

COA

Unit - 3 Control Unit:

- Instruction types ✓
 - Formats ✓
 - Instruction Cycle and Sub Cycle ✓
 - Micro operation ✓
 - Execution of a complete instruction ✓
 - Program Control ✓
 - Reduced Instruction Set Computer ✓
 - Pipelining ✓
 - Hardware and micro programmed control
 - * Micro Program Sequencing
 - * Concept of horizontal and vertical microprogramming

Types of ~~for~~ Instructions

- a) Memory Reference Instruction | Opcode - 0 to 6, 8 to E
 b) Register | Opcode - 7
 c) I/O Instruction | Opcode - F

Memory Reference Instruction: Divided into three parts -
mode (1 bit), opcode (3 bits), address (12 bits).

If addressing mode specifies '0' then direct addressing,
if bit specifies '1' then indirect.

Instructions can be: AND, ADD, LDA, STA, etc.

Register Reference Instruction: Same as memory, but opcode is always '111' and last bit (addressing mode bit) is always 'zero (0)'.

Instructions - HLT, SZE, INC, etc.

Input-Output Instruction: Same as memory, but opcode is always '111' and last bit is always 'one (1)'.

Instructions - IOP, IOIN, OUT, etc.

Addressing modes: Indicates how the data is represented

Opcode: Indicates the operation type on the data

Operand: Indicates either the data or the address of the data

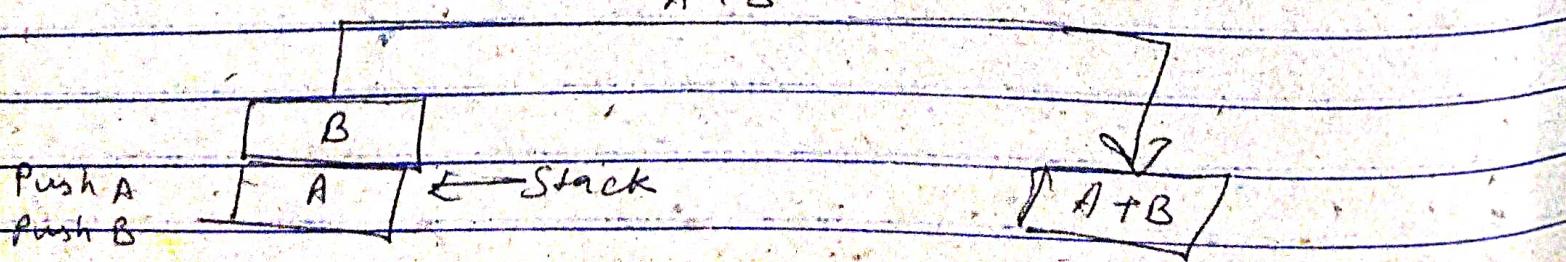
Instruction format: refer to the way instructions are encoded and represented in machine language

- Four types

- i) Zero Address Instruction
- ii) One " "
- iii) Two "
- iv) Three "

Zero Address: These instruction do not specify any operands or addresses. Instead they operate on data stored in registers or memory locations.

$$A + B$$



Ex - Expression $X = (A + B) * C + D$
TOP means top of stack

PUSH A TOP = A

PUSH B TOP = B

ADD TOP = A + B

PUSH C TOP = C

PUSH D TOP = D

ADD TOP = C + D

MUL TOP = $(A + B) * (C + D)$

POP X $M[X] = \text{TOP}$

One Address : specify one operand or address, which typically refers to a memory location or register. The instruction operates on the content of operand and the result may be stored in the same or a different location.

opcode | operand / address of | mode |
 | operand

Ex - Expression $X = (A + B) * (C + D)$
AC is Accumulator

LOAD A AC = $M[A]$

ADD B AC = $AC + M[B]$

STORE T $M[T] = AC$

LOAD C AC = $M[C]$

ADD D AC = $AC + M[D]$

MUL T AC = $AC * M[T]$

STORE X $M[X] = AC$

Two Address: specify two operands or addresses. The instruction operates on the content of both operands, and the result may be stored in the same or a diff. location.

[opcode | destination address | source address | mode]

Ex- Expression $X = (A+B) * (C+D)$

R_1, R_2 are registers

MOV R_1, A $R_1 = M[A]$

ADD R_1, B $R_1 = R_1 + M[B]$

MOV R_2, C $R_2 = M[C]$

ADD R_2, D $R_2 = R_2 + M[D]$

MUL R_1, R_2 $R_1 = R_1 * R_2$

MOV X, R_1 ~~$X = R_1$~~ $M[X] = R_1$

Three Address: same as two, but replace with three

[opcode | destination Address | source Address | source Address | mode]

Ex- Expression $X = (A+B) * (C+D)$

R_1, R_2 are registers

ADD R_1, A, B $R_1 = M[A] + M[B]$

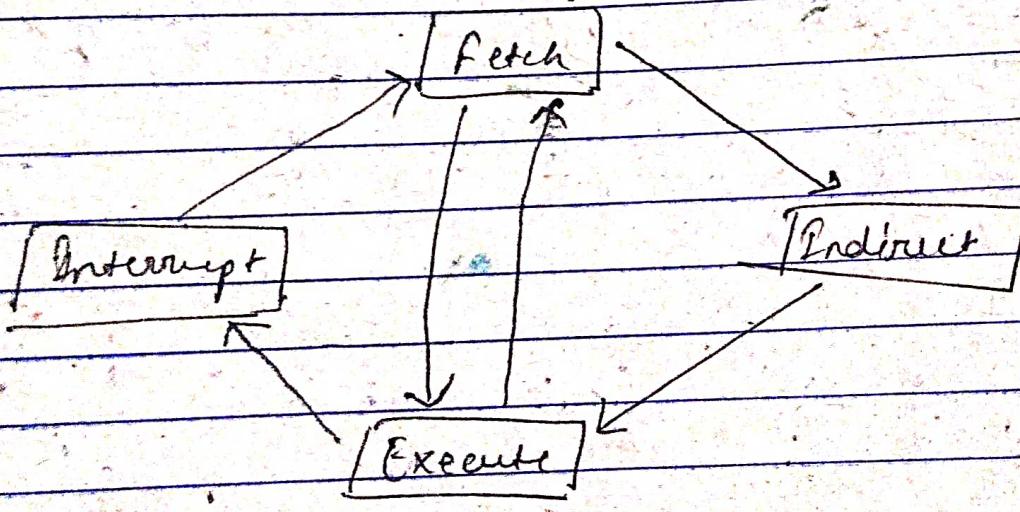
ADD R_2, C, D $R_2 = M[C] + M[D]$

MUL X, R_1, R_2 $M[X] = R_1 * R_2$

Instruction Cycle

Registers involved in Instruction Cycle:

- i) Memory Address Register (MAR): It specifies the address in memory for a read or write operation.
- ii) Memory Buffer Register (MBR): It contains the value to be stored in memory or the last value read from the memory.
- iii) Program Counter (PC): Holds the address of the next instruction to be fetched.
- iv) Instruction Register (IR): Holds the last instruction fetched.



Instruction Cycle

Fetch Cycle consists of 3 steps & 4 microoperations

- ① $MAR \leftarrow PC$
- ② $MBR \leftarrow \text{memory}$
- ③ $PC \leftarrow PC + 1$
- ④ $IR \leftarrow MBR$

Instruction Cycle

- ① MAR \leftarrow IR (address)
- ② MBR \leftarrow memory
- ③ PR (address) \leftarrow MBR (address)
- ④ PR (address) \leftarrow MBR (address)

Execute Cycle: Operation / Task is performed in this cycle

- ① MAR \leftarrow IR (address) Ex- ADD R, X
- ② MBR \leftarrow memory
- ③ R \leftarrow (R) + (MBR)

Interrupt Cycle:

- ① MBR \leftarrow PC
- ② MAR \leftarrow Save address
PC \leftarrow Routine Address
- ③ memory \leftarrow (MBR)

Micro Operation Operations executed on data stored in registers
are known as microoperations.

