Biyani's Think Tank

Concept based notes

Computer Architecture

(MCA)

Shweta Nigam
Namarta Mehta
Dept. of MCA(Information Technology)
Biyani Institute of Science and Management
Jaipur

Published by :
Think Tanks
Biyani Group of Colleges

Sector-3, Vidhyadhar Nagar,
Jaipur-302 023 (Rajasthan)
Ph : 0141-2338371, 2338591-95 • Fax : 0141-2338007
E-mail : acad@biyanicolleges.org
Website :www.gurukpo.com; www.biyanicolleges.org

Edition : 2011
Price :

Leaser Type Settled by :
Biyani College Printing Department

Preface

I am glad to present this book, especially designed to serve the needs of the students. The book has been written keeping in mind the general weakness in understanding the fundamental concept of the topic. The book is self-explanatory and adopts the "Teach Yourself" style. It is based on question-answer pattern. The language of book is quite easy and understandable based on scientific approach.

The text explained all the concepts of Internet & Intranet are very simple way and according the syllabus of (BCA) graduate level students.

Any further improvement in the contents of the book by making corrections, omission and inclusion is keen to be achieved based on suggestions from the reader for which the author shall be obliged.

I acknowledge special thanks to Mr. Rajeev Biyani, *Chairman* & Dr. Sanjay Biyani, *Director (Acad.)* Biyani Group of Colleges, who is the backbone and main concept provider and also have been constant source of motivation throughout this endeavor. We also extend our thanks to Biyani Sikhshan Samiti, Jaipur, who played an active role in co-coordinating the various stages of this endeavor and spearheaded the publishing work.

I look forward to receiving valuable suggestions from professors of various educational institutions, other faculty members and the students for improvement of the quality of the book. The reader may feel free to send in their comments and suggestions to the under mentioned address.

Author

Syllabus

Logic gates, basic combinational logic, multiplexer, decoders, encoders comparators, adder and subtracters, BCD to 7 segment decoder, sequential circuits, RS, JK, D and T flip flops, counter and shift register, programmable logic array (PLA) , programmable logic device (PLD) .

Addressing methods and machine program sequencing-memory locations addresses, encoding of information, instructions and instructions sequencing, addressing modes, paging, relative, indirect and indexed addressing.

Basics of Computer organization: System buses and instruction cycles, memory subsystem organization and interfacing, I/O subsystem organizations and interfacing, Register transfer languages.

CPU design : Specifying a CPU, design and implementation of a simple CPU (fetching instructions from memory, decoding and executing instructions, establishing required data paths, design of ALU, design of the control unit and design verification), design and implementation of a simple micro-sequencer.

Memory systems, virtual and cache memory.

Input and Output organization: Asynchronous data transfer, programmed I/O Interrupts (types, processing of interrupts implementing interrupts inside CPU) Direct memory access, I/O processors, serial communication.

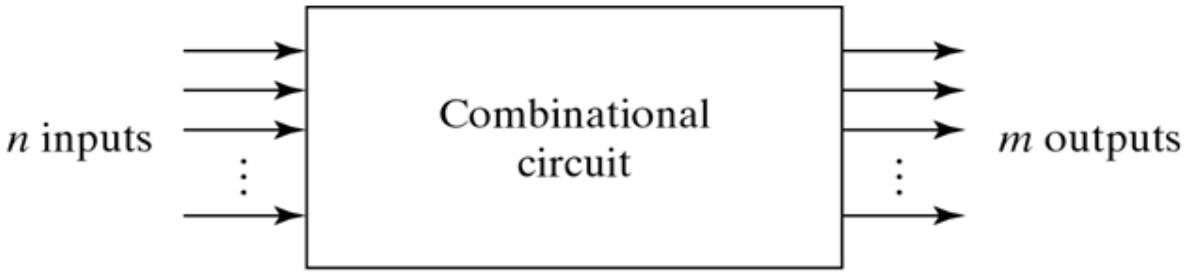
Contents

Chapter 1

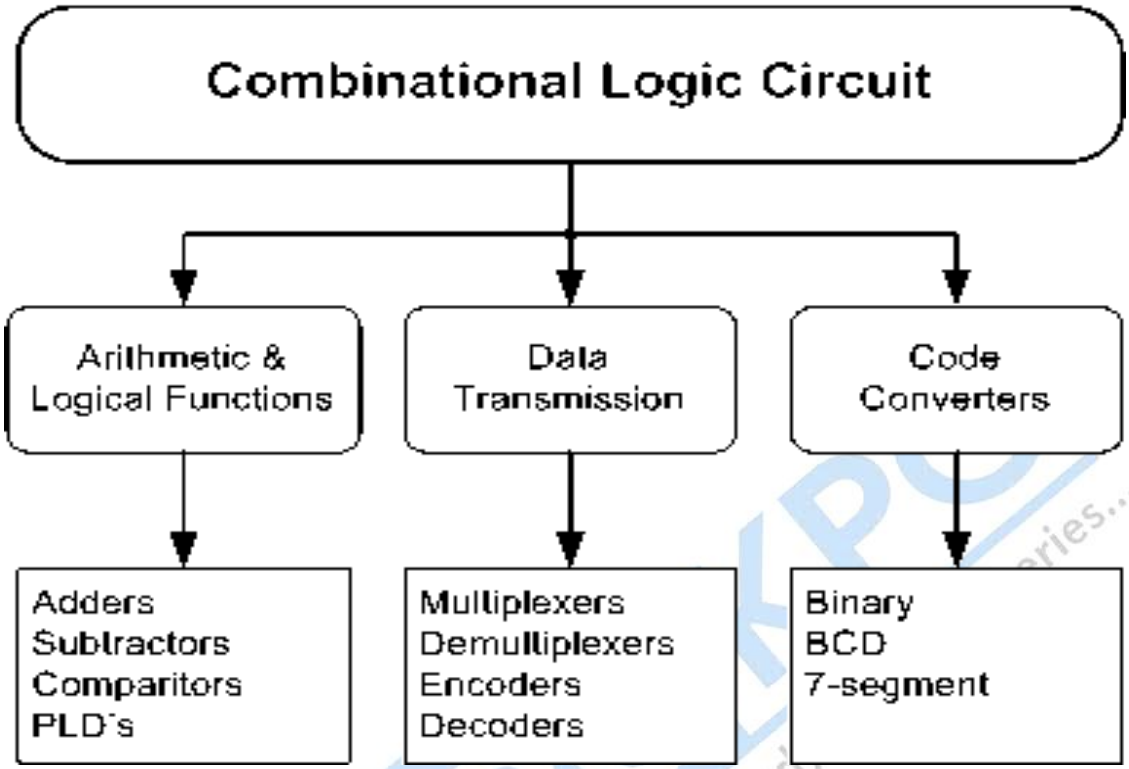
Combinational Circuits

Q1. Define Combinational Circuits.

