

Have You Made Full Use of the OCR Feature?

Make a scan, enhance it and save it. Are these all the features you know about CamScanner? If so, you have missed too many cool experiences. CamScanner offers you lots of features rather than scanning. What we are sharing today is the OCR(Optical Character Recognition) feature.

What can you do with OCR feature?

1. Searching

What can you do if you want to search for a document but just can't remember the names of some docs? Use this feature to recognize all the texts on your scans. Next time you just need to enter some key words in the search box and all the documents within the words will be found.

2. Text extraction

Just purchase the one-time paid version and you can enjoy the text extraction for lifetime! Ever want to edit some texts on a paper document or a PDF file? Import it into CamScanner and all texts can be extracted as .txt file after OCR!

Why wait? Follow the steps to start using OCR!

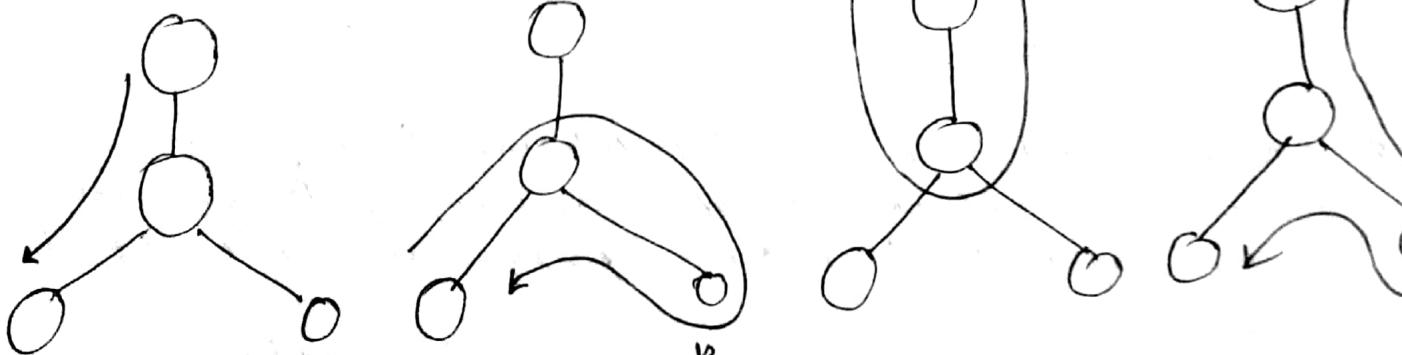
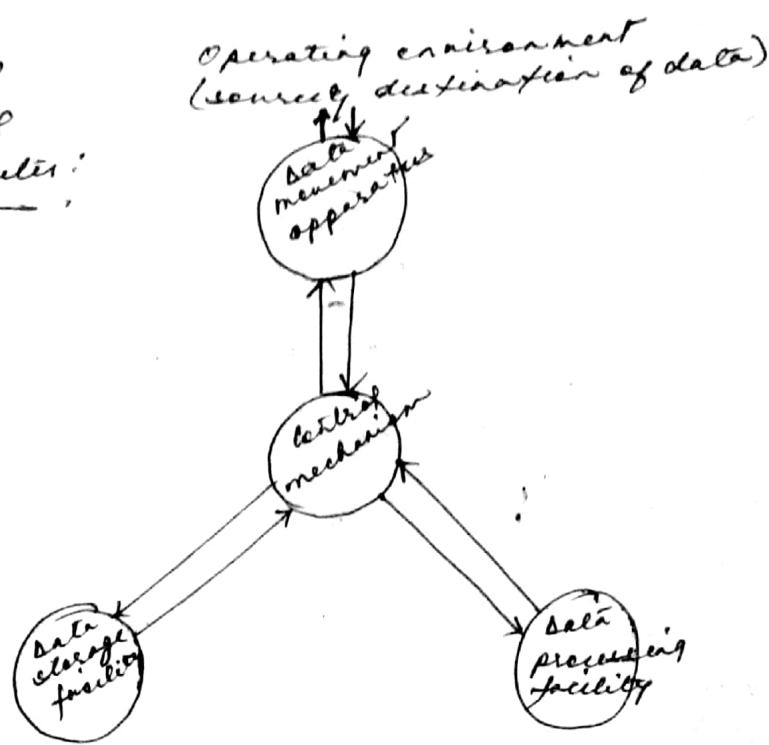
1. Sign in to CamScanner to sync all your docs -> All texts will be auto recognized after syncing.
2. If you don't want to sign in, you can open one single page of any doc-> Tap the Recognize button -> All recognized texts will be shown in a dialog box-> Tap Share to export the texts.

COMPUTER ORGANIZATION

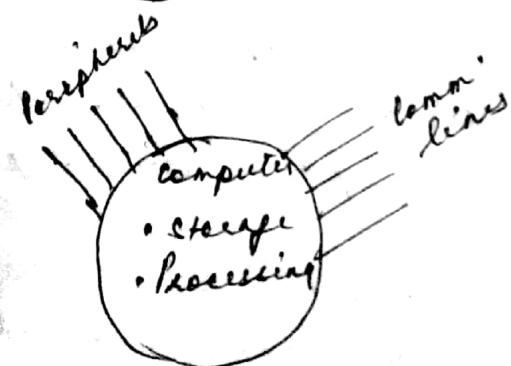
(1)

Module 3:

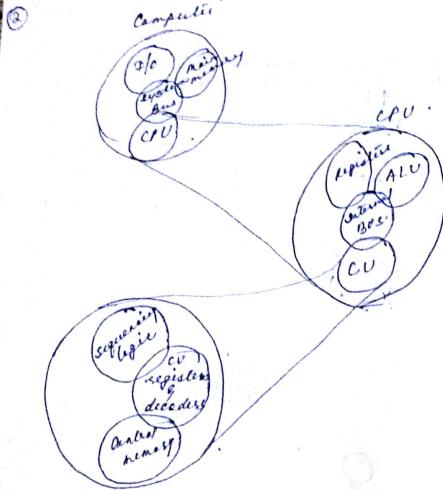
Functional
view of
computer:



Simple depiction



Computer top level architecture:

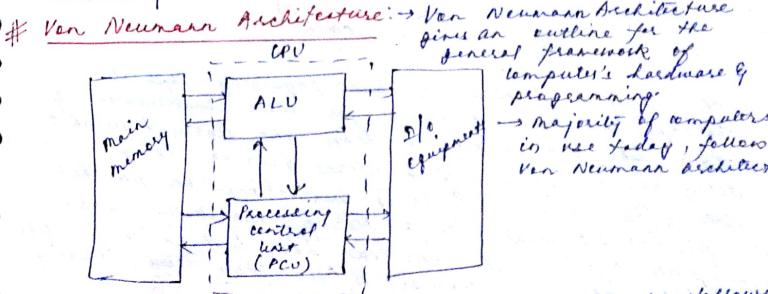


ENIAC (Electronic Numerical Integrator & Computer), designed & constructed at the University of Pennsylvania, is the world's first general purpose electronic digital computer.