Ex- Shift, Clear, Count, Load, Mul

Types

- i) Data Transfer M.O. $\xrightarrow{\quad}$ Register
- ii) Arithmetic M.O. $\xrightarrow{\quad}$ Memory
- iii) Logical M.O.
- iv) Shift M.O.

Register T.M.O. \rightarrow Transfers data from reg to reg.
Memory T.M.O. \rightarrow Reg to memory
Memory to Reg data transfer.

Arithmetic M.O. \rightarrow Performs arithmetic operations on data (numeric)

Logical \rightarrow Performs bit manipulation

Shift M.O. \rightarrow supply of bits

$\begin{array}{r} 100011 \\ \times 00101 \\ \hline 00011001 \end{array}$ left shift

Instruction execution

- PC register of the processor gives the address of the instruction which needs to be fetched from the memory.
- If the instruction is fetched then, the instruction opcode is decoded. On decoding, the processor identifies the number of operands. If there is any operand to be fetched from the memory, then that operand address is calculated.
- Operands are fetched from the memory. If there is more than one operand, then the operand fetching process may be repeated.
- After this, the data operation is performed on the operands and a result is generated.
- If the result has to be stored in a register, the instruction ends here.

- If the destination is memory, then first destination address has to be calculated. Then the result is stored in memory. If multiple, process repeats.
- Now the current instructions have been executed. Side by-side, the PC is incremented to calculate the address of the next instruction.
- The above instruction cycle then repeats for further instructions.

Program Control Program Control Instructions are the machine code that are used by machine or in assembly language by user to command the processor act accordingly.

- Types:
- i) Compare
 - ii) Unconditional Branch
 - iii) Conditional Branch
 - iv) Subroutines
 - v) Halt/Stop
 - vi) Interrupt

OR

Program Control Instructions modify or change the flow of a program. It is the instruction that alters the sequence of the program's execution, which means it changes the value of the program counter, due to which the execution of the program changes.

Ex- Jump, Skip, Return

JMP SKP

Reduced Instruction Set Computer

The concept of RISC architecture involves an attempt to reduce execution time by simplifying the instruction set.

CISC

- i) Complex Instruction Set Computer
- ii) Large no. of instruction
- iii) Variable length Instruction format
- iv) Large no. of addressing modes
- v) Cost is high
- vi) More powerful
- vii) Several Cycle Instructions
- viii) Manipulation directly in memory
- ix) Microprogrammed Control Unit
- x) Example: Mainframe, Motorola 6800, Intel 8080

RISC

- Reduced Instruction Set Computer
- less no. of instruction
- fixed length Instruction format
- few no. of addressing modes.
- Cost is less
- less Powerful
- Single Cycle Instructions
- Only in Registers (but simple load & store operations for memory access).
- Hardwired Control Unit
- Example: MIPS, ARM, SPARC, RISC

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

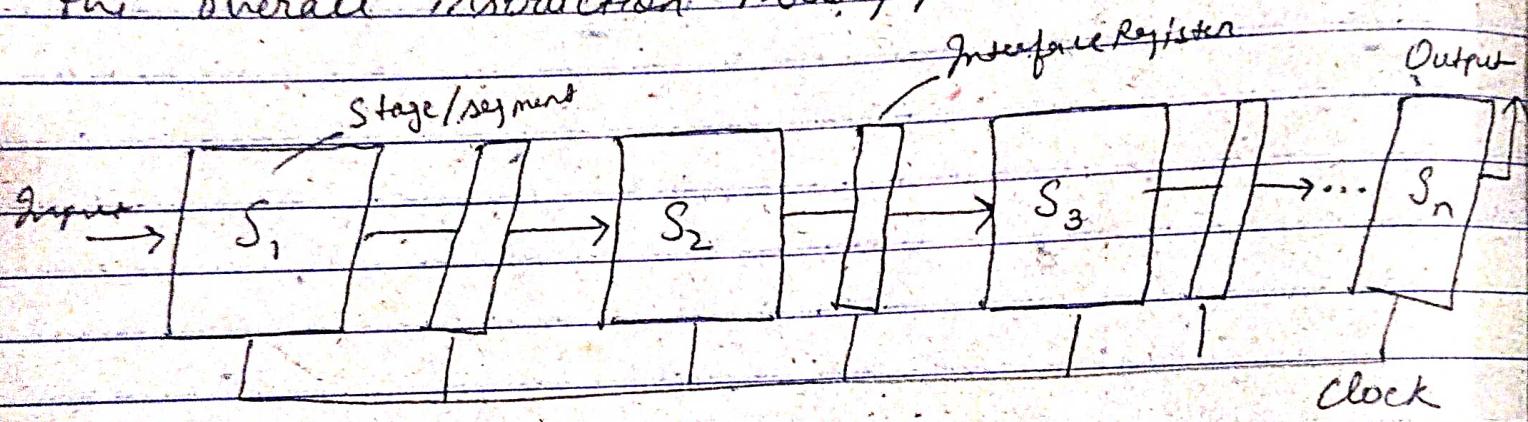
Simultaneous execution of more than one instruction taken

place in pipelined processor

In pipelining multiple instructions overlapped in execution.

OR

Pipelining is a technique where multiple instructions are overlapped during execution. Pipeline is divided into stages and these stages are connected with one another to form a pipe like structure. Instructions enter from one end and exit from another end. Pipelining increases the overall instruction throughput.



Types of Pipelining

- i) Arithmetic
- ii) Instruction

Execution in a pipelined processor: Execution sequence of instructions in a pipelined processor can be visualized by a space-time diagram. For Ex - consider a processor having 4 stages and let there be 2 instructions to be executed.

Non Overlapped Execution

Stage / Cycle	1	2	3	4	5	6	7	8
S1	I ₁		T ₂					
S2		I ₂		P ₂				
S3			I ₃			P ₂		
S4				I ₄	T ₄		P ₂	

Total time = 8 cycles

Overlapped Execution

Stage / Cycle	1	2	3	4	5
S1	I ₁	T ₂			
S2		I ₁	P ₂		
S3			I ₁	P ₂	
S4				I ₁	T ₂

Total time = 5 cycles

Hardwired and Microprogrammed Control

Hardwired :-

A hardwired control unit is a control unit that uses a fixed set of logic gates and circuitry to execute instructions. $Rai \infty$

The control signals for each instruction are hardwired into the control unit, so the control unit has a dedicated circuit for each possible instruction. Hardwired CU are simple and fast, but they can be inflexible and difficult to modify.

Microprogrammed:-

MPCU is a CU that uses a microcode to execute instructions. The microcode is a set of instructions that can be modified or updated, allowing for greater flexibility and ease of modification.

The control signals for each instruction are generated by a microprogram that is stored in memory, rather than being hardwired into control unit.

Dif. b/w hardwired & microprogrammed

	HW	MP
Implementation	fixed set of logic gates and circuits	Microcode stored in memory.
Flexibility	less flexible, difficult to modify	More flexible, easier to modify.
Instruction Set	Support limited instruction sets	Supports complex instruction sets.
Complexity of Design	Simple design, easy to implement	Complex design, more difficult to implement
Speed	fast operation	Slower operation due to microcode decoding

Debugging and Testing	Difficult to debug and test	Easier to debug and test.
Size and cost	Smaller size, lower cost	Large size, higher cost.
Maintenance & Upgradability	Difficult to upgrade and maintain	Easier to upgrade & maintain.