Ans. **Combinational Logic Circuits** are made up from basic logic **NAND**, **NOR** or **NOT** gates that are "combined" or connected together to produce more complicated switching circuits. These logic gates are the building blocks of combinational logic circuits. An example of a combinational circuit is a decoder, which converts the binary code data present at its input into a number of different output lines, one at a time producing an equivalent decimal code at its output.

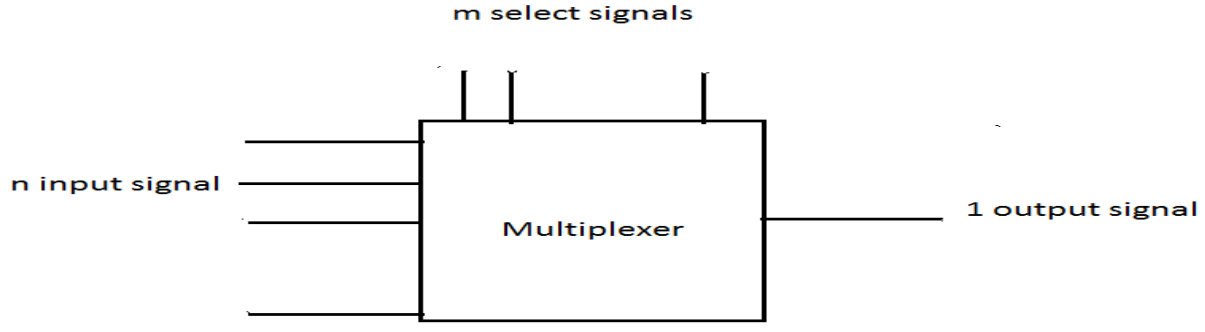


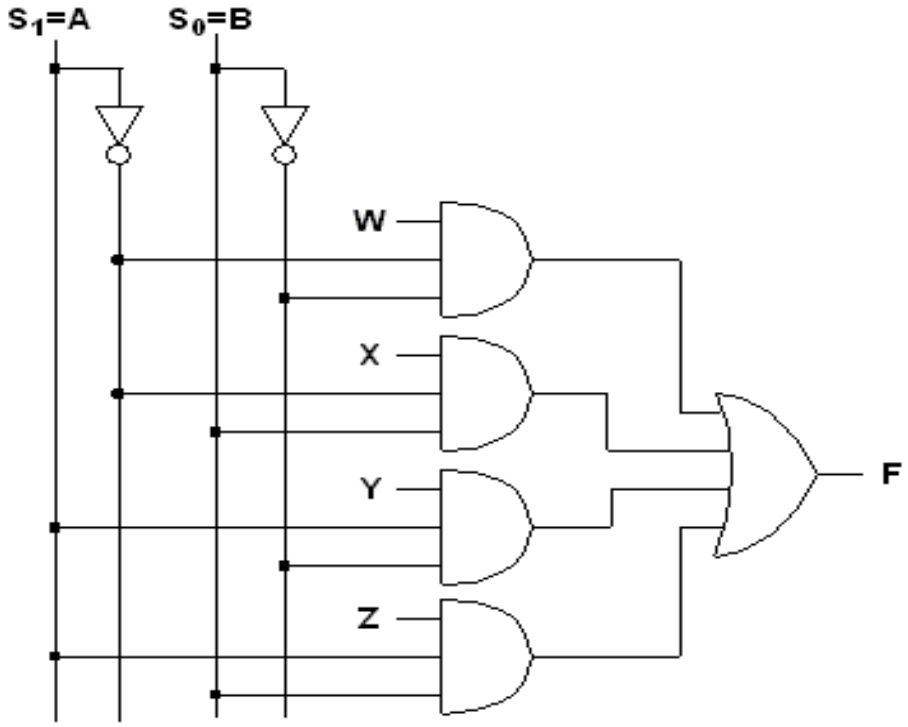
Block Diagram of Combinational Circuit



Q2. Define Multiplexer .Draw 4X1 MUX.

Ans. A **multiplexer** or **mux** is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2^n inputs has n select lines, which are used to select which input line to send to the output. Multiplexers are mainly used to increase the amount of data that can be sent over the network within a certain amount of time and bandwidth.





