x	x	x
RAM 4	0	1	1	x	x	x	x	x	x	x
ROM	1	x	x	x	x	x	x	x	x	x

Memory connecting to CPU :-



CACHES MEMORY

1. Caches memory is placed between main memory and CPU.
2. Caches memory is small in size and faster memory.
3. It is contain most frequently access instruction and data
4. It is located inside the CPU chips or mather board.

CPU chip	motherboard
L1 cache	L3
L2 cache	(External cache)
(Internal cache)	

CPU ↔ L1 ↔ L2 ↔ L3 ↔ RAM

CPU ↔ Cache ↔ RAM

Working of Cache

The CPU inisially lookes for the cache for other da if the data is there, it will retrive and proce it.

The data is not there then CPU access the system main memory and puts a copy of a new data in the ~~cache~~ cache memory put a copy of a new data in the cache before processing it.

Next time if the CPU needs to access some data agai it will just retrive it from the cache in speed of going through the hold loading process again

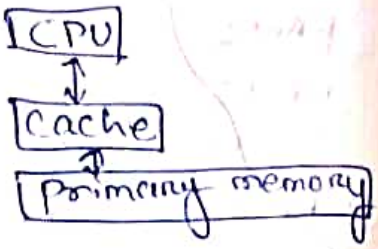
CPU ↔ Cache ↔ RAM

RAM to cache → data → block from
cache to RAM → data → word from.

Inter Leaved Memory.

(i) Wide memory organization with singal memory module :

A singal memory module cause sequencial access to instruction that is only one memory access can be perform at a time.



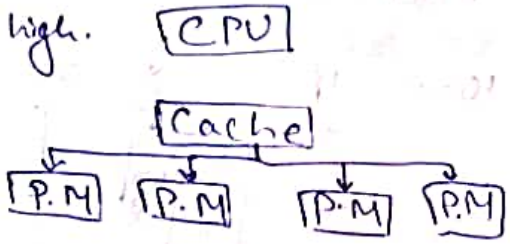
hance throughput is may drop.

* Throughtput is low.

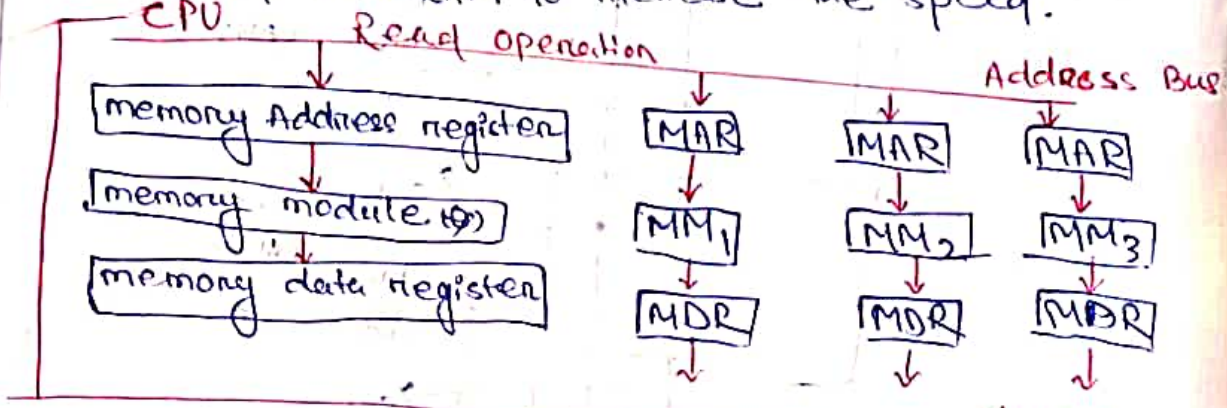
Total amount of work done in a given time that is called throughput.

(ii) Parallel :

* In parallel, Throughtput is high.
* Memory inter leaved is technique to increase the throughtput modules which one read or write mathod is parallaly.



It is used in DRAM to increase the speed.



Advantage :

i) System performance is enhance because read and write operation occures simantaneously, across multiple module in a similar fashion.

Memory address register received information from a common address Bus.

memory data register communicate with a bia final data Bus and Address Bus (unidirectional)

Functional Table

CS1	$\overline{CS2}$	RD	WR	memory function	State of Data Bus
0	0	x	x	Inhibit	High implement
0	1	x	x	"	"
1	0	0	0	"	"
1	0	0	1	Read	input data to RAM
1	0	1	x	Write	Output data to RAM
1	1	x	x	Inhibit	High implement.

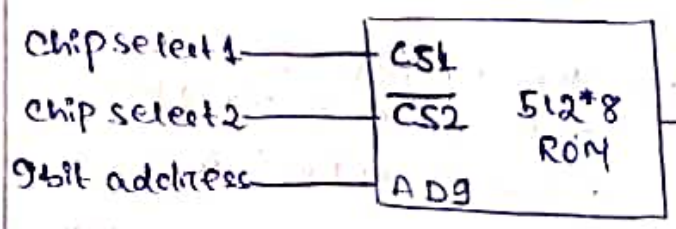
We can conclude that the unit is in operation only when CS1 = 1 and $\overline{CS2} = 0$

Data Bus State → 0 , 1 , High implement.

ROM CHIPS

ROM chips are also available in a variety of size and are also used as per the system requirement.

The chip inter connection in 512 * 8 ROM.



A ROM chip has a similar organization as a RAM chip. A ROM can only perform read operation ; The data Bus can only operate in an output mode.

The 9 bit address lines in the ROM chip specify any one of the 512 byte stored in it.

The value for chip select 1 and chip select 2 must be 1 and 0.

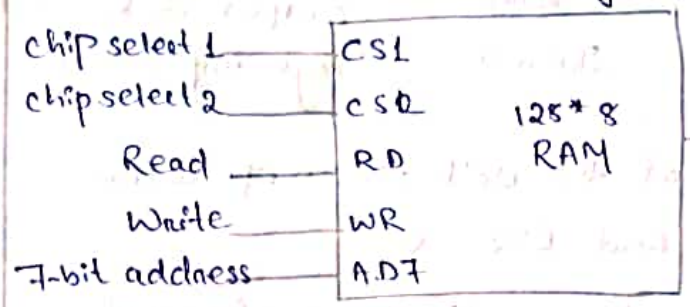
The data bus is said to be in a high ~~impli~~ impedance state.

1. RAM (Random access memory) integrated Circuit
2. ROM (Read only memory) integrated circuit chips.

RAM CHIPS ORGN

RAM chips are available in a variety of size and are used as per the system requirement.

The chip interconnecting in 128 * 8 RAM chips.



(i) 128 * 8 RAM chip has a memory capacity of 128 words. of 8 bit (one byte) per word.
Combination of several bit is known as words.
This required a 7 bit of address ∴ (Address flow → AB) and 8 bit of Data Bus (bidirectional) (Data flow → DB)

(ii) The 8 bit of Data Bus allow the transfer of data. either from memory to CPU during read operation or CPU to memory during a write operation.

(iii) The read and write inputs specify the memory operation, and the two chip select (CS) control inputs are for enabling the chip only when the microprocessor select it.

(iv) The bidirectional data Bus is constructed using three-state buffers.

(v) The output generated by three-state buffer can be, placed in one of the three possible state which include a signal equivalent to logic 1, a signal equal to logic 0, or a high impendance state.