→ Storing data in ENIAC was a tedious task.
So, mathematician John von Neumann & Alan Turing came up with the concept of stored program.

→ In 1946, Neumann & his colleagues started working on the design of stored program computer, popularly known as IAS, at the Princeton Institute for Advanced Studies. IAS, though not completed until 1952, is the prototype of all subsequent general purpose computers.

Stored program concept: The stored program concept holds that the data as well as the instruction to manipulate the data, should be stored together in the same area of the memory. The instructions should be carried out sequentially, one instruction at a time. The sequential execution programming imposes a cost of speed limit on the program execution, since only one instruction at a time can be handled by the computer's processor. It means that the CPU can either be reading an instruction or reading/writing data from/to the memory. Both cannot occur at the same time as both data & instructions use the same signal pathways of memory. The speed limit is also called Von Neumann bottleneck.



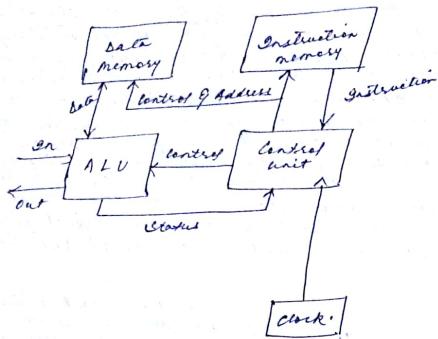
The IAS computer, with Von Neumann style, has the following parts:

- Main memory, which stores both data & instruction
- ALU, capable of operating on binary data
- CU, which interprets the instructions in memory & causes them to be executed
- I/O equipments operated by the CU

→ Despite of all chip developments, Von Neumann's stored program principle still prevails.

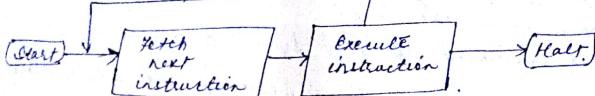
Harvard Architecture:

(9)



- The Harvard Architecture has physically separate storage and signal pathways for their instructions & data. It was originated from the Harvard Mark 2 relay based computer which stores data in relay latches (33 bit wide) & stores instructions on punched tapes (33 bit wide). In a Harvard architecture based computer system, the CPU can fetch both an instruction and data from the memory at the same time, leading to double the memory BW.
- Microcontroller based computer system & DSP based computer system are examples of Harvard Architecture.

Operation sequence for execution of a program:



- A list of sequential instruction may constitute a computer program in memory.
- The Program Counter (PC) points to the most recent instruction to be executed.

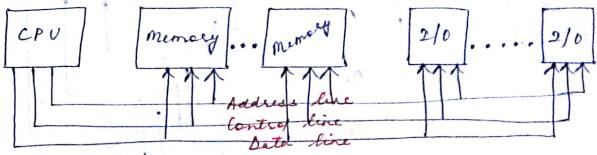
→ After an instruction is fetched in (IR) for execution, PC (5) automatically updates itself to point to the next instruction.

→ Sequential execution is done.

Basic execution of each instruction has three phases

- Fetch it from the main memory
- Decode the instruction
- Execute instruction

BUS



→ Bus has three main lines: Address, Control, Data, + Power lines

→ All three are bidirectional
A system bus consists, typically, of from about 50 to hundreds of separate lines. Each line is assigned a particular meaning or function.

→ Address line: 16, 32 or 64 lines depend on word size
Control line: depends on units controlled
Data lines: depends on output word size

Bus Architecture with Tri-state logic:

Bus system can be made with tri-state buffer.
Graphic symbol of tri-state buffer is as follows:

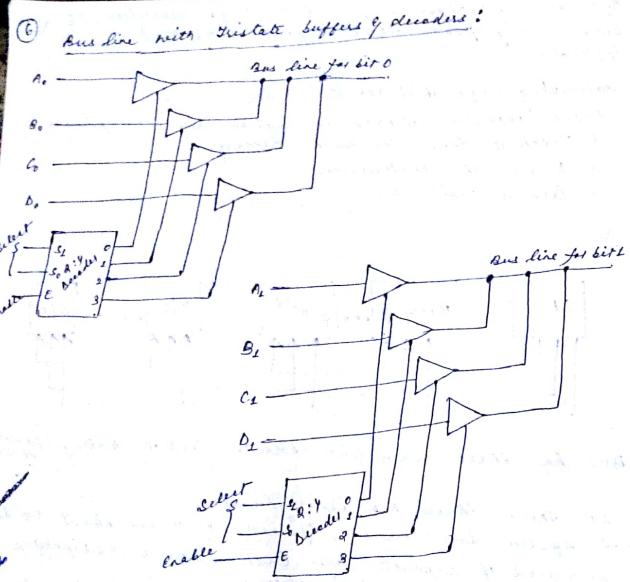
Normal inputs: $I = A$ if $C = 1$
High impedance: $I = Z$ if $C = 0$.