Types of Microprogrammed

i) Horizontal Micro-programmed CU

The control signals are represented in the decoded binary format that is 1 bit / CS.

means 1 bit for every Control Signal (CS)

- It supports longer control words.
- It is used in parallel processing applications.
- It requires no additional hardware (decoders). It means it is faster than vertical MP.
- It allows a higher degree of parallelism. If degree is n , n CS is enabled at a time.

ii) Vertical MP CU

The control signals are represented in the encoded binary format. For N control signals - $\log_2(N)$ bits are required.

- It allows a lower degree of parallelism i.e., the degree of parallelism is either 0 or 1.

Unit - 4 Memory :

- Basic concept and hierarchy ✓
- Semiconductor RAM memories ✓
- 1D & 2½ D memory organization ✓
- ROM memories ✓
- Cache memories
 - * Concept ✓
 - * Design Issues ✓
 - * Performance ✓
- Address Mapping & replacement ✓
- Auxiliary memory ✓
 - * Magnetic disk
 - * Magnetic tape
 - * Optical disks
- Virtual memory ✓
 - * Concept, implementation

Memory of computer is an off-work device that accept data processes on that

Computer memory is just like the human brain. It is used to store data/info. and instructions. It is a data storage unit or a data storage device where data is to be processed and instructions required for processing are stored. It can store both the input and output can be stored here.

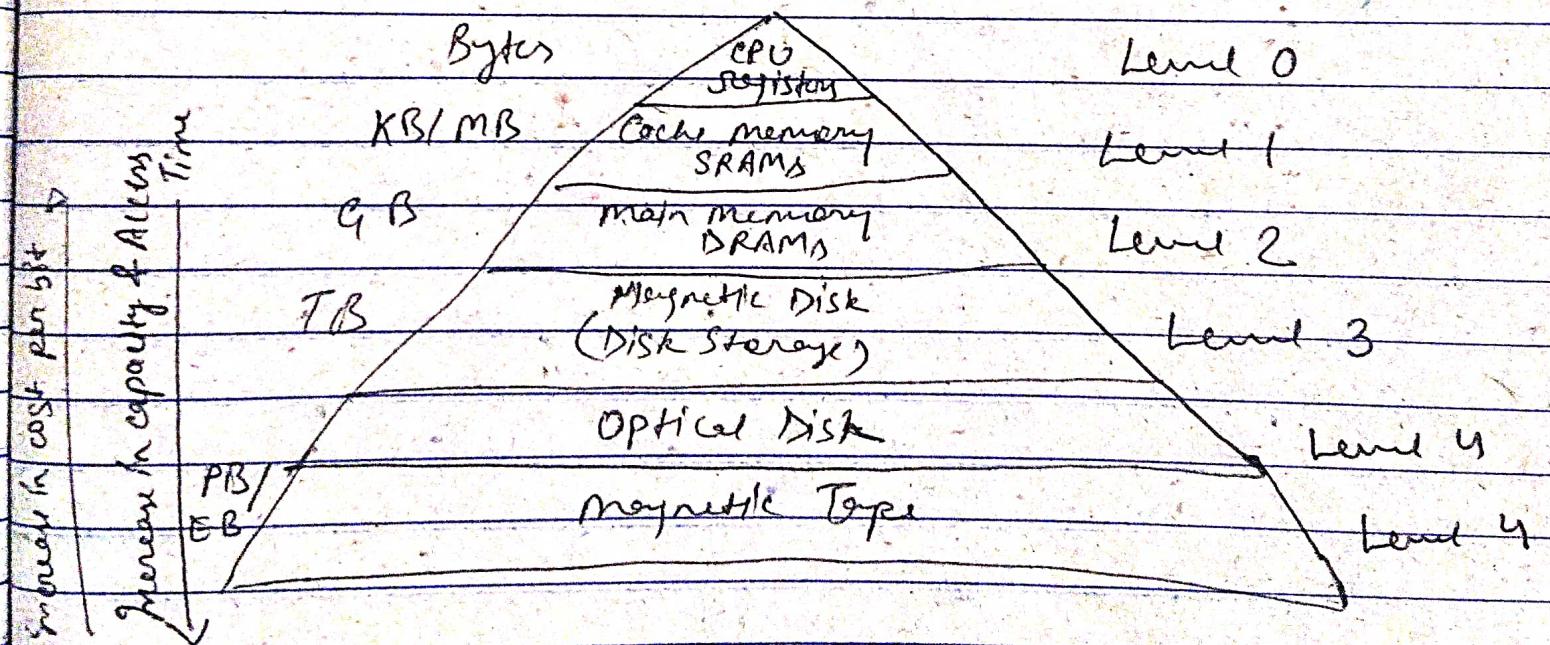
Types of Computer memory

- i) Primary
- ii) Secondary
- iii) Cache

Memory Hierarchy: Memory Hierarchy is one of the most required things in Computer memory as it helps in optimizing the memory available in the computer. There are multiple levels present in the memory, each having a diff. size; diff. cost, etc. ~~Some types of~~

Types of Memory Hierarchy

- i) External memory or Secondary memory: Comprising of magnetic disk, Optical disk & magnetic tape i.e., peripheral storage devices which are accessible by the processor via an I/O module.
- ii) Internal or Primary: Comprising of main memory, Cache memory & CPU registers. This is directly accessible by the processor.



1. Registers are small, high speed memory units located in the CPU.

They are used to store the most frequently used data and instructions. Registers have

Registers have the fastest access time and the smallest storage capacity ranging from 16 to 64 bits.

2. Cache memory is a small, fast memory unit located close to the CPU.

It stores frequently used data and instructions that have been recently accessed from the main memory; Cache memory is designed to minimize the time it takes to access data by providing the CPU with quick access to frequently used data.

3. Main memory also known as RAM is the primary memory of a computer system. It has a larger storage capacity than cache memory, but it is slower. Main memory is used to store data and instructions that are currently in use by the CPU.

Types: SRAM

DRAM

4. Secondary Storage: It is a non-volatile memory unit that has a larger storage capacity than main memory. It is used to store data and instructions that are not currently in use by the CPU. EX- HDD, SSD

5. Magnetic Disk are simply circular plates that are fabricated with either a metal or a plastic or a magnetized material.

The magnetic disks work at a high speed inside the computer and these are frequently used.

6. Magnetic Tape is simply a magnetic recording device that is covered with a plastic film. It is generally used for the backups of data.

Semiconductor Memory

A type of electronic memory is known as semiconductor memory. Stores digital data by making use of semiconductor materials, most commonly silicon.

Data is stored in binary format in this memory with '1s' & '0s' representing electrical charges.

Two types: RAM
ROM

Random Access memory is a form of semiconductor memory technology that is applied for reading and writing data in any order.

Types of RAM

- i) SRAM
- ii) DRAM

Static RAM

- Characterized by high speed and high cost
- It has larger storage
- It takes less power to perform
- Use six transistors to store data
- Can accept one command and transfer one word of data per clock cycle.

Dynamic RAM

- It is used for main memory
- It has high power consumption
- It is slower memory in comparison to SRAM
- It is less costly
- Access time 60 - 70 nanoseconds

Read Only Memory is a form of semiconductor memory technology used where the data is written once and then not changed.

It is used where data needs to be stored permanently, even when the power is removed.

Types of ROM : PROM

EPROM

EEPROM

PROM :

- Data can be programmed once after manufacturing.

- Programming involves blowing fuses within the chip to represent binary ones.
- Once programmed, data becomes permanent and cannot be changed.
- Used for situations where firmware might need minor adjustments after production.