Capacity :

It is the global volume of information the memory can store. As we move from top to bottom in the hierarchy, the access ~~time~~ capacity increases.

Access Time :-

It is the time interal between the read/write request and the availability of the data.
As we have move from top to bottom, the access time increases.

Performance

When the computer system was design, without memory Hierarchy design, the speed gap increases between the CPU register and main memory due to large difference in access time.
This result in lower performance to the system ~~and~~
Thus, the enhancement was ~~made in the from of~~ required.
This enchancement was made in the From of memory hierarchy Design because of which the performance of the system increase. One of the most significant ways to increase system performance is minimizing how far down the memory hierarchy one has to go to manipulate data.

Cost per bit :-

As we move from bottom to top in the Hierarchy, the cost per bit increase ie. Interal memory is costlier than External memory.

* Main Memory :-

(i) The main memory is acts as a central storage unit in a computer system.

(ii) It is large and fast memory used to store program and data during the run time operation.

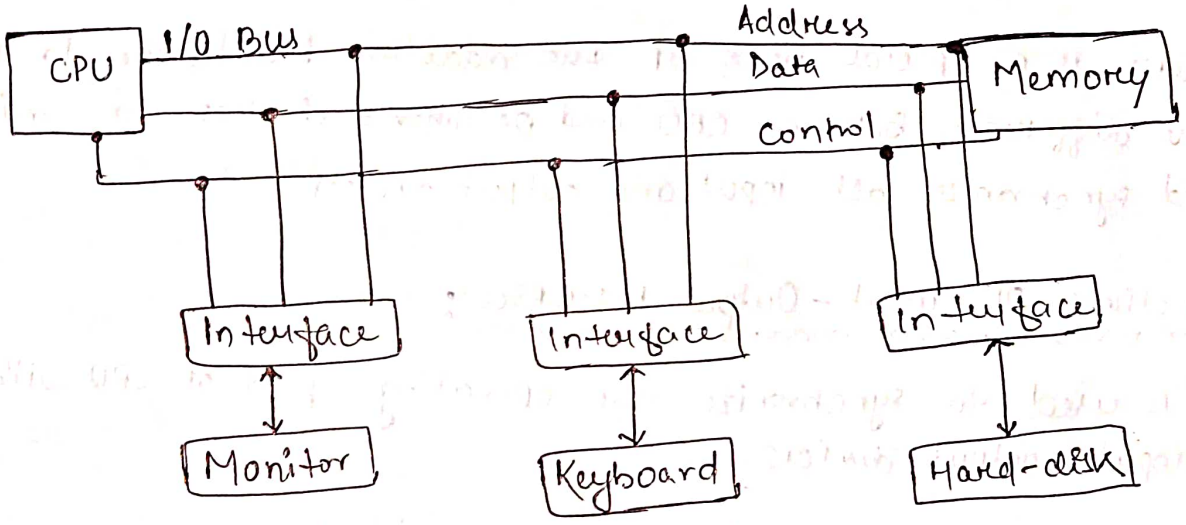
(iii) The primary ~~mor~~ technology used for the main memory is based on semiconductor integrated circuit.

(iv) The integrated circuit for the main memory are classified into two major unit.

CHAPTER-4 INPUT OUTPUT INTERFACE

Input Output Interface:

It is used as an method which helps in transferring of information between the internal storage devices that is memory and the external peripheral device. A peripheral device is that which provide input and output for the computer. It is also called input output devices. for ex: A Keyboard, Mouse provide input to the computer are called input devices, while a Monitor and printer that provide output to the computer are called output devices. Just like the external hard-drives, there is also availability of some peripheral devices which are able to provide both, input & output.



In micro-computer base_system, the only purpose of peripheral devices is just to provide special communication links for the the interfacing them with the CPU. To resolve the differences between peripheral devices and CPU, there is a special need for communication links.

The major differences are as follows.

1. The nature of peripheral devices is electromagnetic and ~~electro~~ electro-mechanical. The nature of the CPU is electronic.

There is a lot of difference in the mode of operation of both peripheral devices and CPU, there is a special need for commun' links.

The major difference are as follows:

1. The nature of peripheral devices is electromagnetic and electro-mechanical. The nature of the CPU is electronic There is a lot of difference in the mode of operation of both peripheral device & CPU.
2. There is also a synchronization mechanism because the data transfer rate of peripheral devices are slow than CPU.
3. In peripheral devices, data code and formats are differ from the format in the CPU & memory
4. The operating mode of peripheral devices are different & each may be controlled so as not disturb the operations of other peripheral devices connected to CPU.

There is a special need of the addition hardware to resolve the differences between CPU and peripheral devices to survives and synchronize all input and output devices.

Functions of Input - Output Interface:

1) It is used to synchronize the operating speed of CPU with respect to input - output devices.
2) It selects the inputs-output devices which is appropriate for the interpretation of the input - output devices
3) It is capable of providing signals like control & timing signals.
4) In this data buffering can be possible through data bus.
5) There are various error detectors.
6) It converts digital data into analog signal & vice-versa

Control command :-

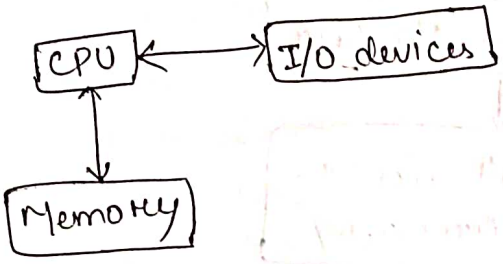
It shows activate the peripherals & to inform what to do.