Address input C

Truth Table

A	C	I
0	0	Z
1	0	1
0	1	0
1	1	1

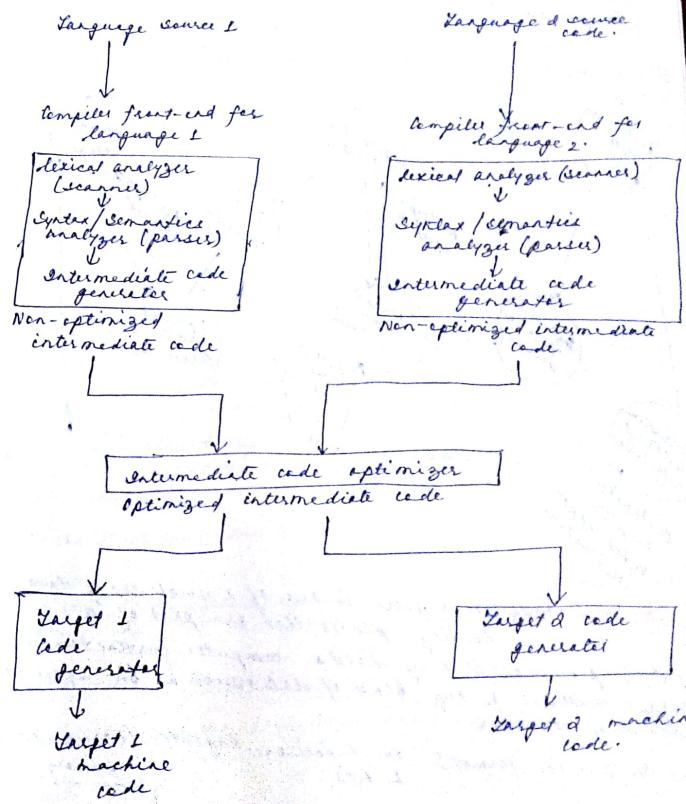
Z = High impedance,
neither 0 nor 1
Tri-state buffer
sets its value



Operating System:

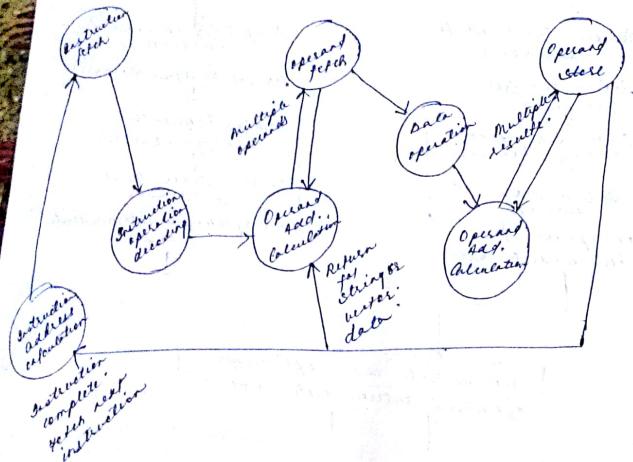
- The operating system of a processor is the system software that controls the execution of programs on a processor & that manages the processor's resources.
- A number of important functions performed by OS, including process scheduling & memory management, can be performed efficiently & rapidly if the processor's hardware has capabilities to support the OS.
- Virtually, all the processor's have this capability to a greater or lesser extent, including virtual memory management hardware and process management hardware.
- The hardware includes special purpose registers & buffers, as well as, circuitry to perform basic resource management tasks.

Compiler: A compiler is a computer program (or set of programs) that transforms instructions written in programming language (the source language) to a language that the compiler can comprehend (the destination language), usually in binary form known as object code.



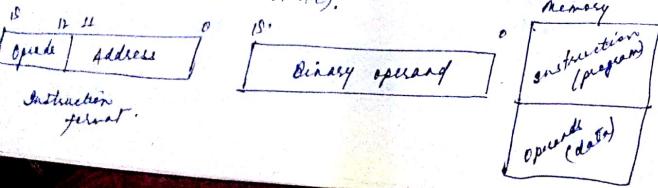
Assembler: An assembler is a program that takes up basic computer instructions & converts them into a pattern of bits that the computer's processor can use to perform its basic operations. Some people call these instructions the assembler language. The others refer to it as assembly language.

Instruction Cycle: Fetch → Decide → Execute.



Registers & Memory: A register is one of a group of data holding places that are part of a computer's processor. It may hold a computer instruction, a storage address or any kind of data, such as bit sequence or individual characters.

Basic instruction format: for Processor Registers (Accumulator, PC AC).



→ List of registers for basic use:

Register	No. of bits	Name	Description
DR	16	Data register	Holds memory operand
AR	12	Address register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction register	Holds instruction operand
PC	12	Program counter	Holds address of instruction
TR	16	Temporary register	Holds temporary data
INPR	8	Input register	Holds input character
OUTR	8	Output register	Holds output character

Instruction format:

- Instruction format consists of the following fields:
 - Operation code**: Specifies the operation to be performed. (e.g. ADD, SUB). The operation is specified by a binary code, known as, operation code or opcode.
 - Source operand reference**: The operation may involve one or more operands, that is, operands that are input for the operation.
 - Result operand reference**: The operation may produce a result.
 - Next instruction reference**: This tells the processor where to fetch the next instruction after the execution of this instruction is complete.

Operands: Operands are represented by abbreviations, called mnemonics. Eg:

ADD	add
SUB	subtract
MUL	multiply
DIV	divide
LOAD	load data from memory
STOR	store data to memory

Instruction Types:

- Data Processing: Arithmetic & Logic instructions
Eg: ADD A,B
 ANI T
- Data storage: movement of data into or out of registers and/or memory locations.
Eg: MOV A,G
- Data movement: I/O instructions
IN A,07
OUT A,07
- Control: Test & Branch instructions
Eg:
 ANI 0E7H
 JNZ L1

$$Y = (A - B) / [C + (D * E)]$$

Sol: • using zero address instruction:

```

PUSH A
PUSH B
SUB B : 100 ← A - B
PUSH D
PUSH E
MUL : 100 ← D * E
PUSH C
ADD : 100 ← C + D * E
DIV : 100 ← (A - B) / [C + (D * E)]
POP Y

```

* Using one address instruction.