EPROM

- Can be programmed and erased multiple times with UV light.
- Data is stored in floating gate transistors that lose their charge when exposed to UV.
- Offers more flexibility than PROM for development and prototyping.
- Erasure process can be slow and inconvenient.

EEPROM

- Most versatile type of ROM, allowing data to be programmed and erased electrically.
- No need for UV light, making it faster and more convenient than EPROM.
- Widely used in embedded systems and microcontrollers for storing configuration data and firmware updates.
- Has limited write endurance, meaning the number of erase/write cycles is finite.

2D & 2^{1/2}D Memory Organization

The memory is in the form of a multidimensional array of rows and columns, in which each cell stores a bit and a complete row contains a word.

A memory can be divided into the mentioned form:

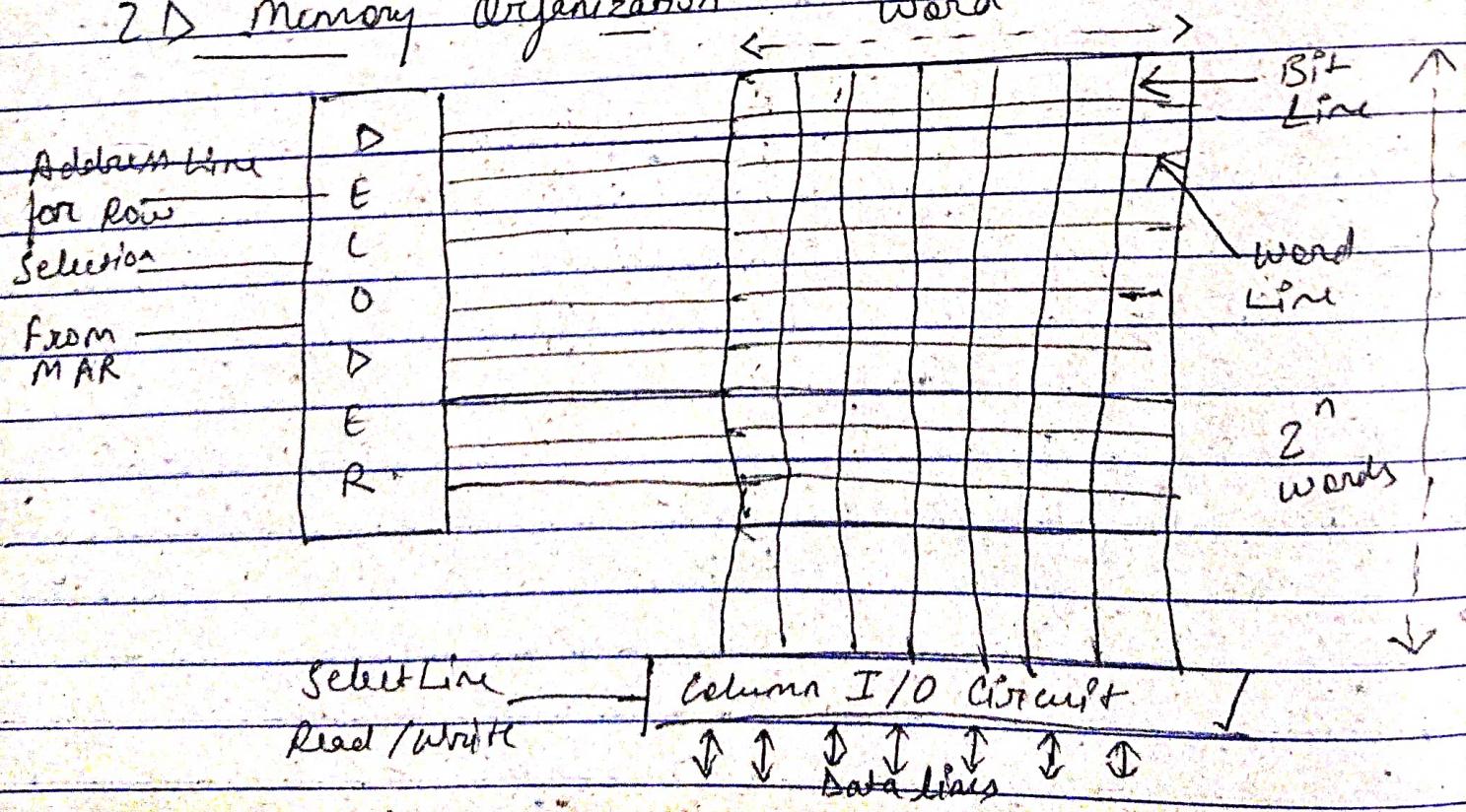
$$2^n = N$$

n = no. of Address lines

N = Total memory in bytes

There will be 2^n words.

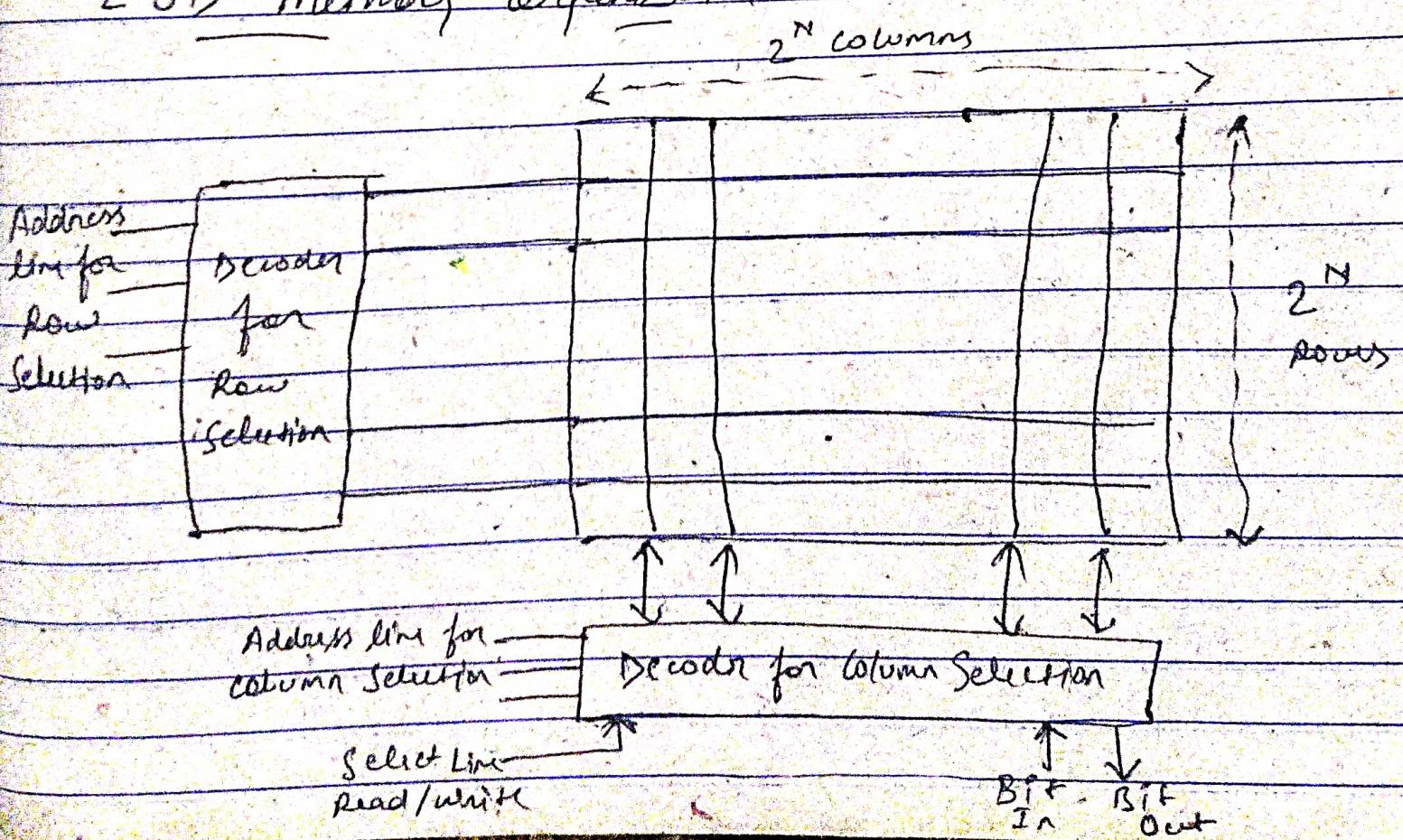
2D Memory Organization



Decoder is mainly used for selection of rows.