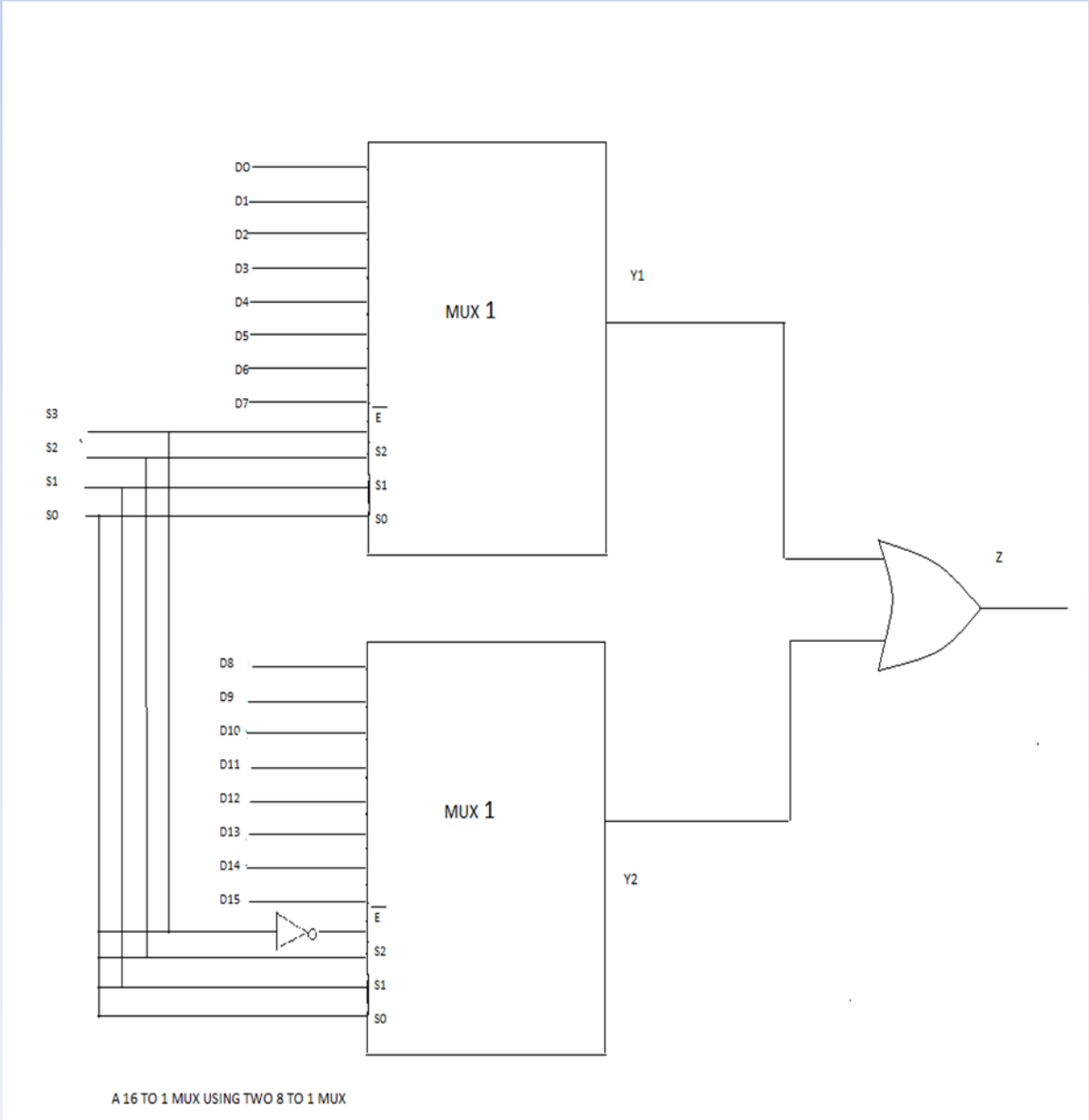
$F = \bar{A}\bar{B}W + \bar{A}BX + A\bar{B}Y + ABZ$

Truth table of 4 to 1 MUX

Data select inputs		Outputs
S1	**S0**	**F**
0	0	W
0	1	X
1	0	Y
1	1	Z

Q3. Design a 16- to -1 MUX using two 8 – to -1 MUX

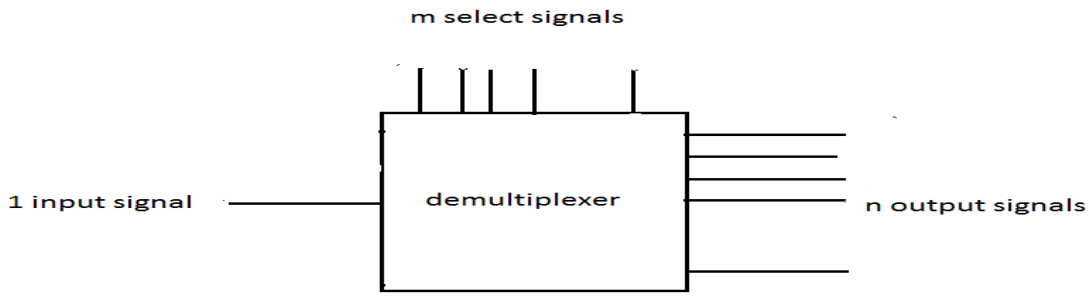
Ans

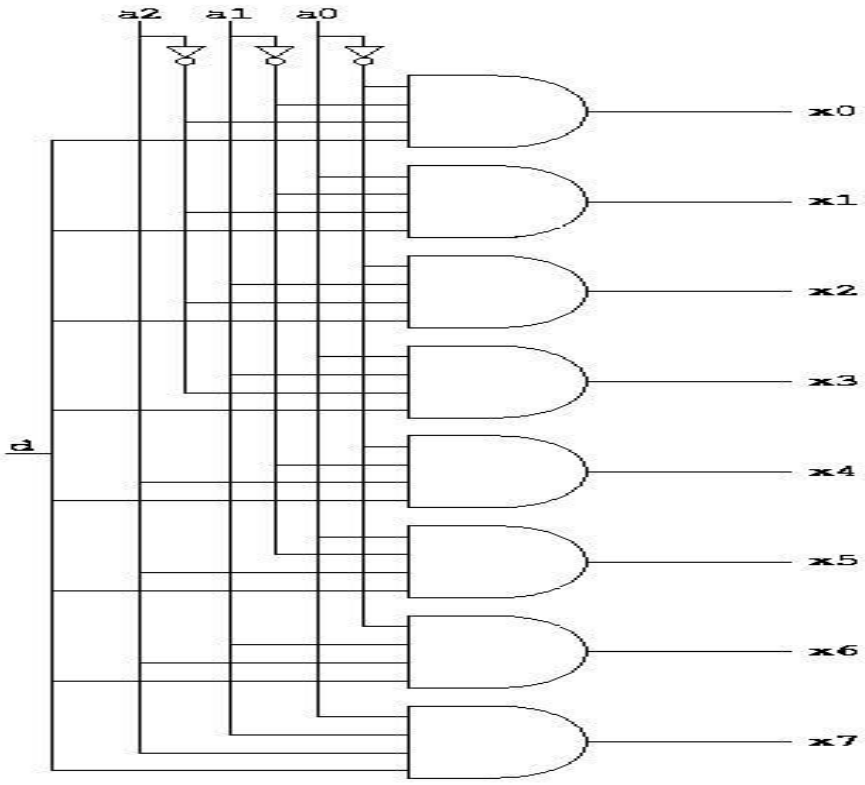


A 16 TO 1 MUX USING TWO 8 TO 1 MUX

Q4. Define Demultiplexer .Along with 1X8 Demux.