LOAD D	Instruction	Comment
	AC ← D	
MUL E	AC ← AC * E	
ADD C	AC ← AC + C	
STOR Y	Y ← AC	
LOAD A	AC ← A	
SUB B	AC ← A - B	
DIV Y	AC ← AC ÷ Y	
STOR Y	Y ← AC	

* Using 2 address instruction

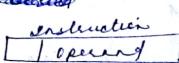
Instruction	Comment
MOV Y,A	Y ← A
SWB Y,B	Y ← Y - B
MOV T,D	T ← D
MUL T,E	T ← T * E
ADD T,C	T ← T + C
DIV Y,T	Y ← Y ÷ T

* Using 3 address instruction

Instruction	Comment
SUB Y,A,B	Y ← A - B
MUL T,D,E	T ← D * E
ADD T,D,C	T ← T + C
DIV Y,T,P	Y ← Y ÷ T

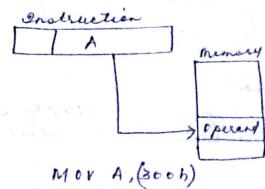
Addressing Modes:

1) Immediate

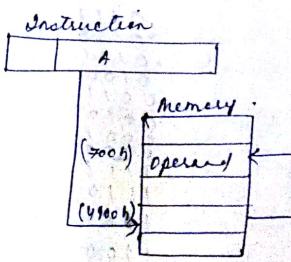


MOV A, 50h

2) Direct



3) Indirect

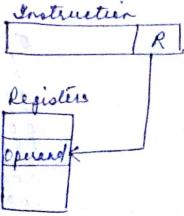


MOV A, [500h]

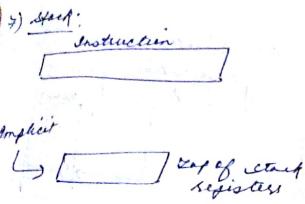
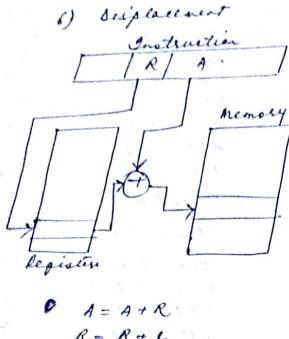
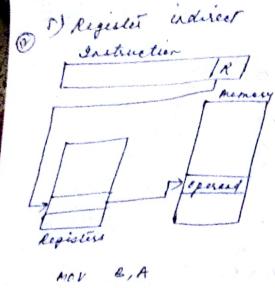
500h ← 700h

700 ← D2h

4) Register



MOV B, A



PUSH A
PUSH B
ADD / add top two
elements/ place
result on stack

Common Number Systems:

Binary rep. Sign: mag.

+8	0000
+7	0001
+6	0010
+5	0011
+4	0100
+3	0101
+2	0110
+1	0111
0	1000
-1	1001
-2	1010
-3	1011
-4	1100
-5	1101
-6	1110
-7	1111
-8	-

2's complement

0111
0110
0101
0100
0011
0010
0001
0000
1111
1110
1101
1100
1011
1010
1001
1000

" To find the complement :

I for -ve no: ① find the binary form of the no.

② Complement the entire no.

③ Add 1

II for the no: just ①

" To find decimal form of 2's complement

I. If it starts with 0

↳ positive numbers

simply determine the decimal form

II. If it starts with 1

↳ negative numbers

find the first '1' bit from right & complement the part left to it.

↳ determine the decimal form.

Unsigned integers, range: 0 - $(2^n - 1)$

Signed " : $-(2^{n-1} - 1) \text{ to } +(2^{n-1} - 1)$

2's complement, " : $-(2^{n-1}) \text{ to } (2^{n-1} - 1)$

Floating point representation:

sign	exponent	mantissa
1	2	9.10

for 32 bit, bias = 127

for 64 bit, bias = 1023 [11 bit | 52 bit] 64

$$g) (40.15625)_{10} = (401000.00101)_2$$

$$= 1.0100000101 \times 2^5$$

$$0 \quad 127 + 5 = 132$$

$$132_{10} = 10000100_2$$

$$\therefore \text{Sol}: \quad \begin{matrix} 0 & \overbrace{10000100}^{\text{(exp)}} & \overbrace{0100000101000\ldots00}^{\text{(mag)}} \end{matrix}$$

0 (+ve no)

1 (-ve no)

$$\begin{aligned}
 f: -24.75 &= -(11000.11)_2 \\
 &= -1.100011 \times 2^8 \\
 &= -10000000_2 \\
 127+8 &= 131 \\
 131_{10} &= 100000011_2
 \end{aligned}$$

$$\begin{aligned}
 \therefore L &= 100000011 \quad 1000110 \dots 0 \\
 (\text{Ans}) & \\
 &= -0.001001 \times 2^8 \\
 &= -1.001 \times 2^7
 \end{aligned}$$

$$\text{bias} = 127 - 8 = 127$$

f) To convert to decimal e.g.

$$L \quad 01111101 \quad 010 \dots 0.$$

$$\begin{aligned}
 \text{note: } \text{bias} &= 127_{10} \\
 \text{exp} &= 2^{-2} \\
 -1.010 \times 2^{-2} & \\
 = -0.0101 &= -(0.25 + 0.0625) = -0.3125
 \end{aligned}$$

MODULE 4:

TYPES OF MEMORY:

There are two types of memory used in digital system
 (i) RAM
 (ii) ROM

The process of storing new information in memory is referred to as memory write operation. The process of transferring the stored information out of memory is referred to as memory read operation.

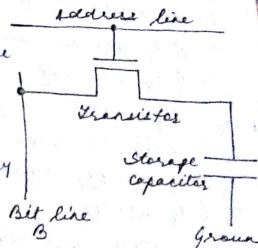
(i) RAM: RAM stores new information for later use. One distinguishable feature of RAM is that it can both read data from the memory & write data into the memory easily & rapidly. Read & write operations are performed through electric signals.

→ Another distinguishable feature of RAM is that it is volatile. If required constant power supply. If the power supply is interrupted the data is lost. Thus, RAM can be used only as temporary storage.