Status command :-

It gives the status of peripherals devices

Keyboards, mouse, printer, external memory

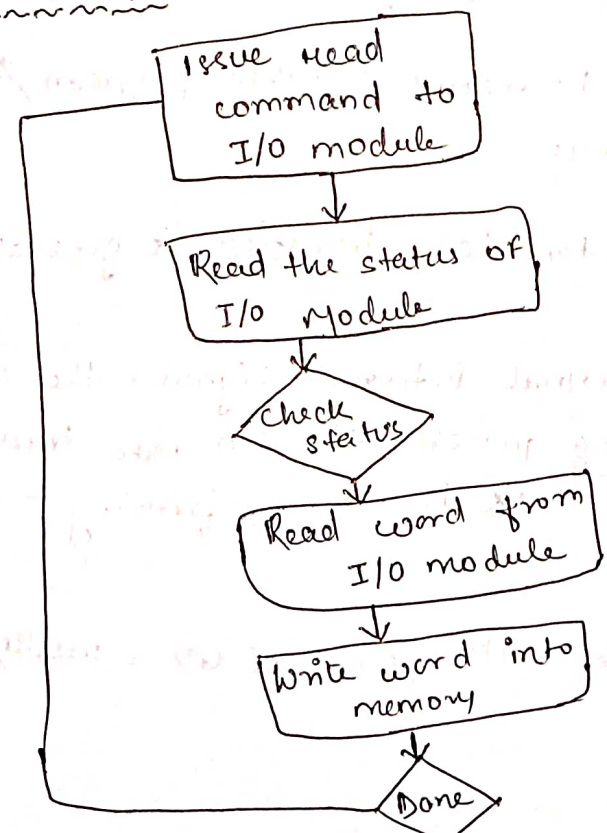
Modes of data transfer :-



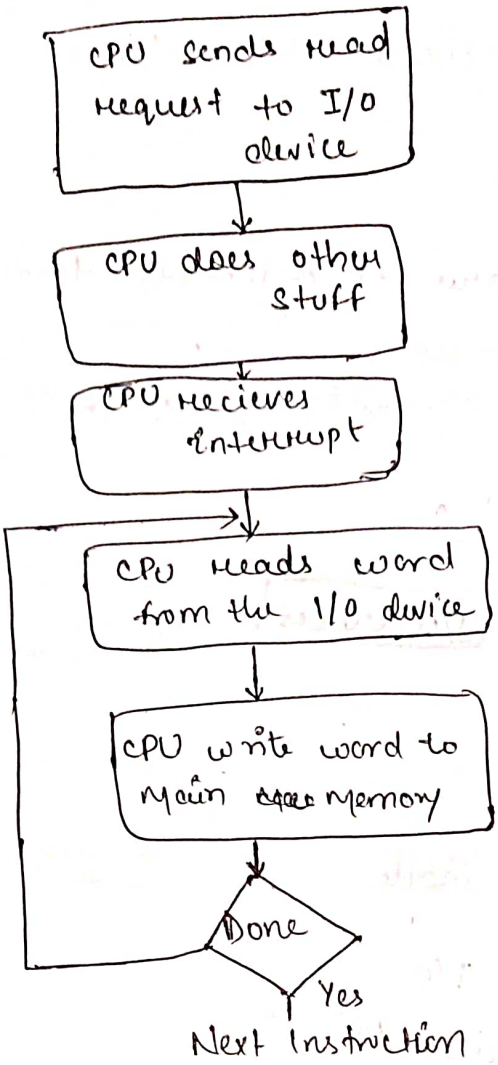
There are 3 types of mode

i) Programmed I/O
ii) Interrupt driver I/O
iii) Direct memory access

i) Programmed I/O :-



2. Interrupt driven I/O :



i) In Interrupt initiated I/O, instead of continuous monitoring of CPU, Interface will be informed to issue an interrupt request signal. When data are available from the device.

ii) Meanwhile, CPU process to execute another program & interface keeps monitoring the device.

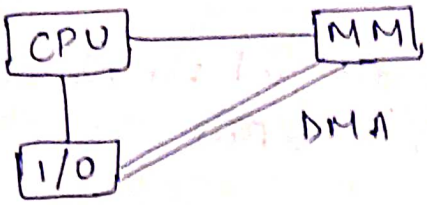
iii) When device is ready for data transfer it generate interrupt request

iv) Upon detecting the external interrupt signal, the CPU stops the task its performing process the I/O data transfers & then resume the original task it was performing.

Advantage

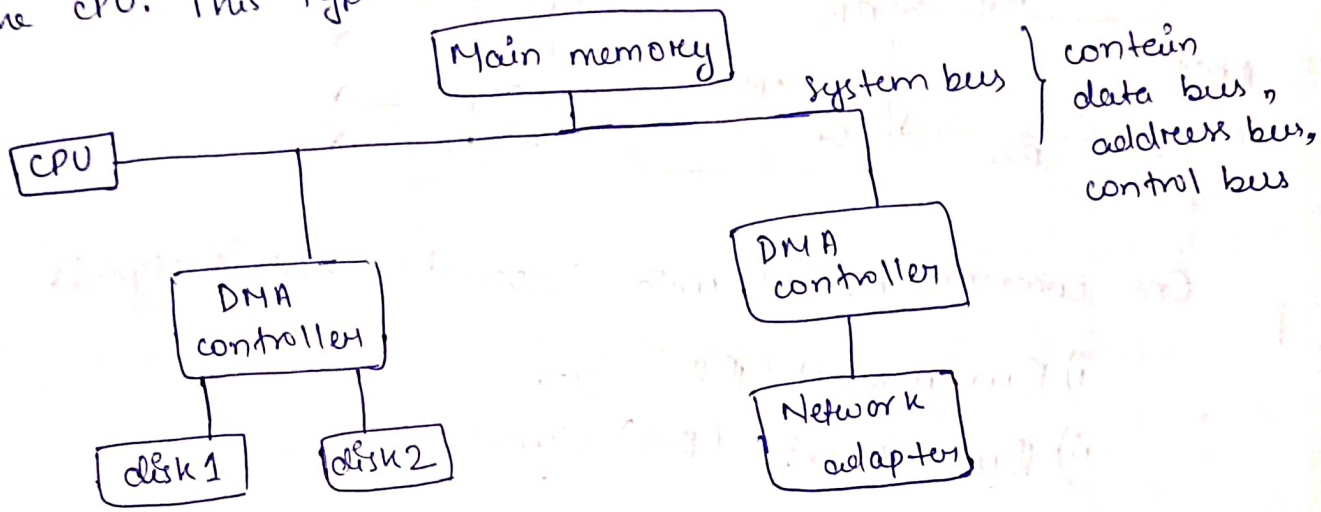
Using interrupt, the WAIT period of CPU is ideally eliminated

3. Direct Memory Access (DMA) :-

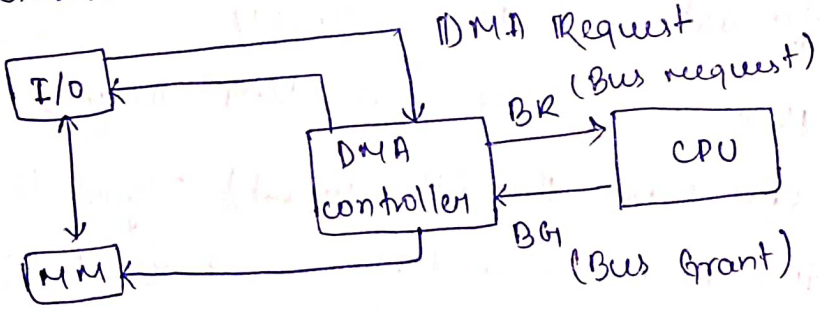


~~i) In modern computers, DMA is used to transfers large~~

i) The data transfer between a fast storage media such as magnetic disk and memory unit is limited by the speed of the CPU. Thus we can allow the peripherals directly communicate with each other using the memory buses, removing the intervention of the CPU. This type of data transfer technique is known as DMA



Actual diagram of DMA Controller



i) The direct memory access (DMA) is a I/O Technique that provide direct access to the main memory while CPU is temporarily disable to speed up the memory operation.

ii) The process is managed by chip known as DMA controller (DMAC)

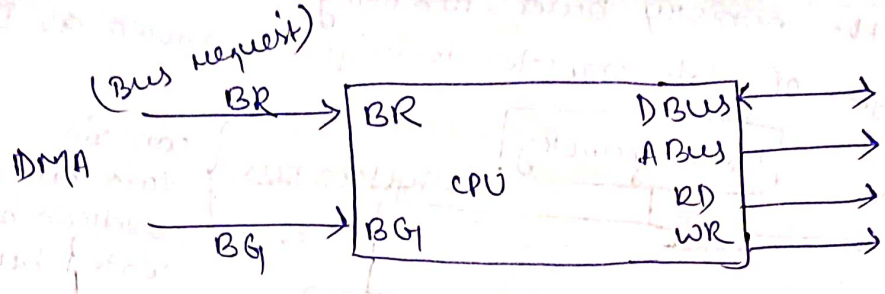
iii) I/O devices are connected to system has via a special interface circuit called "DMA controller".

iv) In DMA both CPU & DMA controller have access to main memory via a shared system bus having data, address & control lines.

v) During DMA transfer the CPU is idle and has no control of the system bus or also called memory ~~address~~ Bus

vi) DMA transfer is also used to do high speed memory-to-memory transfer

How to make CPU in ideal state?