Ans. The demultiplexer is the inverse of the multiplexer, in that it takes a single data input and n address inputs. It has 2n outputs. The address input determine which data output is going to have the same value as the data input. The other data outputs will have the value 0.





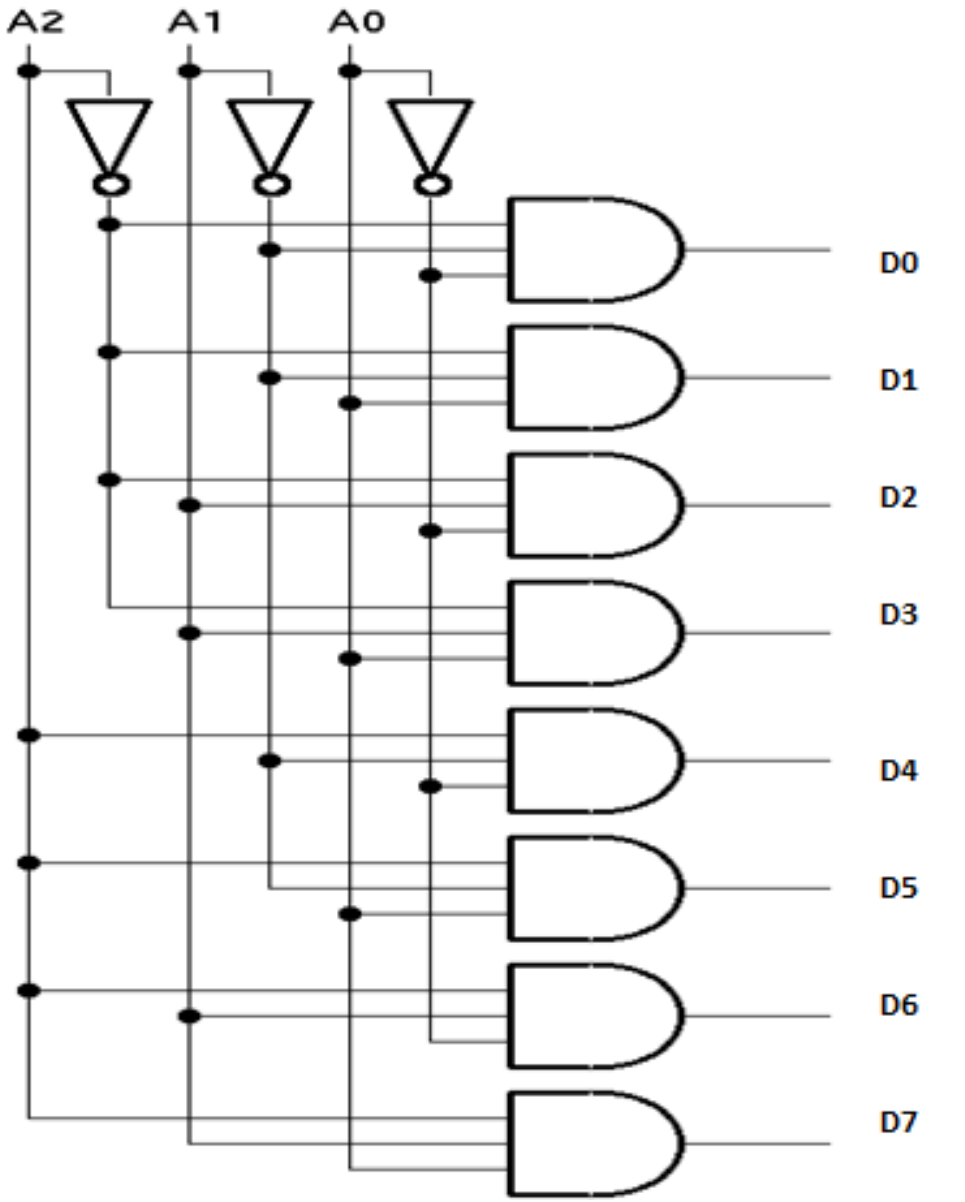
1 TO 8 DMUX

Truth table of 1 to 8 DEMUX

a2	a1	a0	d	x7	x6	x5	x4	x3	x2	x1	x0
0	0	0	c	0	0	0	0	0	0	0	c
0	0	1	c	0	0	0	0	0	0	c	0
0	1	0	c	0	0	0	0	0	c	0	0
0	1	1	c	0	0	0	0	c	0	0	0
1	0	0	c	0	0	0	c	0	0	0	0
1	0	1	c	0	0	c	0	0	0	0	0
1	1	0	c	0	c	0	0	0	0	0	0
1	1	1	c	c	0	0	0	0	0	0	0

Q5. Explain 3X8 Decoder. And give its application.

Ans. Is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines For example if the number of input is n=3 the number of output lines can be $m=2^3$.


A2
A1
A0
D0
D1
D2
D3
D4
D5
D6
D7

Truth table for binary to octal

Inputs			Outputs							
A_2	A_1	A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
×	×	×	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Applications

1. They are used in counter systems.
2. They are used in analog to digital converters
3. Decoder output can be used to drive a display system

Q6. What are the applications of Multiplexer ?