→ Two forms of RAM are: a) Dynamic RAM (DRAM)
 b) Static RAM (SRAM)

a) DRAM :

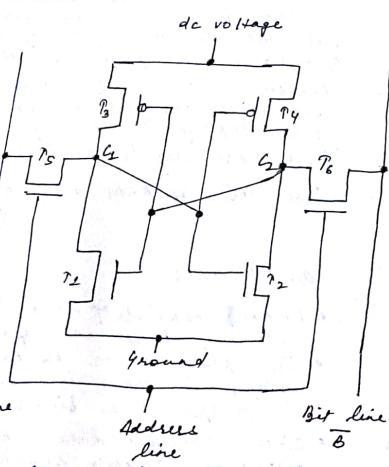
- DRAM consists of cells that store data as charge on capacitors.
- The presence or absence of voltage in charge in a capacitor is interpreted as binary 1 or 0.
- Since capacitors have a natural tendency to discharge, DRAM's require periodic charge refreshing to maintain data storage.
- Dynamic refers to this tendency of the stored charge to leak away, even with power continuously supplied.
- The address line gets activated when the bit line from this cell is to be read or written.
- The transistor acts as a switch that is closed if no voltage is applied to address line & open if no voltage is applied to address line.



- In write operation, a voltage signal is applied to bit line; a high voltage represents 1 & a low voltage represents 0. A signal is then transferred to the address line allowing a charge to be transferred to the capacitor.
- In read operation, when the address line is selected, the transistor turns on & the charge stored in capacitor is put out onto a bit line & to a sense amplifier. The sense amplifier compares the capacitor voltage to reference voltage & determines if it contains a binary 1 or 0. The readout from the cell discharges the capacitor, which must be restored to complete the operation.

a) SRAM

- Four transistors (T_1, T_2, T_3 & T_4) are used connected in a particular arrangement to produce stable logic state.



- In logic state 1, point C_1 is high, C_2 is low, T_1 & T_3 are on while T_2 & T_4 are off.

- In logic state 2, point C_1 is high, C_2 is low, T_1 & T_3 are off while Bit line B is on. T_2 & T_4 are on.

- Both states are unstable as long as dc voltage is applied.
- Unlike DRAM, no refresh is required to retain the data.

~~SRAM's~~ address line is used to open or close a switch.

The address line controls two transistors T_5 & T_6 .

When a signal is applied to the address line, the two transistors ~~turn on~~ are switched on, allowing a read or write operation.

In write operation, the desired bit value is applied to line B, while its complement is applied to line \bar{B} .

This forces the four transistors (T_1, T_2, T_3, T_4) into proper state. For a read operation, the bit value is read from line B.

DRAM

- Cells are volatile i.e. power must be continuously supplied to the memory to preserve the bit values.
- simpler & smaller than SRAM
- more dense (more cells per unit area).
- less expensive
- required supporting refresh circuitry. For larger memories, the cost of the refresh circuitry is compensated by the smaller size of DRAM cells.
- Thus, DRAM's are favoured for larger memory requirements.
- DRAM is used for main memory.

SRAM

- somewhat faster than DRAM.
- SRAM is used for cache memory (both on & off chip).

b) ROM

ROM can perform only the read operation. This means that binary information is stored inside memory & can be retrieved or read at any time. However, that information cannot be altered. ROM is a non-volatile device.

→ ROM is an example of programmable logic device (PLD). The binary information that is stored within is a device is specified in some fashion & then embedded within the hardware in a process called programming. The device.

↳ The word 'programming' here refers to the hardware procedure that specifies the no. of bits that are inserted into the hardware configuration of the device.

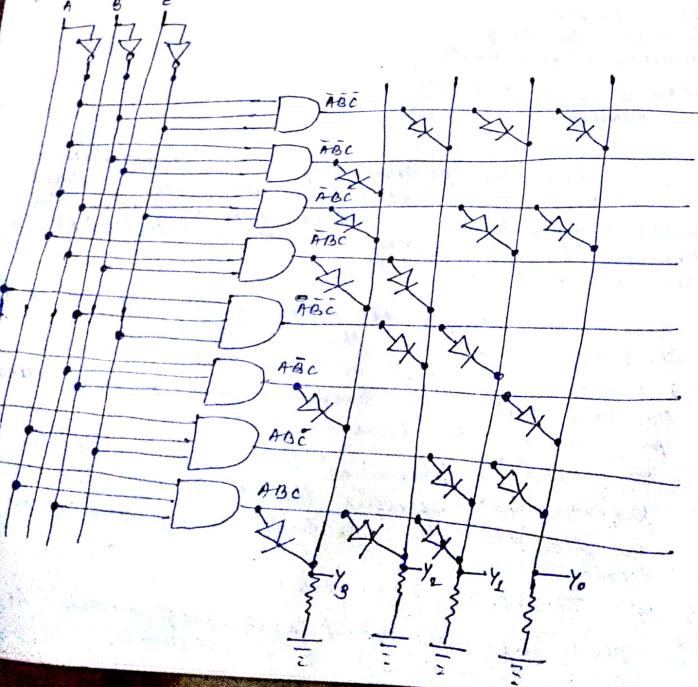
↳ Eg: TMS4164: 2192x8 bit

→ These chips are called mask-programmable. As programming is set by manufacturer.

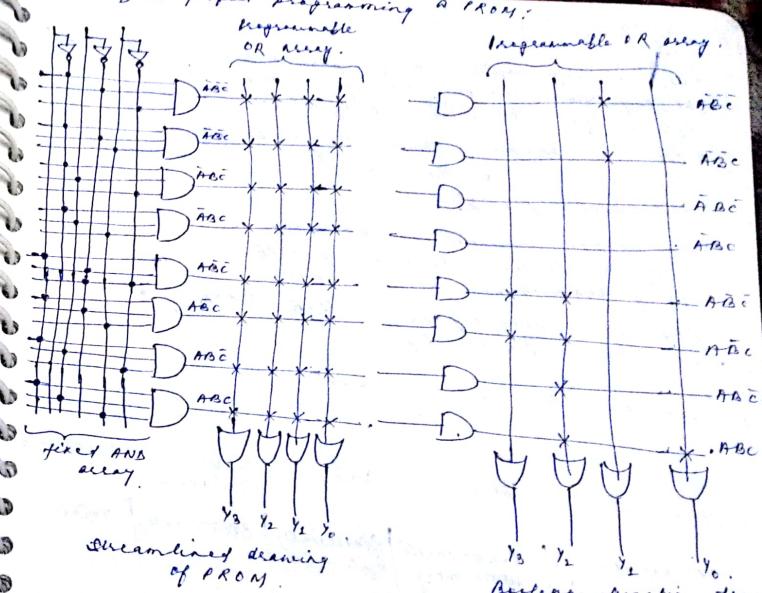
PROM (Programmable ROM)

- PROM's are said to be field programmable device as it can be programmed by the user.
- A device called PROM programmer is used to store the words by burning them in.
- Originally, there are diodes connected at the cross points.
- These diodes have a fuseble link each.
- PROM sends destructively high current through all diodes required to be removed.
- Only the desired diodes remain after programming a PROM.
- Programming like this is permanent because data cannot be erased once it has been burned in.
- Eg. 7451 8x8 256 bits organised as 32×8 .