One common method with 2 specials control signals

i) Bus request (BR) signal
ii) Bus Grant (BG) signal

DMA Working:

i) I/O wants to transfer data with main memory
ii) DMAC sends DMA request to DMA controller (DMAC)
iii) DMAC waits until CPU sends BG (Bus Grant) signal to DMA
v) CPU relinquishes control of buses and places address bus (ABus) data Bus (DBus), RD & WR lines into a high impedence state
vi) DMAC takes control of the buses to conduct direct memory transfer without CPU intervention.

BUS STRUCTURE :-

i) Bus is a group of wires or lines that connects several devices with in a computer systems.

ii) Each lines can transfer 1 unit of information

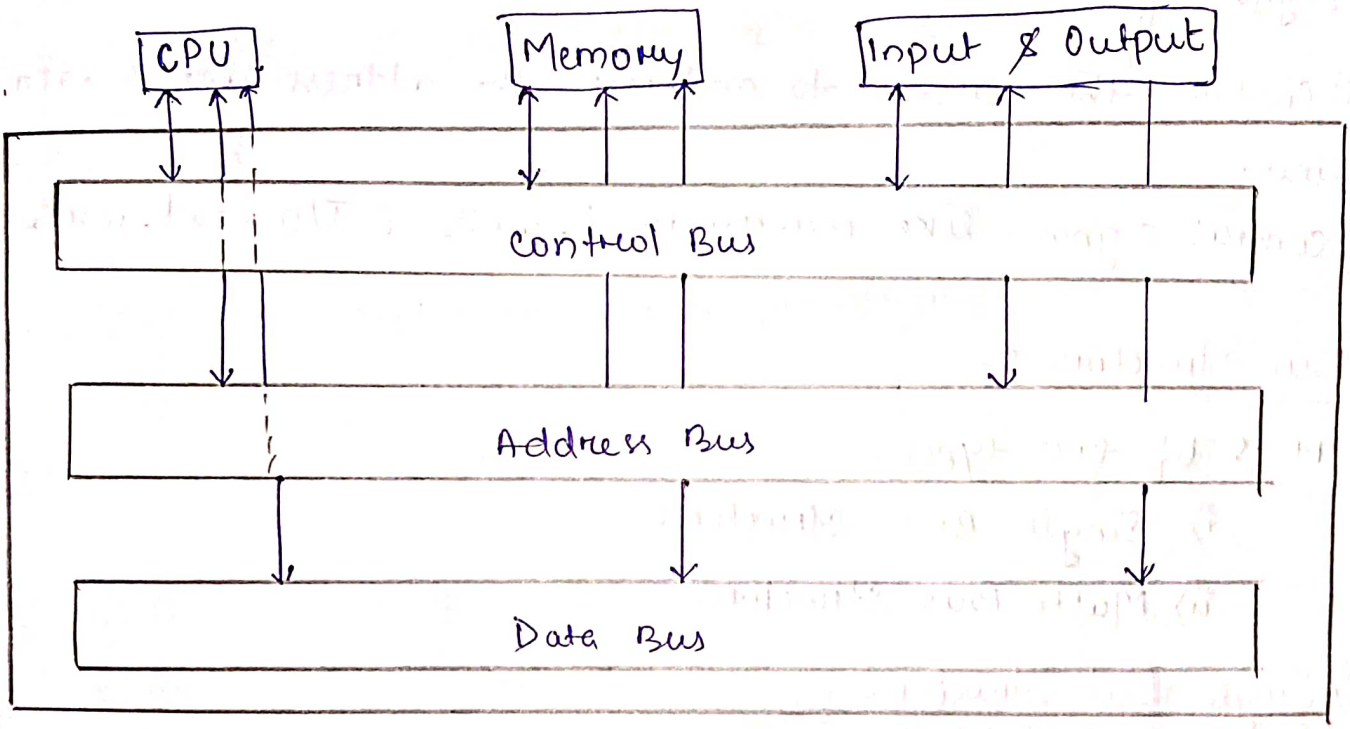
iii) Bus carries data, address, control information

iv) There are 3 types of buses

1) Address Bus

2) Data Bus

3) Control Bus.



1. Address Bus :

i) It is unidirectional and group of wires which carries address information bits from processor to peripheral

ii) The address bus with width determines the maximum memory capacity.

ex- If the address line = 3 bit the $2^3 = 8$ that is 3 address line required to select 8 location.

2. Data Bus :-

i) It is bidirectional and group of wires which contains data information bits from processor to peripherals & vice-versa

ii) In this bus data instruction move between CPU & peripherals

iii) Data Bus width determine the system performance (word length of computer)

3. Control Bus :-

i) It is bidirectional and group of wires which carries control signal from processor to peripheral & vice-versa.

ii) Control the access to and use the address line & data lines.

iii) Control signal like memory read, write & I/O read, write etc.

Bus Structure :-

It is of two types

i) Single Bus Structure

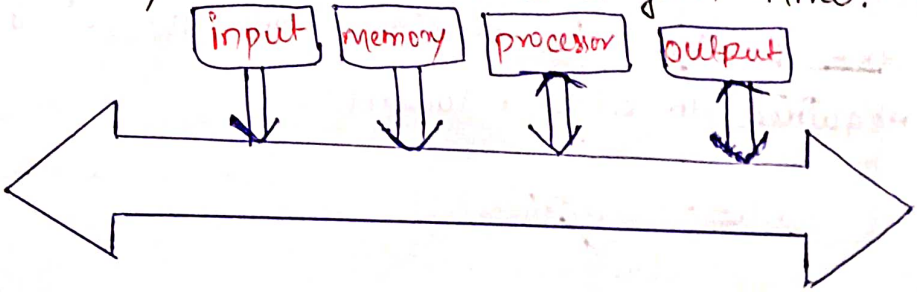
ii) Multi Bus Structure

i) Single Bus Structure :-

1) It is a simplest way to increment functional unit

2) Common bus is used to communicate between peripheral and processor

3) Single bus does ~~not~~ one transfer at a time so that only two units can actively used the bus at a given time.