Ans. These circuits use mostly find in numerous and varied applications in digital systems of all types such as **data selection, data routing, operation sequencing, parallel-to-serial conversion.**

Application areas

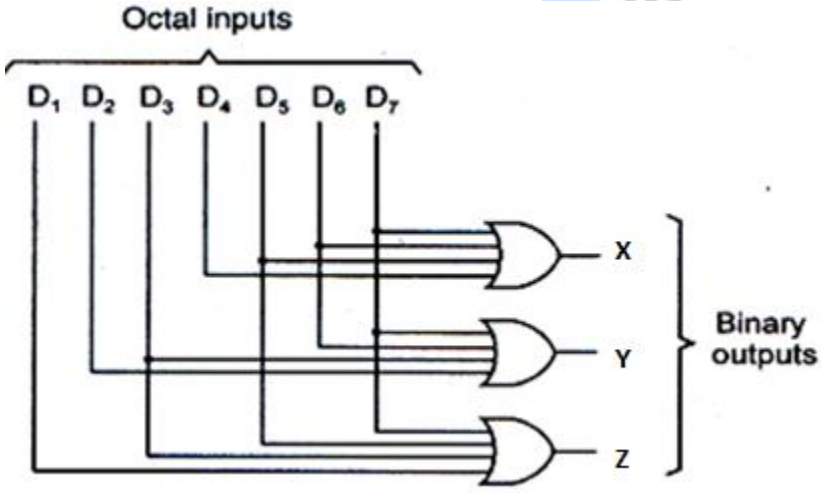
1. Telephony
2. Video processing
3. Digital broadcasting
4. Analog broadcasting

Q7. Explain Encoder.

Ans. An encoder is a digital function that produces a reverse operation from that of a decoder. An encoder has 2^n(or less) input lines and n output lines. The output lines generate the binary code for the 2^n input variables.

Octal –to-Binary Encoder

inputs								outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



Q8 Define Programmable Logic Devices (PLDs)

Ans. An IC that contains large numbers of gates, flip-flops, etc. that can be configured by the user to perform different functions is called a Programmable Logic Device (PLD).
The internal logic gates and/or connections of PLDs can be changed/configured by a programming process

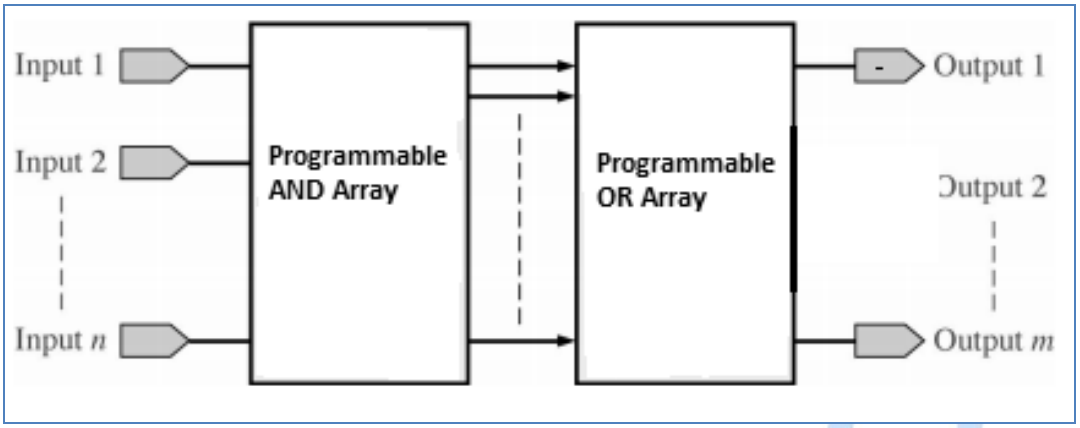
Q9. What are the types of PLDs?.Explain.

Ans. The three fundamental types of PLDs differ in the placement of programmable connections in the AND-OR arrays.

- The PROM (Programmable Read Only Memory) has a fixed AND array (constructed as a decoder) and programmable connections for the output OR gates array. The PROM implements Boolean functions in sum-of-minterms form.
- The PAL (Programmable Array Logic) device has a programmable AND array and fixed connections for the OR array.
- The PLA (Programmable Logic Array) has programmable connections for both AND and OR arrays. So it is the most flexible type of PLD.

Q10. Explain Programmable Logic Array (PLA).

Ans. A **programmable logic array** (PLA) is a kind of programmable logic device used to implement combinational logic circuits. The PLA has a set of programmable AND gate planes, which link to a set of programmable OR gate planes, which can then be conditionally complemented to produce an output. This layout allows for a large number of logic functions to be synthesized in the sum of products (and sometimes product of sums) canonical forms.



Block diagram of a PLA (programmable logic array)

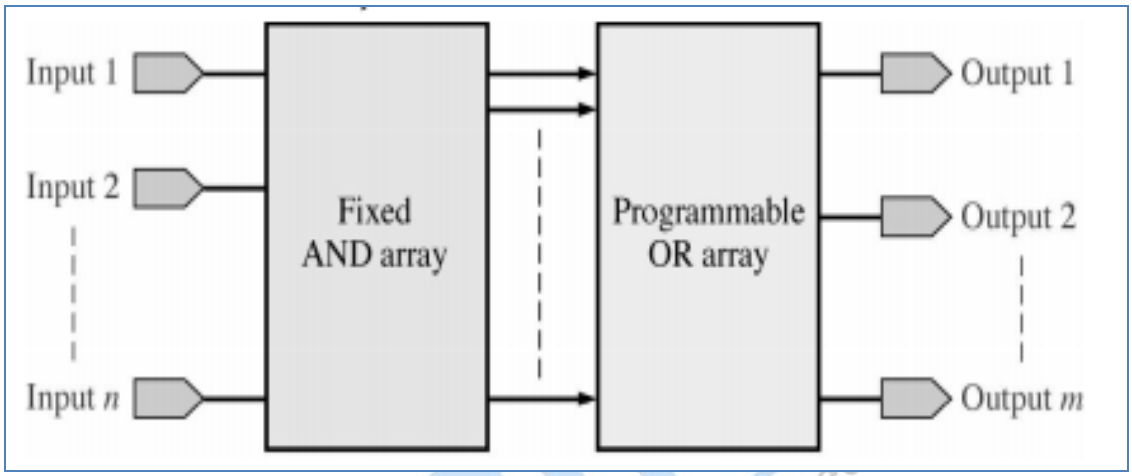
APPLICATION

One application of a PLA is to implement the control over a datapath. It defines various states in an instruction set, and produces the next state (by conditional branching). [eg. if the machine is in state 2, and will go to state 4 if the instruction contains an immediate field; then the PLA should define the actions of the control in state 2, will set the next state to be 4 if the instruction contains an immediate field, and will define the actions of the control in state 4]. Programmable Logic Arrays should correspond to a state diagram for the system. Other commonly used programmable logic devices are PAL, CPLD and FPGA.