- In 2D array memory is divided in the form of rows & columns (matrix).
- Each row contains a word.
- Here decoder is used, a decoder is a combinational circuit which contains n input lines and 2^n output lines.
- One of the output line will select the row by the address contained in the MAR.
- The word which is represented by that row will get selected and is either read or write through the data lines.

2.5) Memory Organization



- 2.5D Organization same as 2D organization but two diff. decoders are used, one is column decoder and another is row decoder.
- Column Decoder used to select the column and row decoder is used to select the row.
- Address from the MAR will go in decoders' input
- Decoders will select the respective cell through the bit outline, then the data from that location will be read or through the bit in line data will be written at that memory location.

Read and Write Operation

- If the select line is in reading mode then the word bit which is represented by the MAR will be available to the data lines and will get read.
- If the select line is in write mode then the data from the memory data register (MDR) will be sent to the respective cell which is addressed by the memory address register (MAR).

Comparison

2D

- i) Hardware is fixed
- ii) Requires more no. of gates
- iii) More complex
- iv) Relatively diff. to implement
- v) Error correction is not possible

2.5D

- i) Hardware can be changed
- ii) Requires less no. of gates
- iii) Less complex
- iv) Easy to implement
- v) Error correction could be done easily

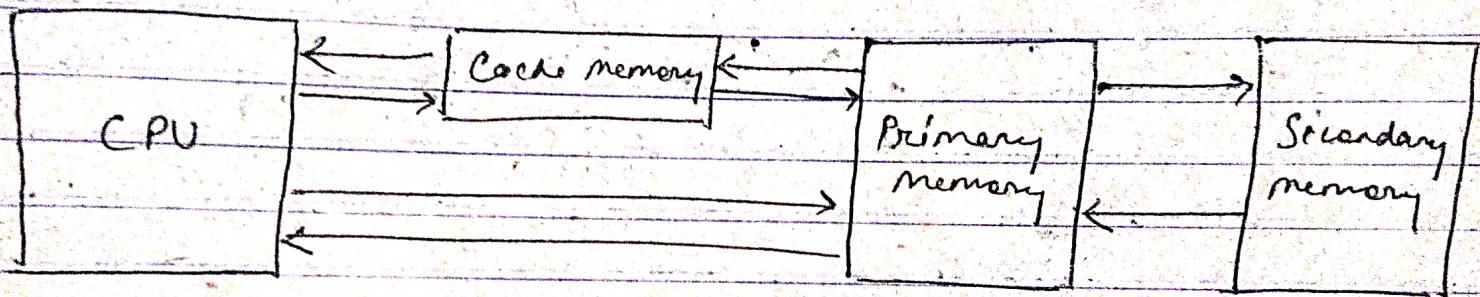
v) More difficult to fabricate
vi) ROM circuits rarely uses it

Relatively easy to fabricate
Currently most of RAM circuits uses it.

Cache Memory

Cache memory is a special very high speed memory.
The cache is a smaller and faster memory that stores copies of the data from frequently used main memory locations.

The most important use of cache memory is that it is used to reduce the average time to access data from the main memory.



Design Issues

- i, Limited Capacity
- ii, Increased Complexity
- iii, Cache Consistency Issues

Cache Performance when the processor needs to read or write a location in the main memory, it first checks for a corresponding entry in the cache.

- If the processor finds that the memory location is in the cache, a Cache Hit has occurred and data is read from the cache.
- If the processor does not find the memory location in the cache, a Cache Miss has occurred.
For a cache miss, the cache allocates a new entry and copies in data from the main memory, then the request is fulfilled from the contents of the cache.

The performance of cache memory is frequently measured in terms of a quantity called Hit Ratio.

$$\text{Hit Ratio} = \frac{\text{hit}}{\text{hit + miss}} = \frac{\text{no. of hits}}{\text{total accesses}}$$

$$\text{Miss Ratio} = \frac{\text{miss}}{\text{hit + miss}} = \frac{\text{no. of misses}}{\text{total accesses}} = 1 - \text{hit ratio}$$

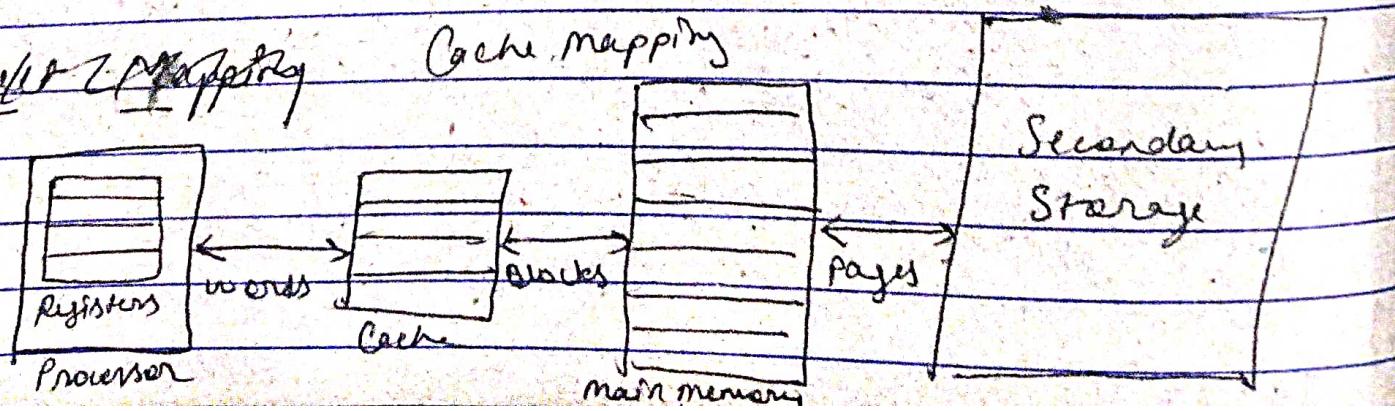
Cache Mapping

Cache mapping is a technique by which the contents of memory are brought into the cache memory.