Advantage :-

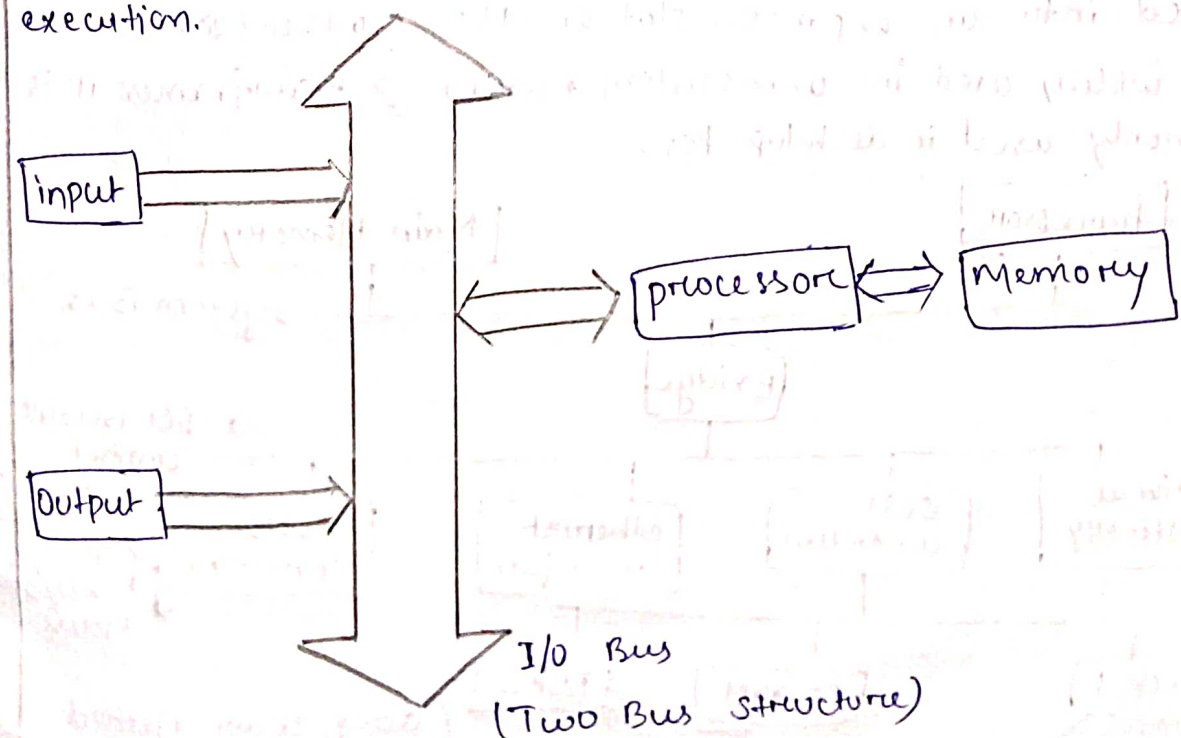
1) It is simple & low cost
2) Very flexible for attaching peripheral devices.

Drawbacks :-

i) Speed gets slow because devices connected to the Bus vary widely in their speed of operation.
ii) Efficient transfer at a time so that only two units can actively used the Bus at given time
ii) Efficient transfer mechanism is needed to solve these problem
iii) Common approach is to include buffer register with the device to hold the information during transfer.

ii) Multi Bus Structure :-

1) System that contains multiple buses to achives more parallel that leads to better performance but increase the cost
2) In multiBus structure each of which connects subset of module eg: In two Bus structure bus can be used to fetch instruction other can be used to fetch data to required for execution.



SCSI (Small Computer System interface):

i) Scsi full form is "small computer system interface"

ii) Standard parallel bus interface for connecting peripheral device to a PC.

iii) It can connect maximum 16 peripherals devices.

iv) It follows daisy chai Method

(priority wise)

v) Hand shaking of signals done by SCSI bus between devices

requesting from both side

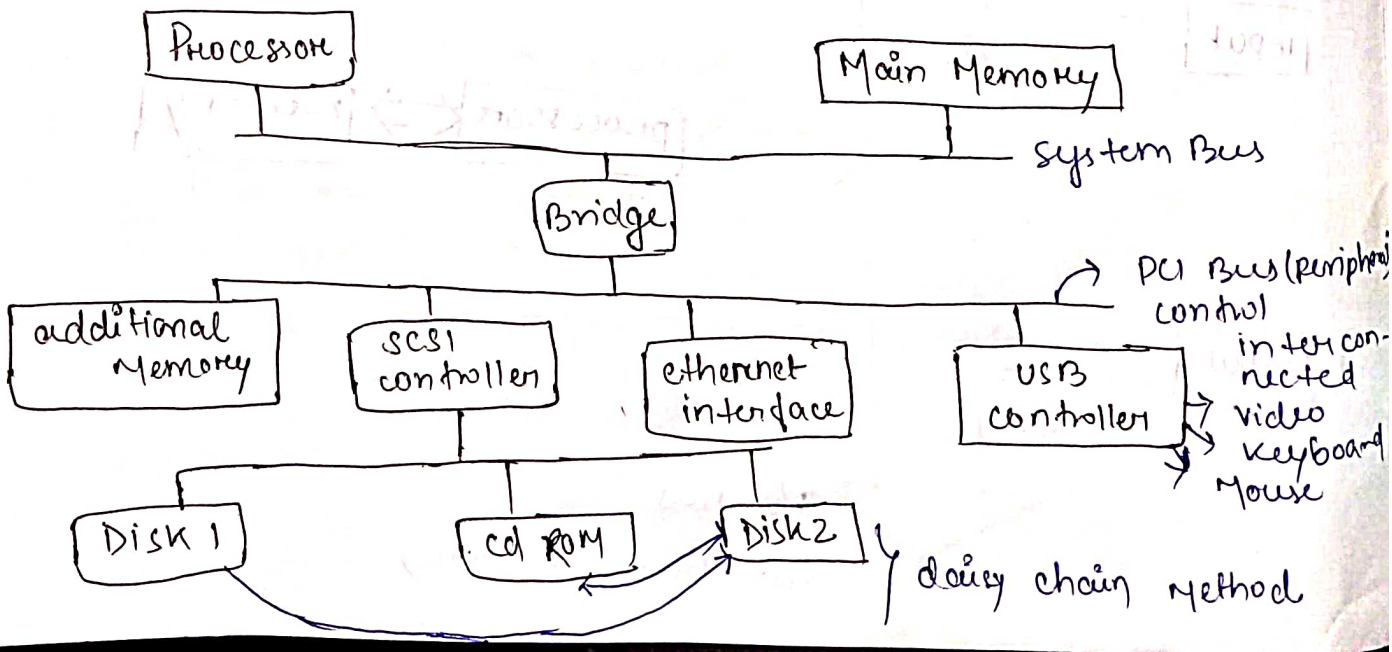
vi) SCSI is used to increase performance, deliver faster data transfer transmission.

vii) Provide expansion for device such as Hard drive, CD-ROM, drives, scanners, DVD Drivers, and CD writters, tape driver

viii) SCSI has a controller in charge of transferring data between device and SCSI bus.

ix) It is either embedded on the motherboard or a host adapter is inserted into an expansion slot on the motherboard

x) SCSI is widely used in workstation, server & Mainframes it is less commonly used in desktop Pcs.



* Parallel Bus : In this bus data transfer parallelly which increases the speed of transferring data.

USB Controller :

→ 127 peripheral devices we can add
→ Same as SCSI controller
→ Maximum speed 480mbps speed.
→ Without hub 5 meter can travel
→ app : Keyboard, printer, scanner
→ we can add mobile & tablet also.

PARALLEL PROCESSING

Serial Computing :-

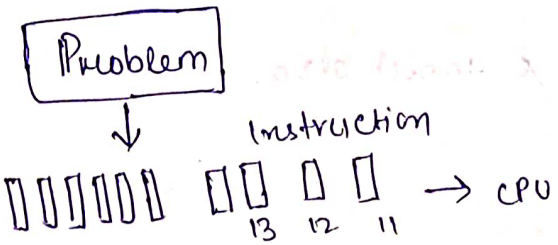
Traditionally, software has been written for serial computations

i) To be run on a single computer having a single central processing unit (CPU).

ii) A problem is broken into a discreate ~~parts that can be~~ series of instructions

iii) Instruction are executed one after another

iv) Only one instruction may execute at any moment in time



Parallel Computing :-

Parallel computing is the simultaneously use of multiple computing resources to solve a computational problem

i) To be run using multiple CPUs.

ii) A problem is broken into discreate parts that can be solved concurrently.

iii) Each parts is further broken down a series of instructions

iv) Instruction from each part execute simultaneously on different CPUs

* Why parallel computing :

The real world is parallel :

i) Complex, inter related events happen simultaneously

Eg - galaxies, planetary movements, functioning of the brain
weather, traffic

ii) Parallel computing is better suited for modeling, simultaneously these process.