Q11 Explain Programmable Read-Only Memory(PROM)

Ans. The first PLD is PROM was introduced in 1970. PROMs was introduced for use as computer memories in which to store program instructions and constant data values. PROM have fixed AND plane and programmable OR plane. PROM can be use to program any combinational logics with limited numbers of inputs and outputs. Given n variables, it would necessary to have 2n AND gates, one for each possible minterm. A figure below shows the unprogrammed PROM for 3 inputs and 3 outputs, where AND plane is fixed and OR plane is programmable. The programmable links in OR array can be implemented as fused link, or as EPROM transistor or E2PROM cells depend on vendors. PROMs are useful for equations requiring a large number of product terms, but they can

support few inputs as every input combination is always decoded and used.
The PROM is used primarily as an addressable memory and not as a logic device because of limitations imposed by fixed AND gates.

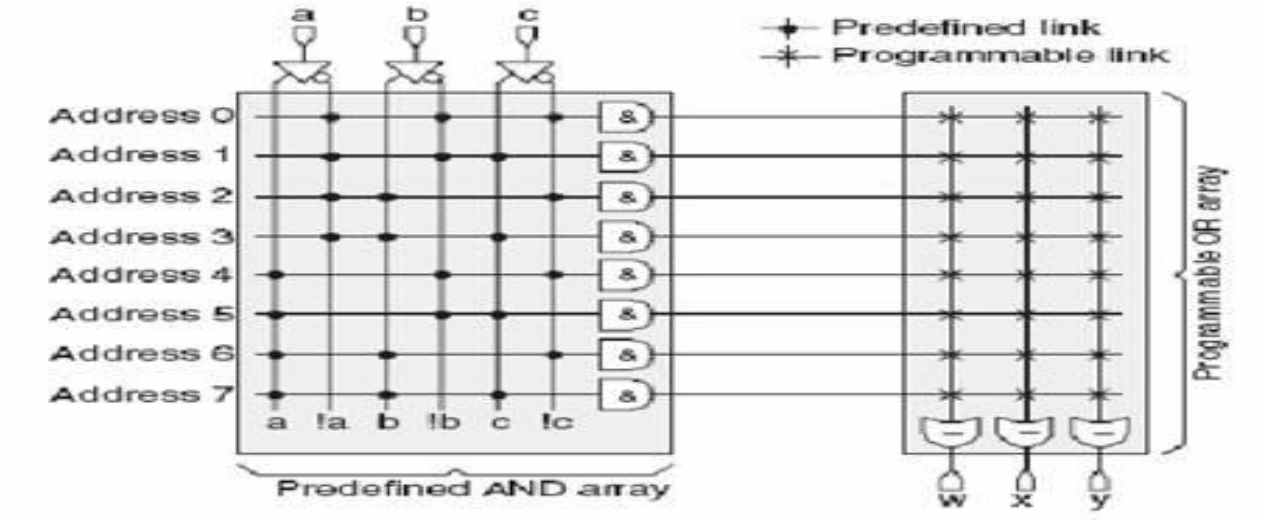


Block diagram of a PROM (programmable read-only memory)

Q12. Explain the Two basic versions of PROM.

Ans. **1) Mask-Programmable:** can be programmed only by the manufacturer. Mask-programmable chip has less delay because connections within the device can be hardwired during manufacture.

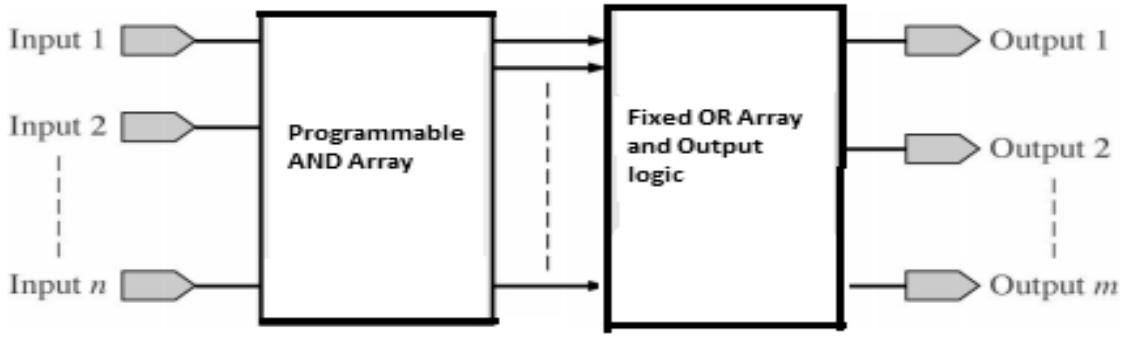
2) Field-Programmable: can be programmed by the end-user .Field-programmable chips are less expensive, and can be programmed immediately. The Field Programmable PROM developed into two types, the Erasable Programmable Read-Only Memory (EPROM) and the Electrically Erasable Programmable Read-Only Memory (E2PROM). The E2PROM has the advantage of being erasable and reprogrammable many times.



Q.13 Explain Programmable Array Logic (PAL).

Ans. A PLD in which the OR array is fixed (pre-defined) but the AND array is programmable. PAL chips use fuse-programmable logic (i.e., overvoltage is applied to portions of the chip to physically blow a circuit open).

It was developed to overcome certain disadvantages of PLA, such as longer delays due to the additional fusible links that result from using two programmable arrays and more difficult complexity.



Block diagram of a PAL (programmable array logic)

Chapter 2

Sequential Circuits

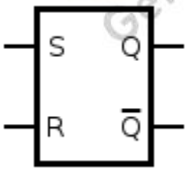
Q1. What is the difference between latch and flip flop.