PROM programmer using diodes:



Before & After programming a PROM:



EEPROM (Erasable PROM)

- EEPROM uses metal-oxide-semiconductor Field Effect Transistors (MOSFETs).
- Data is stored with EEPROM programmer.
- Later data can be erased with UV light.
 - ↳ The light passes through a quartz window in the package.
 - ↳ When it strikes the chip, the UV light releases all charges.
 - ↳ The effect is to wipe out all stored contents.
- So, EEPROM's are electrically reprogrammable by storing charges in MOSFETs & erasable by UV light.
- The erasure process can take 10 minutes to perform & can be performed repeatedly.
- Eg: 9716: 16384 bits organised as 2048×8 .

EEPROM (Electrically Erasable PROM)

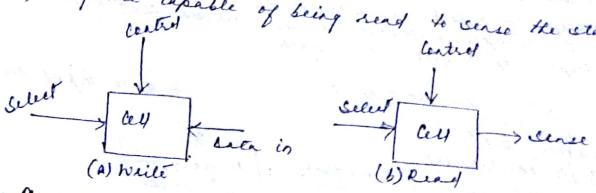
- It is a read-mostly memory capable of being written into at any time without erasing the prior contents; only the byte or bytes addressed are updated.
- The write operation takes considerably longer than read operation, on the order of several hundred microseconds per byte.
- The EEPROM combines the advantage of non-volatility with the flexibility of being updatable in place, using ordinary bus control, address and data lines.
- EEPROM is more expensive than EPROM of same density (fewer bits per chip).

Semiconductor Memory Types

Memory Type	Category	Erasure mechanism	Write mechanism	Volatility
ROM (read-only memory)	Read-write memory	Electrically by byte level	Electrically	Volatile
RAM (read only memory)	Read-only memory	Not possible	Mask	Non-volatile
EPROM	Read-mostly memory	Electrically by byte level	UV light, chip level	Non-volatile
EEPROM	Read-mostly memory	Electrically by byte level	Electrically by byte level	Non-volatile
Flash memory	Read-mostly memory	Electrically by block level	Electrically by block level	Non-volatile

MEMORY SYSTEM OVERVIEW:

- The basic element of semiconductor memory is the memory cell. Although a variety of electric technologies are used, all semiconductor memory cells share certain properties:
 - They exhibit two stable (or, semi-stable) states, which represent binary 1 & 0.
 - They are capable of being written into (written once), to be able to set the state.
 - They are capable of being read to sense the state.



→ Memory unit: Memory unit is the device to which binary information is transferred for storage & from which information is retrieved when needed for processing.

→ When data processing takes place, information from memory is transferred to select registers in the processing unit.

→ Intermediate / final results obtained in the processing unit are transferred back to be stored in memory.

→ Binary information received from an input device stored in memory & information transferred to output device is taken from memory.

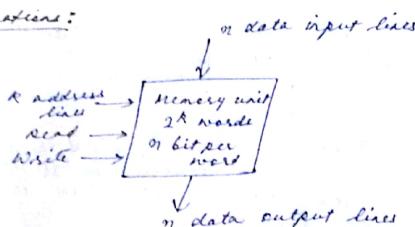
→ A memory unit is a collection of memory cells capable of storing a large quantity of binary information.

Memory storage data UNIT:

- ↳ A memory unit stores binary information in groups of bits called words.
- ↳ A word in memory is an entity of bits that move in a unit of storage as a unit.
- ↳ A memory word is a group of 16 or 32 bits that may represent a number, an instruction, one or more alphanumeric characters, or any other binary-coded information.
- ↳ A group of 2 bits is called a bit.
- ↳ Most computer memories use words that are multiples of 8 bits in length.

Memory unit specifications:

↳ Communication b/w
memory & its environment
is achieved through
data input/output
lines, & address
selection lines &
control lines that
specify the addr of
word.



- ↳ The address lines specify the particular word chosen among the many available.
- ↳ The read/write control input causes binary data to be transferred out/into the memory unit along the data lines.

↳ The address lines select one particular word.
↳ Each word in memory is assigned an identification number, called an address, starting from 0 upto $2^k - 1$, where k is the number of address lines.

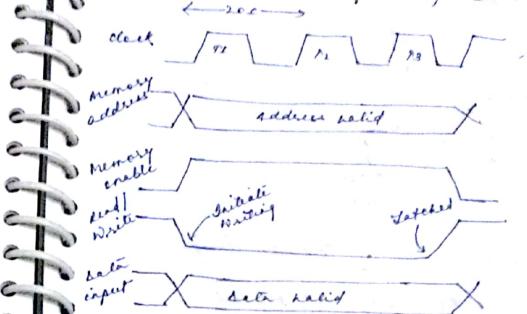
→ Memory word size θ may range from 16 words, requiring an address of 4 bits, to 2^{32} words, requiring 32 bits address bits.

→ Number of words (or bytes) in memory is referred to as kilo(K) = 2^{10} , mega(M) = 2^{20} , giga(G) = 2^{30} .