Need of parallel computing:-

i) Save time & money: Money resources working together. will reduce the time and cut potential costs.

Loop Level :-

i) At this level consecutive loop ~~interaction~~ iteration are the candidates for parallel executions

ii) However, data dependencies between subsquent iterations may rest parallel execution of instruction at loop level. There is a lot of scope for parallel execution at loop level

iii) Ex: In the following loop in C language

```
for (i=0; i<=n; i++)
    A(i) = B(i) + c(i)
```

iv) Each of the instruction A(i) = B(i) + C(i) can be executed by different processing elements there are at least n processing elements.

v) However, the instruction in the loop

```
for (J=0 , J<=n , J++)
    A(J) = A(J-1) + B(J)
```

vi) Cannot be executed in parallel as a(J) is data dependent on A(J-1). This means that before exploiting the loop level parallelism the data dependencies must be checked.

Von Neumann Architecture:

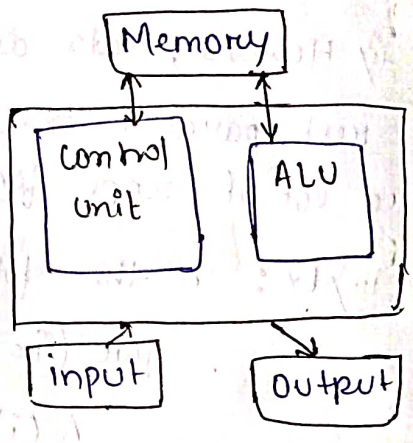
i) Von-Neumann proposed his computer architecture design in 1945 which was later known as Von-Neumann architecture.

ii) It consisted of a control unit, Arithmetic, and logical unit (ALU), resisters, Memory units & input/output units. The combination of ALU & control unit is called central processing unit or processing element (PE)

iii) Von-Neumann architecture is based on stored-program computer concept, where instruction data & program data are stored in the same memory. This desing is still used in most computer produced today.

iv) A Von-Neumann beased computer:

1) Uses a single processor
2) Uses one memory for both instruction ~~and~~ & data
3) Executed programms following the fetch-decode executes cycle



v) compiled of four main components.

1) Memory
2) Control unit
3) Arithmetic logic unit
4) Input/Output

vi) Read/Write. (RAM) is used to both program & instruction & data

1) Program instruction are coded data which tell the computer to do something.

2) Data is simply information to be used by the program

vii) Control unit fetches instruction/data from memory, decode the instruction and they sequentially co-ordinates operations to accomplish the program task.

iii) Arithmetic logic unit performs basic arithmetic & logic operations input/output unit is the interfaces to the human operator.

Linear pipeline processor :-

Linear pipeline processor is a cascade of processing stage which are linearly connected to perform a fixed function over a steam of data following from one end to another.

i) A linear pipeline processor is constructed with K processing stages

ii) External input is supplied to the pipeline from stage S_1.

iii) The processed result is raised from S_i to stage S_{i+1}, from all $i = 1, 2, \ldots, K-1$

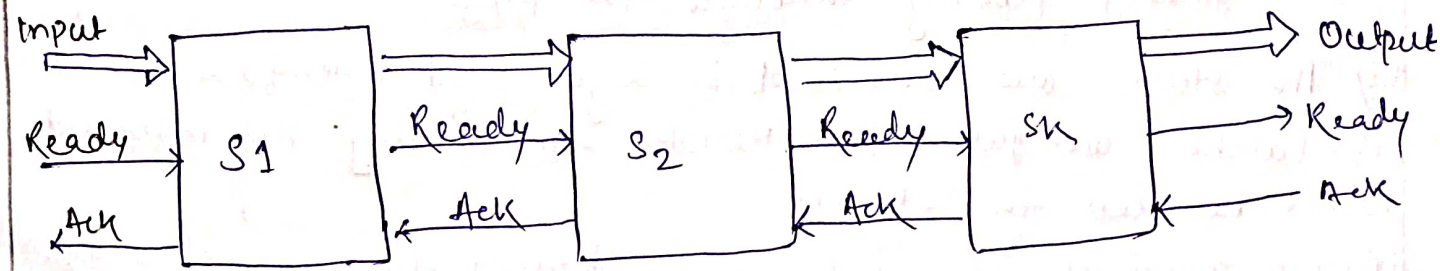
iv) The final result is produced from the last pipeline stage S_K

v) Depending on the control of data flow linear pipeline ~~stage~~ are divided into two model:

1) Asynchronous pipeline model
2) Synchronous pipeline model

Asynchronous pipeline model :-

i) In Asynchronous pipeline the data flow between adjecent stages are cancelled by handshaking protocol.



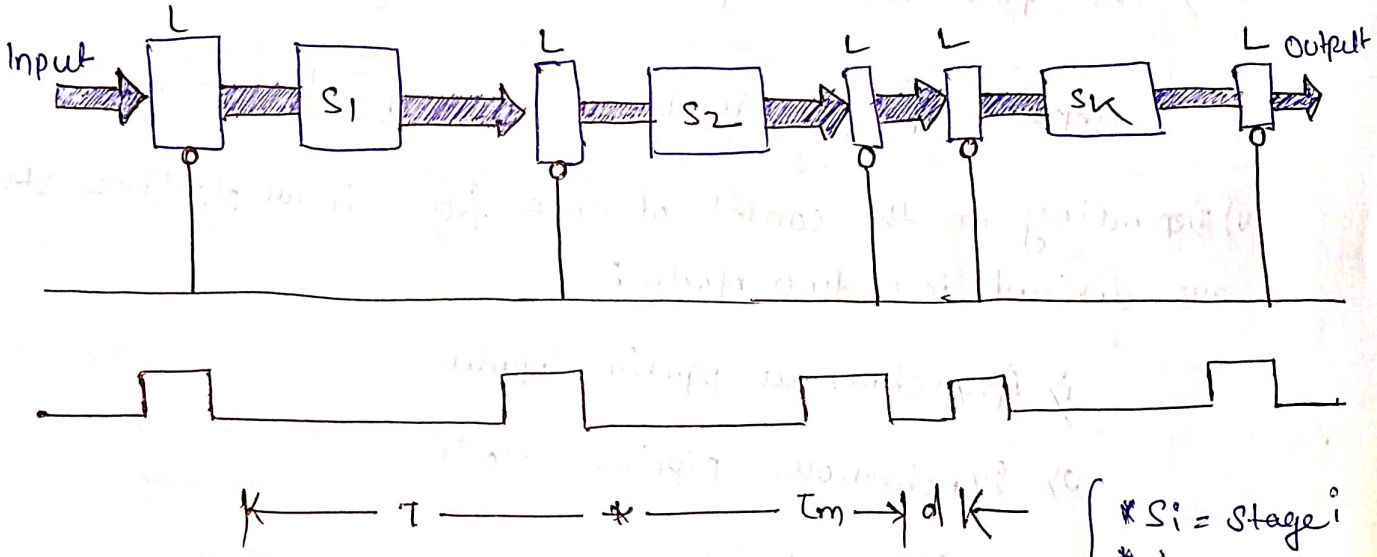
ii) When the stage S_i is ready to transmit it sends a ready signal to stage S_{i+1}. After S_{i+1} receives the incoming data, it returns acknowledgement signal (Ack) to S_i.

iii) This pipeline are useful in designing communication channel in message passing multicomputer

iv) This pipeline may have a variable throughput rate because different amount of delay may be experienced in different steges different ~~amount of~~ stages.