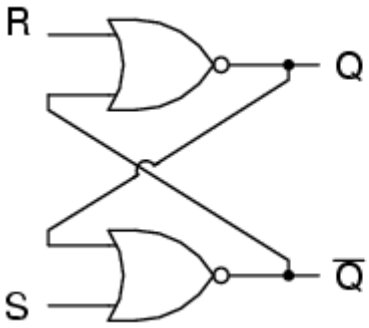
Ans. *latches* and *flip-flops* are the building blocks of sequential circuits.

While gates had to be built directly from transistors, latches can be built from gates, and flip-flops can be built from latches. This fact will make it somewhat easier to understand latches and flip-flops. Both latches and flip-flops are circuit elements whose output depends not only on the current inputs, but also on previous inputs and outputs. The difference between a latch and a flip-flop is that a latch does not have a *clock signal*, whereas a flip-flop always does. Latches are *asynchronous*, which means that the output changes very soon after the input changes. Most computers today, on the other hand, are *synchronous*, which means that the outputs of all the sequential circuits change simultaneously to the rhythm of a global *clock signal*. A *flip-flop* is a synchronous version of the latch.

Q2. Explain SR LATCH Working .

Ans. This latch is called *SR-latch*, which stands for *set* and *reset*.





S	R	Q	$\overline{Q}$
0	0	latch	latch
0	1	0	1
1	0	1	0
1	1	0	0

This latch is called *SR-latch*, which stands for *set* and *reset*.

When S = 0 and R = 0: If we assume Q = 1 and Q' = 0 as initial condition, then output Q after input is applied would be Q = (R + Q')' = 1 and Q' = (S + Q)' = 0. Assuming Q = 0 and Q' = 1 as initial condition, then output Q after the input applied would be Q = (R + Q')' = 0 and Q' = (S + Q)' = 1. So it is clear that when both S and R inputs are LOW, the output is retained as before the application of inputs. (i.e. there is no state change).

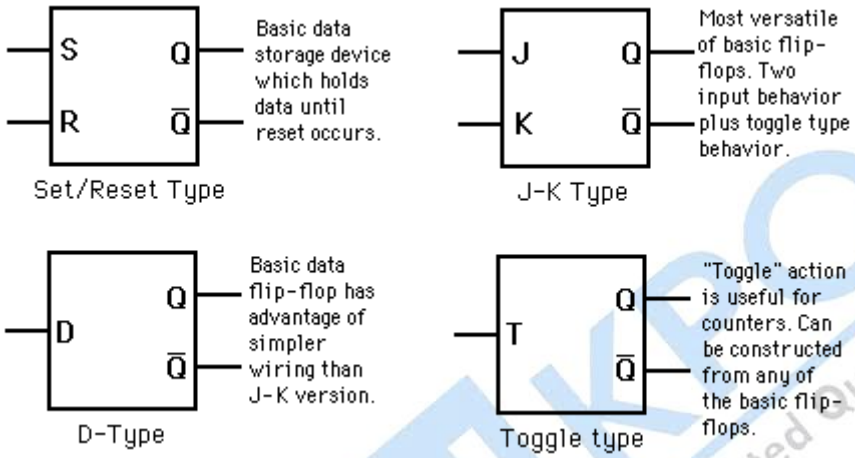
When S = 1 and R = 0: If we assume Q = 1 and Q' = 0 as initial condition, then output Q after input is applied would be Q = (R + Q')' = 1 and Q' = (S + Q)' = 0. Assuming Q = 0 and Q' = 1 as initial condition, then output Q after the input applied would be Q = (R + Q')' = 1 and Q' = (S + Q)' = 0. So in simple words when S is HIGH and R is LOW, output Q is HIGH.

When S = 0 and R = 1: If we assume Q = 1 and Q' = 0 as initial condition, then output Q after input is applied would be Q = (R + Q')' = 0 and Q' = (S + Q)' = 1. Assuming Q = 0 and Q' = 1 as initial condition, then output Q after the input applied would be Q = (R + Q')' = 0 and Q' = (S + Q)' = 1. So in simple words when S is LOW and R is HIGH, output Q is LOW.

When S = 1 and R =1 : No matter what state Q and Q' are in, application of 1 at input of NOR gate always results in 0 at output of NOR gate, which results in both Q and Q' set to LOW (i.e. Q = Q'). LOW in both the outputs basically is wrong, so this case is invalid.

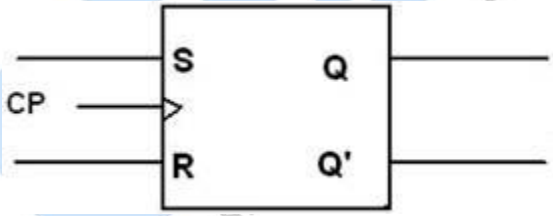
Q3. Explain Flip-Flops. Along with its types

Ans. "Flip-flop" is the common name given to two-state devices which offer basic memory for sequential logic operations. Flip-flops are heavily used for digital data storage and transfer and are commonly used in banks called "registers" for the storage of binary numerical data.



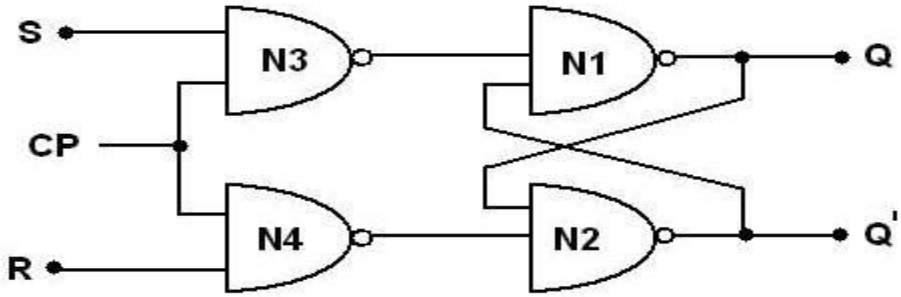
Q4. Explain SR flip flop working.

Ans.