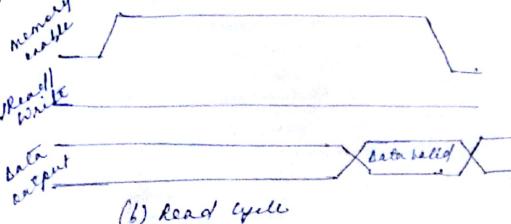
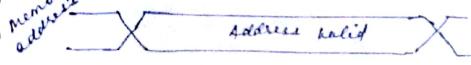
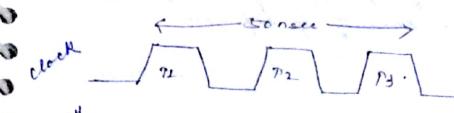
$$\rightarrow \text{So, } 14K = 2^{14}, 2M = 2^{24}, 4G = 2^{32}$$

→ Q: 16K * 16 = 16 bits in the address & 16 bits in each word
64K * 16 = 16 bits in the address & 10 " " " "

Memory cycle timing waveforms:



(A) Write cycle:

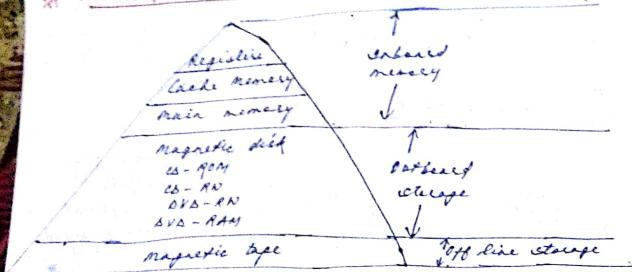


(B) Read cycle

The access time of memory is the time required to select a word and read it.

The cycle time of memory is the time required to complete a write operation.

MEMORY HIERARCHY



- being down hierarchy:
- Increasing cost per bit
 - Increasing capacity
 - Increasing access time
 - Decreasing frequency of access of the memory by the processor
 - Spatial locality of reference

Cache memory

- Used to speed up processing by making current program & data available to CPU at rapid rate.
- Used for storing segments of programs currently being executed in the case of temp data frequently needed in the present calculation.

→ intended to give memory speed approaching that of the fastest memories available & at the same time provide a large memory size at the price of less expensive types of semiconductor memories



Spatial locality of reference: Analysis of large no. of typical programs has shown that access the references to memory at any given time tend to be confined within a few localized areas in memory. This phenomena is known as the property of spatiality of ref.

Virtual Memory: Virtual memory allows programs to address memory from a logical point of view, without regard to the amount of main memory physically available. When virtual memory is used, the address fields of machine memory is used, the address fields of machine memory are called virtual addresses. To read to modifications certain memory locations, MMU translates them from main memory to physical add in main memory.

Logical cache / virtual cache: stores data using virtual addresses. Processor access the cache directly, without going through MMU.

Physical cache: stores data using main memory physical addresses.

Direct mapping

$i = j \text{ mod } (n)$ \rightarrow number of lines in cache
 cache line no. \rightarrow memory no.
 $s \rightarrow$ least significant bit
 (unique word/byte within a byte of main memory)
 3 bits \rightarrow specify one of 2^3 blocks of main memory
 cache interprets 3 bits as a tag of $(e-s)$ i.e.
 least significant portion
 size of tag $= s$ bits; identifies one of 2^s lines of cache memory.

$$\text{Address length} = s + w$$

$$\text{No. of addressable units} = 2^{s+w} \text{ words/bytes}$$

$$\text{Block size/line size} = 2^w$$

$$2^w \text{ bytes}$$

$$\text{size of cache} = 2^{s+w}$$

$$\text{size of tag} = (e-s)$$

Associative mapping: Size of tag: s

$$\text{Address length} = s + w$$

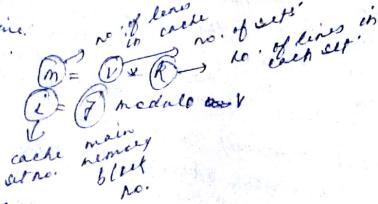
$$\text{No. of addressable units} = 2^{s+w} \text{ words/bytes}$$

$$\text{Block size} = 2^w$$

$$\text{No. of blocks in main memory} = 2^{e-w} = 2^s$$

$$\text{No. of lines in cache at a time: undetermined}$$

Set associative



$$\text{Address length} = s + w$$

$$\text{No. of addressable units} = 2^{s+w} \text{ words/bytes}$$

$$\text{Block size} = 2^w$$

$$\text{No. of blocks in main memory} = 2^e$$

$$\text{No. of lines in set} = p$$

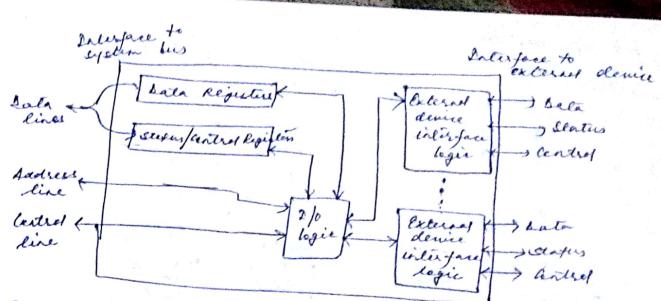
$$\text{No. of sets} = V = 2^d$$

$$\text{No. of lines in cache} = w \times p = e \times 2^d$$

$$\text{Size of cache} = e \times 2^{d+w}$$

$$\text{tag} = (e-d)$$

$i = j \text{ mod } (n)$ \rightarrow number of lines in cache



Q:

location = 3004

address field \rightarrow location = 301.

" " \rightarrow value = 400

Processor register, RL \rightarrow 200

location - contents
 300 - opcode
 301 - 400
 (address field of the above location).

Effective address:

(1) Direct: Address field contains the address of memory location

$$\text{Eff. add} \rightarrow 400$$

(2) Immediate

$$\textcircled{3} \text{ Relative} \\ 301 + 400 = 701.$$

(3) Register indirect

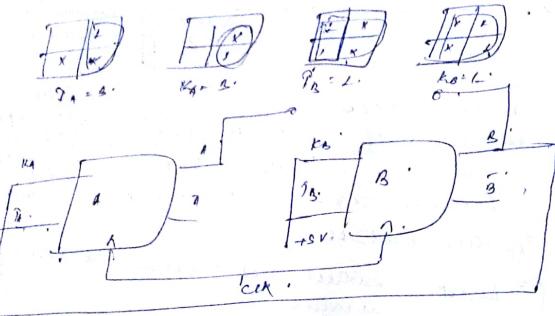
$$RL = 200$$

(4) indexed

$$400 + RL = \underline{\underline{600}}$$

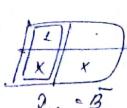
2 bit up counter

Next state		P.C.	
A	B	A + B	
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0