- Direct
- Associative
- Set-Associative

Different Mapping

Cache Mapping



Direct mapping

PAGING
DATA

Cache

L ₀	B ₀ B ₁ ... B ₂₈
L ₁	B ₁ B ₅ ... B ₂₉
L ₂	B ₂ B ₆ ... B ₃₀
L ₃	B ₃ B ₇ ... B ₃₁

$$\frac{16 \text{ bytes}}{4} = 4 \text{ lines}$$

W ₀ W ₁ W ₂ W ₃	B ₀
W ₄ W ₅ W ₆ W ₇	B ₁
W ₈ W ₉ W ₁₀ W ₁₁	B ₂
W ₁₂ W ₁₃ W ₁₄ W ₁₅	B ₃
⋮	⋮
W ₁₁₂ W ₁₁₃ W ₁₁₄ W ₁₁₅	B ₃₁

$$\text{lines size} = \text{block size}$$

$$\frac{128 \text{ bytes}}{4} = 32 \text{ blocks}$$

$$j = K \bmod n$$

Cache no. Block no. no. of lines

$$B_0 \quad 0 \bmod 4$$

$$B_1 \quad 1 \bmod 4$$

$$B_2 \quad 2 \bmod 4$$

$$B_3 \quad 3 \bmod 4$$

$$B_4 \quad 4 \bmod 4$$

P.A.		main memory
5	2	

Block no. Block offset (size)

$$4 = 2^2 \rightarrow \text{size}$$

$$128 = 2^7 \rightarrow \text{total}$$

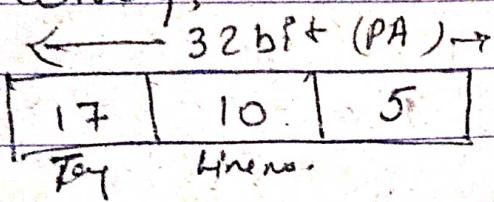
Cache memory	3 2 2
Tag	Line no. Block offset

The simplest technique known as direct mapping, maps each block of main memory into only one possible cache line.

If a line is previously taken up by a memory block when a new block needs to be loaded, the old block is trashed.

Associative mapping

Ques. Consider a direct mapped cache of size 32 KB with block size 32 bytes. A CPU generates 32 bit addresses. The no. of bits required for cache indexing and tag bits respectively,



line size = Block size

← Block no. → Block offset (size)

line size = $\log_2 32 \text{ KB}$
(bits reqd. to represent)
 $= \log_2 32 \text{ B}$

$$\text{Block size} = \log_2 32 \\ (\text{bits reqd. to represent}) = 5$$

$$= \log_2 1024$$

$$= 10$$

Cache Indexing (bits required to represent line no.)

$$= 10$$

$$\text{Tag bits} = 17$$

Associative Mapping / Fully Associative Mapping

A block of main memory can be mapped to any freely available cache line.

This makes fully associative mapping more flexible than direct mapping.

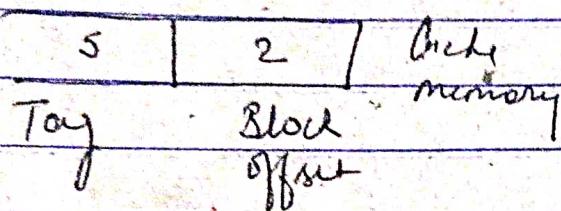
A replacement algorithm is needed to replace a block if the cache is full.

main memory

Cache

L_0	$B_0 B_1 \dots B_{31}$
L_1	$B_0 B_1 \dots B_{31}$
L_2	$B_0 B_1 \dots B_{31}$
L_3	$B_0 B_1 \dots B_{31}$

$w_0 w_1 w_2 w_3$	B_0
$w_4 w_5 w_6 w_7$	B_1
$w_8 w_9 w_{10} w_{11}$	B_2
$w_{12} w_{13} w_{14} w_{15}$	B_3



$w_{16} w_{17} w_{18} w_{19}$, B_3

$\leftarrow PA \rightarrow$	
5	2
Block no.	Block offset

Line size = Block size

Set Associative Mapping / K-way Set Associative Mapping

Main memory

Cache

L_0	$B_0 B_2 B_4 \dots B_{30}$	S_0
L_1	$B_0 B_2 B_4 \dots B_{30}$	
L_2	$B_1 B_3 B_5 \dots B_{31}$	S_1
L_3	$B_1 B_3 B_5 \dots B_{31}$	

16 bytes/bytes

$w_0 w_1 w_2 w_3$	B_0
$w_4 w_5 w_6 w_7$	B_1
$w_8 w_9 w_{10} w_{11}$	B_2
$w_{12} w_{13} w_{14} w_{15}$	B_3

K way set

no. of set = no. of line

$w_{16} w_{17} w_{18} w_{19}$, B_3

no. of set given

K

128 words/bytes

let 2 way set

$$\frac{\text{no. of set}}{\text{set}} = \frac{4}{2} = 2$$

It is mixture of direct & associative.

We use $K \bmod n$

Block no. no. of set

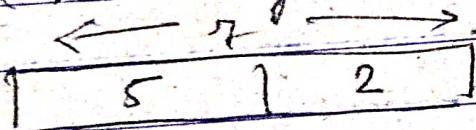
Here block will go to cache acc to set no.

$$B_0 \bmod 2$$

$$= 0$$

so Block B_0 will go to Set 0
means can go to any line L₀ or L₁

Physical Address of Main Memory



Block no., Set no., Block offset

Physical Address of Cache Memory



Tag, Set no., Block offset

Set associative mapping allows that each word that is present in the cache can have two or more words in the main memory for the same index address.

Set associative cache mapping combines the best of direct and associative cache mapping techniques.

Cache Replacement Algos.

- i) FIFO First In First Out
- ii) LRU Least Recently Used
- iii) MRU Most Recently Used

Motive of Cache Replacement is
Miss Penalty Reduce
Hit Increase

Ques. Consider a fully association cache with 8 cache blocks (0-7) and the following sequence of memory blocks requests:

M M H H M H H H
4, 3, 25, 8, 19, 6, 25, 8, 16, 38, 45, 22, 8, 3, 16, 25, 7

If LRU replacement policy is used which cache block will have memory block 7?

memory Block 7
is in

Line 5

0	45
1	22
2	25
3	8
4	3
5	7
6	16
7	95

Ques. Set Association Mapping

A 4-way set associative cache memory with a capacity of 16 KB is built with using a block size of 8 words. The word length is 32 bits. The size of physical address space is 4 GB. The no. of bits for the tag field is _____?

Ans

Main memory space

$$= 4 \text{ GB}$$

$$= 2^2 \cdot 2^{30}$$

$$= 2^{32}$$

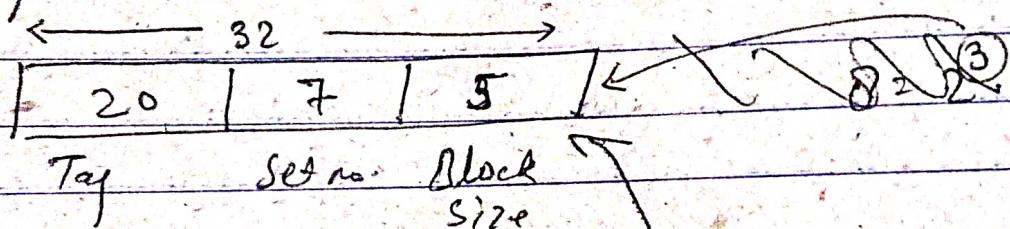
$$K \rightarrow 2^{10}$$

$$M \rightarrow 2^{20}$$

$$C \rightarrow 2^{30}$$

$$T \rightarrow 2^{40}$$

Physical address bits = 32



Word length = 32 bits

= 4 Bytes

Total Block size = 4×8

$$= 32 \text{ B}$$

$$32 = 2^5$$

Cache size = 16 KB

$$\text{no. of lines} = \frac{16 \text{ KB}}{32 \text{ B}} \quad \begin{matrix} \text{Cache size} \\ \text{Block size} \end{matrix}$$

$$= \frac{2^4 \times 2^{10}}{2^5}$$

$$= 2^9$$

$$\text{No. of sets} = \frac{\text{Total no. of lines}}{k(4)}$$

$$= \frac{2^9}{2^2}$$

$$= 2^7$$

So bit to represent no. of sets = 7

So no. of bits for the tag = 20

Auxiliary Memory also known as secondary memory and it is used to store data and programs permanently. In this memory data and programs are retained when the power is turned off. So this memory is non-volatile.