Synchronous pipeline model:

i) In synchronous pipeline, clocked catches are used to interface between stages.



* S_i = Stage i
* L = Latch
* τ = clock period
* τ_m = Maximum stage delay
* d = Latch delay

ii) The pipeline consist of cascade of processing stage (S_i). The pipeline stages are combinational circuits performing arithmetic & logic operation over the data streamy flowing through the pipe

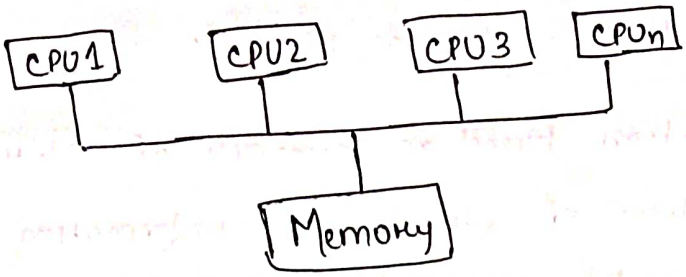
iii) The stages are separated by high speed interface latches (L) The latches are ~~just~~ fast resister for holding the intermediate result between the stages.

iv) Upon arival of clock pulse latches transfer data to next stage

v) The utilization pattern successive stages in this pipeline is specified by reservan table.

Multiprocessor :

There are more than one processor present in the system which can execute more than one processor at the same time



There are 2 types of multiprocessor :-

i) Symmetric multiprocessor
ii) Asymmetric multiprocessor.

Symmetric Multiprocessor :

One operating system control all CPU. Each CPU has equal rights. All the CPU are in peer to peer relationship.

Asymmetric Multiprocessor :

There is a master process that gives instruction to all the other processor. It contains master-solve relationship.

Advantages :

i) Maximum throughput
ii) More reliable system.
iii) fast processing
iv) Efficiency improved
v) More economic system.

flynn's classification :

It is based on the multiplicity of instruction stream & data stream in a computer system.

There are four categories that is:

i) SISD (Single instruction stream single data stream)
ii) SIMD (Single instruction stream multiple data stream)
iii) MISD (Multiple Instruction stream single data stream)
iv) MIMD (Multiple Instruction stream multiple data stream)

1) Flynn's classification based on number of instruction & data
2) Based on the notion of stream of information
two types of information flow into a processor:
instruction & data.
3) Instruction stream is defined as the sequence of instruction executed by processing unit.
4) Data stream, is defined as the sequence of data including inputs, partial or temporary results, called by the instruction stream.

i) Single Instruction Single Data Stream (SISD).

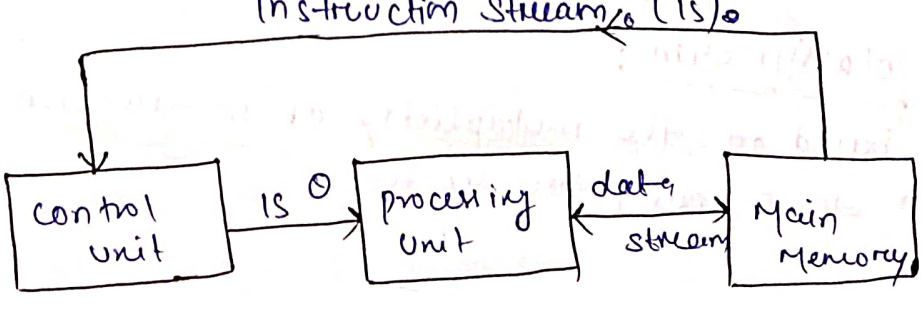
i) SISD computing system is a uni-processor machine which is capable of executing a single instruction, operating on a single data stream.

ii) Conventional signal processor Von-Neumann computer are classified as SISD system.

iii) It is serial (non-parallel) computer.

iv) Instruction are executed sequentially but may be overlapped in their execution stages (pipelining). Most SISD uni-processor system are pipelined.

v) Ex - most pc's, single CPU workstation, minicomputer etc.



ii) Single Instruction Stream Multiple Data Stream (SIMD):

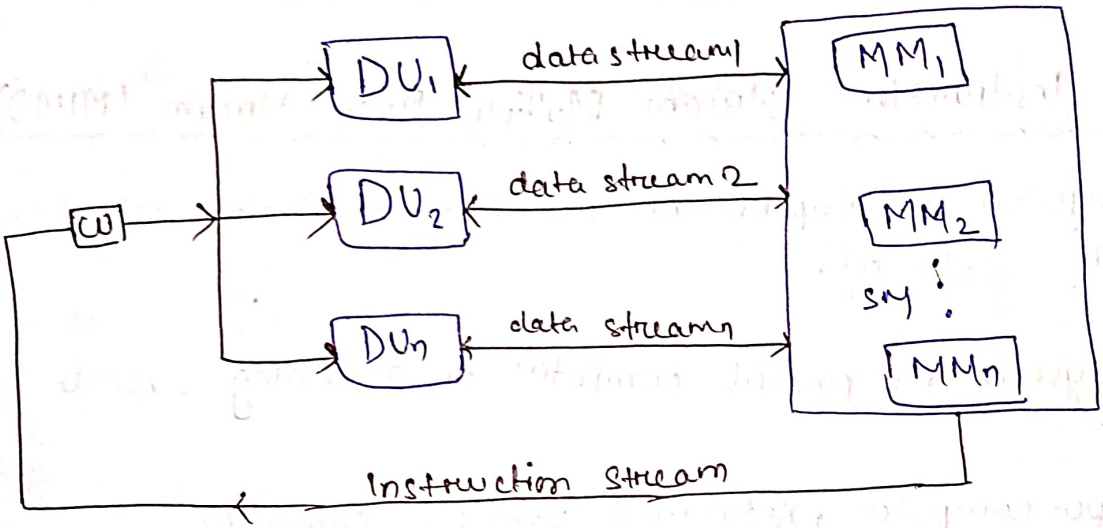
1) SIMD system is capable of executing the same instruction on all the CPUs but operating on different data Stream.

2) SIMD computer has single control unit which issue one instruction at a time but it has multiple ALU's or processing units to carry out on multiple data sets.

3) Ex - Array processor and vector pipelines

Array processor = MPP, ILLIAC - IV

Vector pipelines - IBM 9000, cray x-MAP, YMP & C90.