2 bit down counter

NS		P.C.	
A	B	A + B	
1	0	1	0
0	1	0	1
0	0	0	0
1	1	1	1
0	0	0	0



if for flip flop

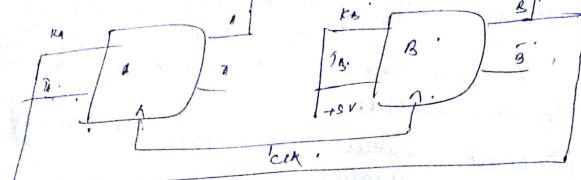
J ₁	J ₂	K ₁	K ₂
0	X	1	X
0	X	X	1
1	X	1	X
X	0	X	1
X	1	X	1

J₁ = A

K₁ = B

J₂ = L

K₂ = L



PLA

Y₁ Y₂ Y₃ Y₄

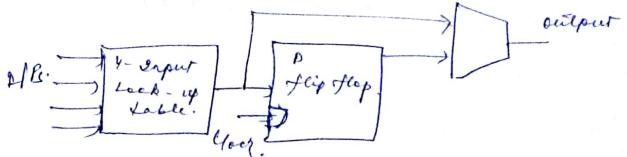
Input buffer & inverter

and plane

OR plane

f1 f2 f3 f4

FPGA

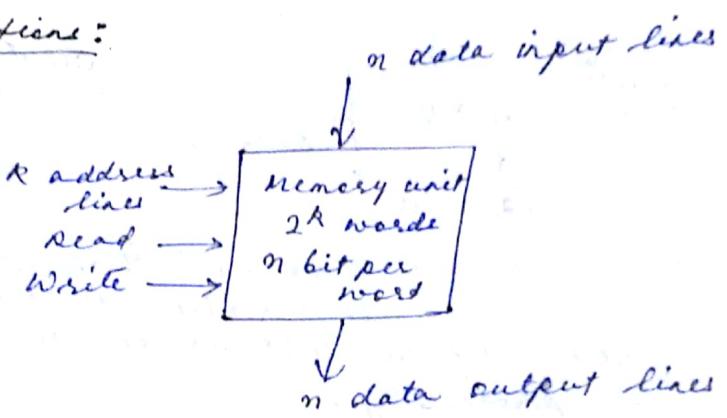


Memory storage data UNIT:

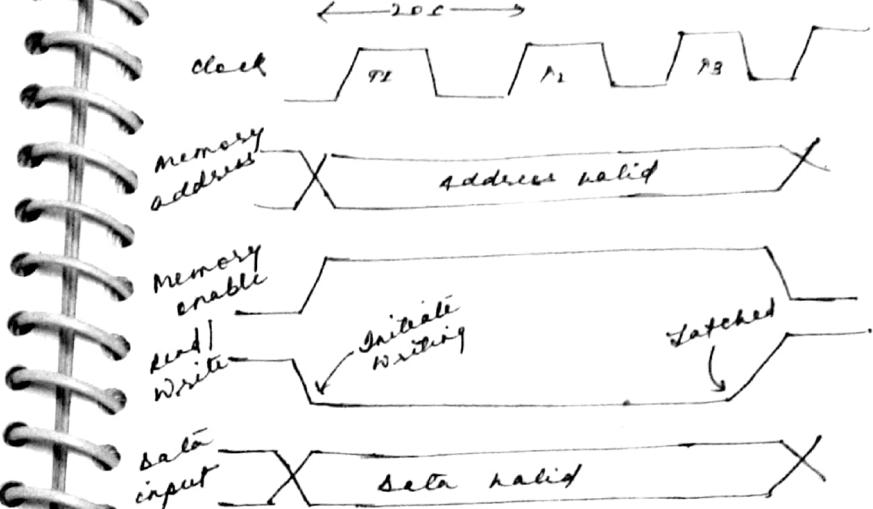
- ↳ A memory unit stores binary information in group of bits called words.
- ↳ A word in memory is an entity of bits that move in & out of storage as a unit.
- ↳ A memory word is a group of 1's & 0's that may represent a number, an instruction, one or more alphanumeric characters, or any other binary-coded information.
- ↳ A group of 8 bits is called a byte.
- ↳ Most computer memories use words that are multiples of 8 bits in length.

Memory unit specifications:

- ↳ Communication bet' memory & its environment is achieved through data input/output lines, & address selection lines & control lines that specify the dir' of transfer.
- ↳ The 1 address lines specify the particular word chosen among the many available.
- ↳ The Read/Write control input causes binary data to be transferred out/into the memory unit using the n data lines.
- ↳ The address lines select one particular word.
- ↳ Each word in memory is assigned an identification number, called an address, starting from 0 upto $2^k - 1$, where k is the number of address lines.
 - Memories vary greatly in size & may range from 1024 words, requiring an address of 10 bits, to 2^{32} words, requiring 32 ~~bits~~ address bits.
 - Number of words (or bytes) in memory is referred to as kilo(K) = 2^{10} , mega(M) = 2^{20} , giga(G) = 2^{30} .
 - So, $64K = 2^{16}$, $2M = 2^{21}$, $4G = 2^{32}$
 - Eg: $64K \times 16 = 10$ bits in the address & 16 bits in each word
 $64K \times 10 = 16$ bits in the address & $10 \times 16 = 160$ bits in each word



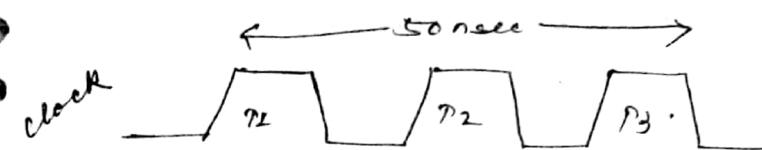
Memory cycle timing waveforms:



(a) Write cycle.

The access time of memory is the time required to select a word and read it.

The cycle time of memory is the time required to complete a write operation.



(b) Read cycle

Memory cycle timing waveforms:

← 200 →

clock

memory address

address valid

memory enable

send write

initiate writing

latched

data input

data valid

(a) Write cycle

The access time of memory is the time required to select a word and read it.

The cycle time of memory is the time required to complete a write operation.

← 50 nsec →

clock

memory address

address valid

memory enable

read/write

data output

data valid

(b) Read cycle