- i) Magnetic Disk
- ii) Magnetic Tape
- iii) Optical Disk

Virtual Memory

A virtual memory block is known as Page, and a Virtual Memory error is known as Page fault.

Virtual memory is a concept used in some large

PAGE NO.
DATE:

computer systems that permit the user to construct programs as though a large memory space were available.

- It gives programmers the illusion that they have a very high memory although the computer has a small main memory.
- Virtual memory is the partition of logical memory from physical memory.
- It creates the function of programming easier because the programmer no longer requires to worry about the multiple physical memory available.
- It maps memory addresses used by a program called virtual addresses into physical addresses in computer memory.

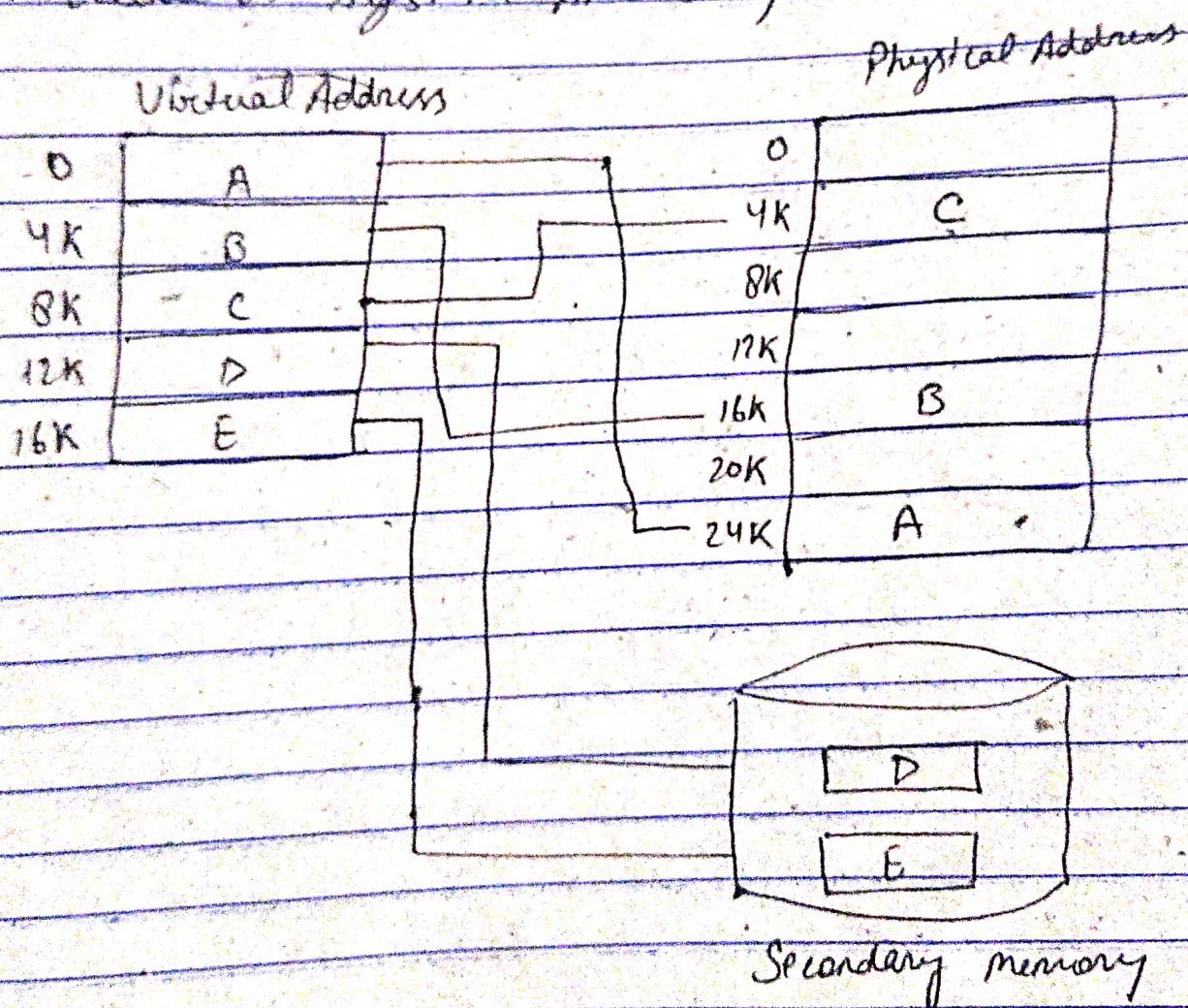
Advantages

- Large programs can be written, as virtual space available is huge compared to physical memory.
- Less I/O required, leads to faster and easy swapping of processes.
- More physical memory available, as programs are stored on virtual memory, so they occupy

very less space on actual physical memory.

Disadvantages

- Applications run slower if the system is using virtual memory.
- It takes more time to switch between applications.
- Less hard drive space for your use.
- It reduces system stability.



UNIT - 5

Input / Output:

- Peripheral devices ✓
- P/I/O interface —
- P/I/O ports —
- Interrupts:
 - * Hardware ✓
 - * Types ✓
 - * exceptions
- Modes of Data Transfer
 - * Programmed P/I/O
 - * Interrupt P/I/O
 - * Direct Memory Access
- P/I/O channels & processors
- Serial Comm:
 - * Synchronous Comm
 - * Asynchronous Comm
- Standard Comm Interfaces.

Peripheral Devices: A peripheral device is defined as a device that provides input/output functions for a computer and serves as an auxiliary computer device without computing-intensive functionality.

- A peripheral device is also called a peripheral, computer peripheral, input-output device or I/O device.
- Connected to a computer system but is not part of the core computer system's architecture.
- Ex:- Keyboards, display units, etc.

Types :-

- i) Input
- ii) Output
- iii) Storage

Input Device: It converts incoming data and instructions into a pattern of electrical signals in binary code that are comprehensible to a digital computer.

Ex - Keyboard, Mouse, Scanner, Microphone, etc.

Output Device: i) An output device is generally reverse of the input process.

ii) It translates the digitized signals into a form understandable to the user.

iii) These devices can send data from one computer system to another.

Ex - Monitor, Headphone, etc.

Storage Device: i) They are used to store data in system which is required for performing any operation in system.

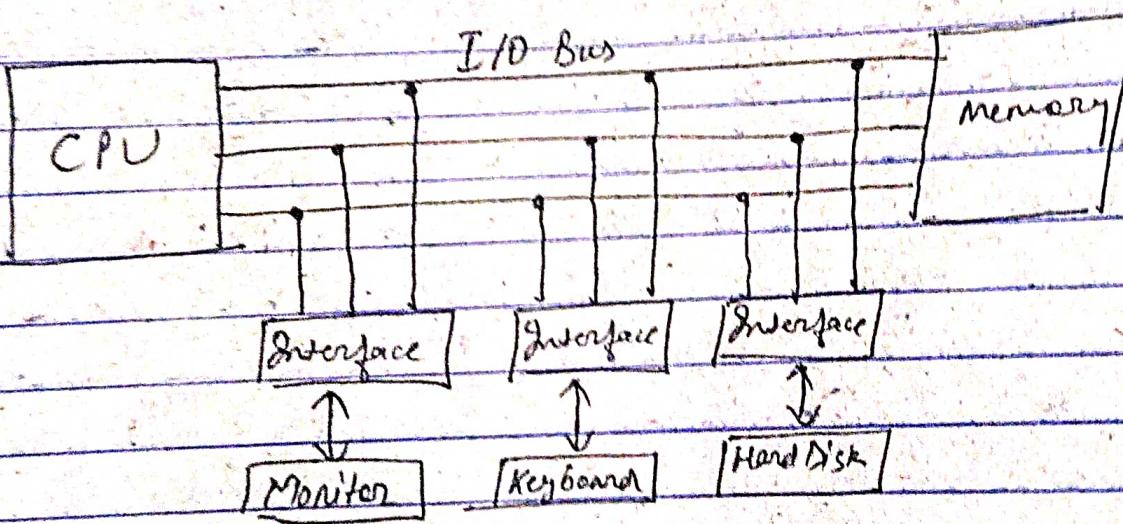
ii) It is very important device and it provides better compatibility also.