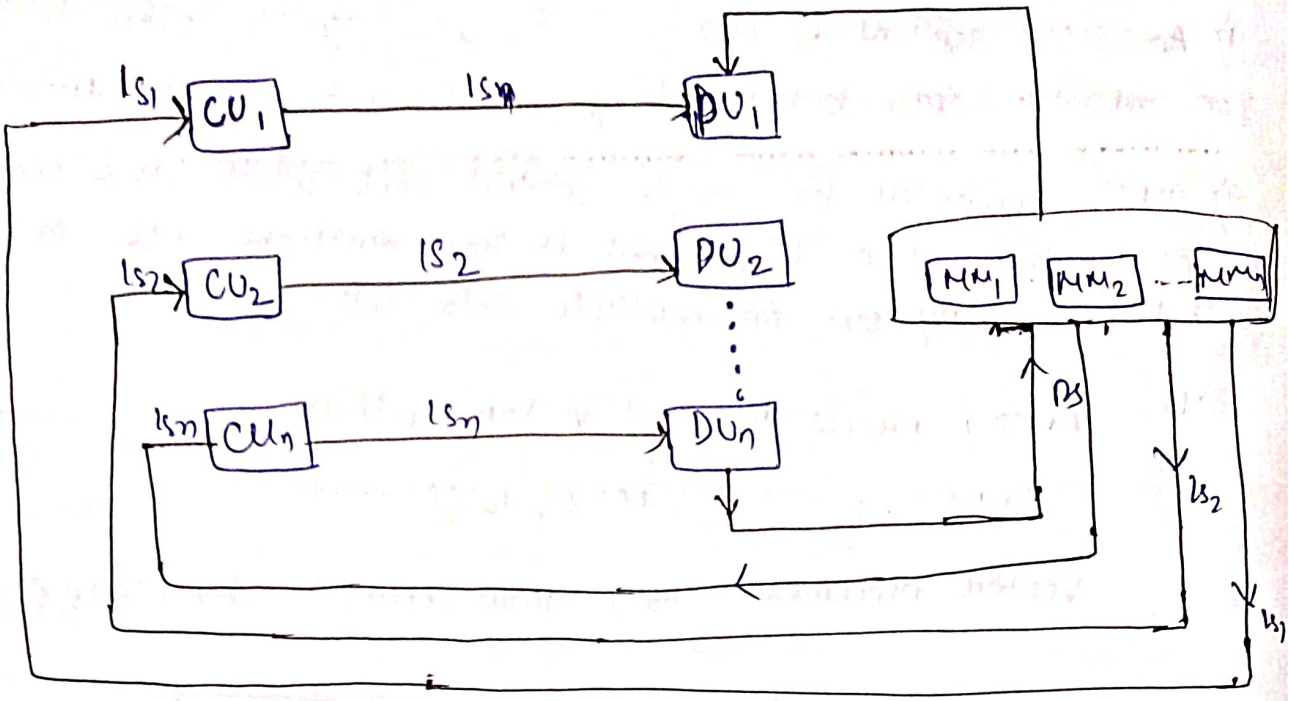
iii) Multiple Instruction Stream Single Data Stream (MISD):

1) In MISD multiple instruction operate on single data stream.

2) MISD computing system is capable of executing different instruction on different processing unit (PU) but all of them operating on the same data set.

3) MISD model are practically not useful in most of the applica"

4) Ex - Systolic arrays.

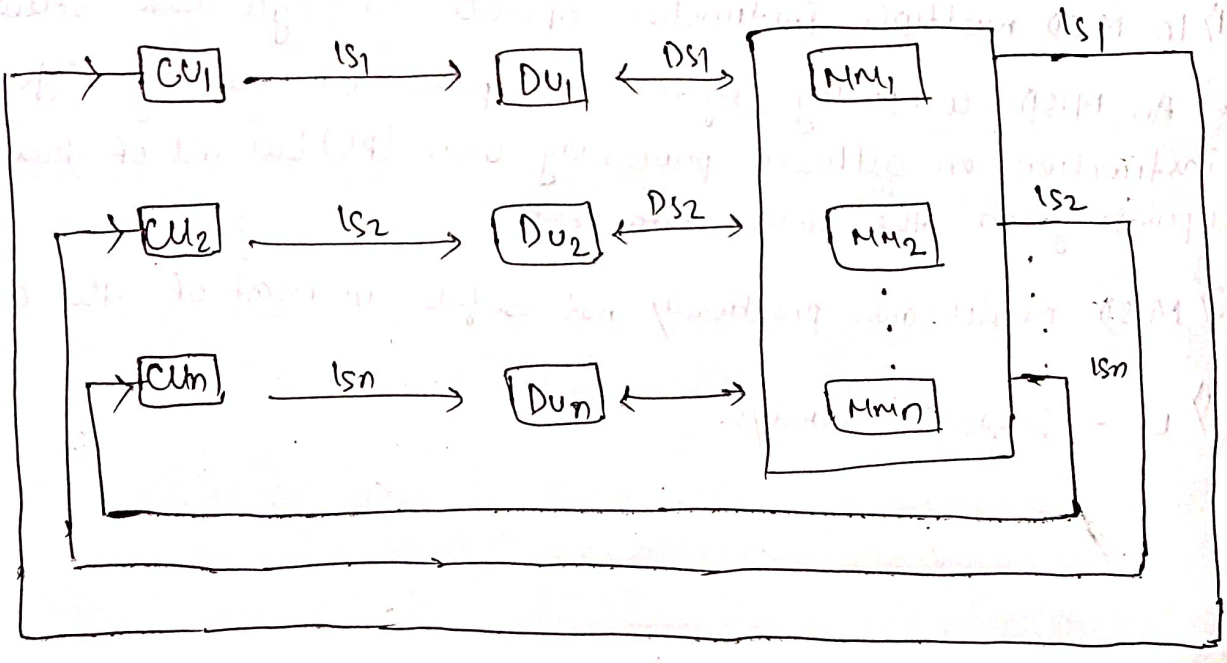


iv) Multiple Instruction Stream Multiple Data Stream (MIMD):

1) MIMD system is capable of executing multiple instruction on multiple data set.

2) MIMD system are parallel computer or processing several program.

3) Ex - Super computer, networked parallel computer, IBM 370, Cray-2



Important Notes:

Difference between CISC and RISC.

CISC	RISC
stands for Complex Instruction set computer.	stand for Reduced Instruction set Computer.
A large number of instructions are present in the architecture	Very few instructions are present.
Variable-length encoding of the instructions. Ex-IA32 instruction size can range from 1 to 15 bytes	Fixed-length encodings of the instructions are used Ex-IA32, generally all instruction are encoded as 4 bytes.
CISC supports array	RISC doesn't not support an array.
Arithmetic and logical operations can be applied to both memory and register operands.	Arithmetic and logical operations only use register operands.
Condition codes are used	No condition codes are used.
The stack is being used for procedure arguments and return addresses.	Registers are being used for procedure arguments & return address.

Define:

Clock Cycle: It is simply a "cycle" or it is a single electronic pulse of CPU. During each cycle, a CPU can perform basic operation such as fetching, an instruction, accessing memory, or writing data.

Hit Rate: The chief measurement of cache which is the percentage of all accesses that are satisfied by the data in cache. also known as "hit ratio".

Subroutine: A set of instructions that are used repeatedly in a program can be referred to as subroutine. Only one copy of this instruction is stored in the memory. When a subroutine is required it can be called many times during the execution of a particular program. A call subroutine instruction calls the subroutine.

Macro: It is used to make a sequence of computing instructions available to the programmer as a single program statement, making the programming task less tedious and less error-prone.