Ex - Hard disk, Magnetic Tape, etc.

I/O Interface

Interface is a shared boundary between two separate components of the computer system which can be used to attach two or more components to the system for communication purposes.

Input/Output Interface is used as an method which helps in transferring of info. between the internal storage devices i.e. memory and external peripheral device.



Connection of I/O Bus to I/O Devices

The commands that Interface may receives any of the following are:

i) Control: A command control is given to activate the peripheral and to inform it what to do.

ii) Status: A status command can test multiple test conditions in the interface and the peripheral.

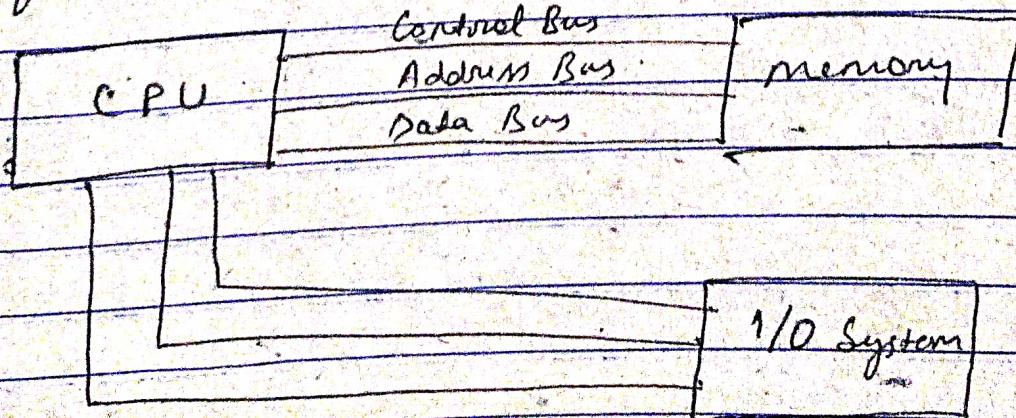
iii) Data Output: A data output command creates the interface counter to the command by sending data from the bus to one of its registers.

iv) Data Input: It is opp. to the data output command here the interface gets an element of data from the peripheral and places it in its buffer register.

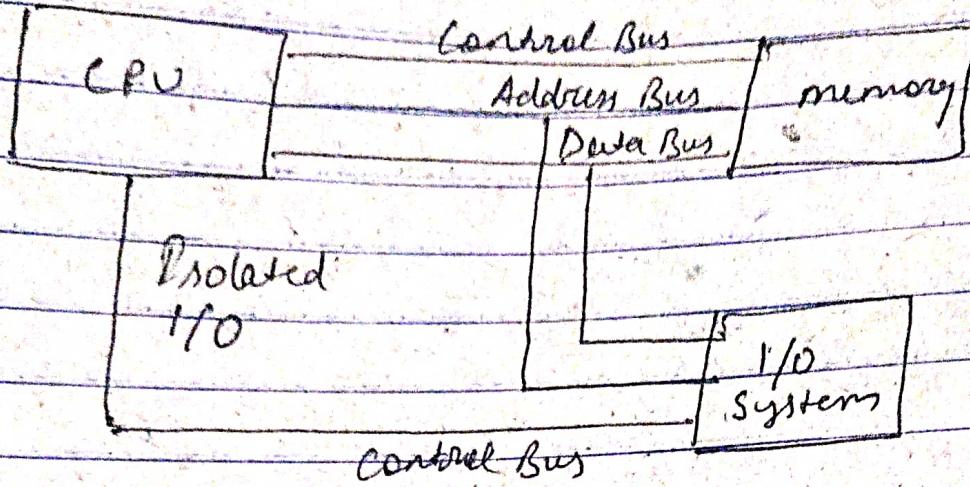
CPU needs to communicate with the various memory and I/O devices, as we know data between the processor and these devices flow with the help of system bus.

There are three ways in which system bus can be allotted to them:

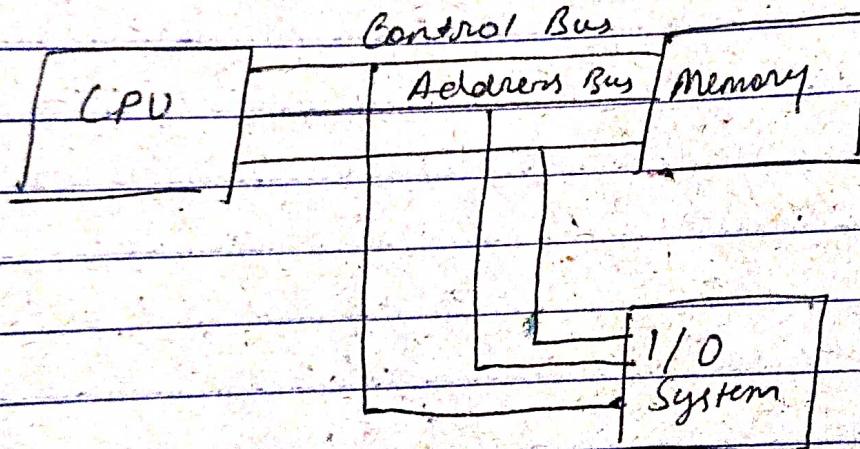
i) Separate set of buses, one for memory & other for I/O.



iii) Common Address and Data bus, but separate control lines



iii) Common Address, data & control bus



I/O Port

The connection point acts as an interface between the computer and external devices like printers, modem etc.

There are two types of ports:

- i) Internal Port
- ii) External Port

Internal Port: It connects the system's motherboard to internal devices like hard disk, CD drive, internal bluetooth, etc.

External Port: It connects the system's motherboard to external devices like a mouse, printer, USB, etc.

Some important ports are:

- i) Serial Port
- ii) Parallel Port
- iii) USB Port
- iv) Firewire Port
- v) Ethernet Port

Interrupts The interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention.

It alerts the processor to a high-priority process requiring interruption of the current working process.

When a device raises an interrupt at let's say process i, the processor first completes the execution of instruction i. Then it loads the Program Counter (PC) with the address of the first instruction of the ISR (Interrupt Service Routine). Before loading the Program Counter with the address, the address of the interrupted instruction is moved to a temporary location. Therefore, after

handling the interrupt the processor can continue with process it.

While the processor is handling the interrupts, it must inform the device that its request has been recognised so that it stops sending the interrupt request signal.

Hardware Interrupt : In a hardware interrupt, all the devices are connected to the Interrupt Request Line. A single request line is used for all the n devices. In hardware interrupt, peripheral devices or any hardware device is responsible for interrupt.

Types of Interrupt

- i) Hardware Interrupt
 - External Interrupt
 - Internal Interrupt

ii) Software Interrupt

The external interrupt occurs when a specified signal is input to the dedicated external interrupt terminal. The internal interrupt occurs by an interrupt request signal from a peripheral circuit built into the microcontroller.