BLOCK DIAGRAM

The basic flip-flop is a one bit memory cell that gives the fundamental idea of memory device. It constructed using two NAND gates. The two NAND gates N1 and N2 are connected such that, output of N1 is connected to input of N 2 and output of N2 to input of N1 . These form the feedback path the inputs are S and R, and outputs are Q and Q'.



LOGIC DIAGRAM

Operation:

1. When CP=0 the output of N 3 and N4 are 1 regardless of the value of S and R. This is given as input to N1 and N2. This makes the previous value of Q and Q′ unchanged.

2. When CP=1 the information at S and R inputs are allowed to reach the latch and change of state in flip-flop takes place.

3. CP=1, S=1, R=0 gives the SET state i.e., Q=1, Q′=0.

4. CP=1, S=0, R=1 gives the RESET state i.e., Q=0, Q′=1.

5. CP=1, S=0, R=0 does not affect the state of flip-flop.

6. CP=1, S=1, R=1 is not allowed, because it is not able to determine the next state. This condition is said to be a "race condition".

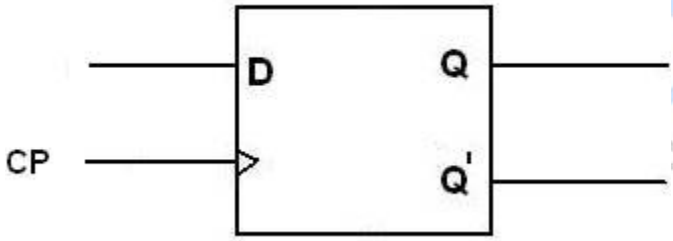
In the logic symbol CP input is marked with a triangle. It indicates the circuit responds to an input change from 0 to 1. The characteristic table gives the operation conditions of flip-flop. Q (t) is the present state maintained in the flip-flop at time 't'. Q (t+1) is the state after the occurrence of clock pulse.

TRUTH TABLE

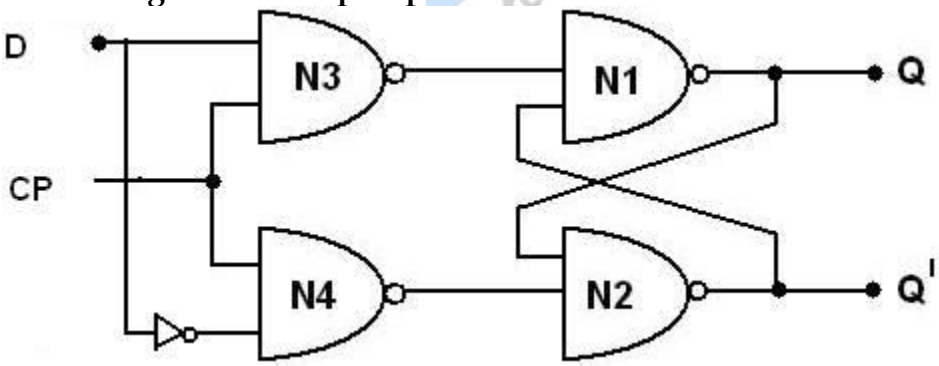
S	R	Q (t+1)	Comments
0	0	Q t	No change
0	1	0	Reset / clear
1	0	1	Set
1	1	*	Not allowed

Q5. Explain D flip flop.

Ans. The D flip-flop is the modified form of R-S flip-flop. R-S flip-flop is converted to D flip-flop by adding an inverter between S and R and only one input D is taken instead of S and R. So one input is D and complement of D is given as another input.



Block diagram of D flip flop



LOGIC DIAGRAM

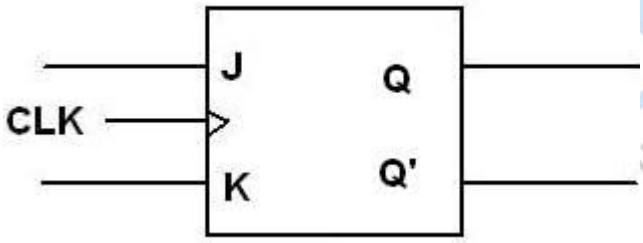
When the clock is low both the NAND gates (N1 and N2) are disabled and Q retains its last value. When clock is high both the gates are enabled and the input value at D is transferred to its output Q. D flip-flop is also called "Data flip-flop".

Truth table

CP	D	Q
0	x	Previous state
1	0	0
1	1	1

Q6. Explain J-K Flip-Flop

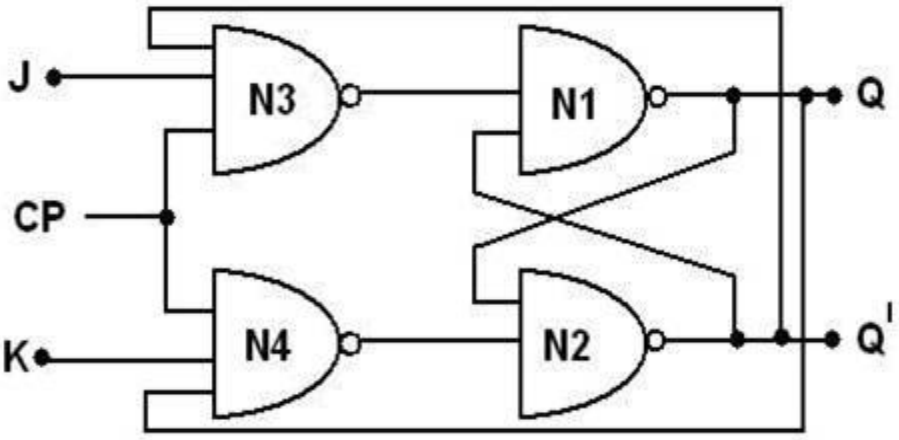
Ans. The J-K flip-flop is the most versatile of the basic flip-flops. It has the input- following character of the clocked D flip-flop but has two inputs, traditionally labeled J and K. If J and K are different then the output Q takes the value of J at the next clock edge.



Block diagram

If J and K are both low then no change occurs. If J and K are both high at the clock edge then the output will toggle from one state to the other. It can perform the functions of the set/reset flip-flop and has the advantage that there are no ambiguous states. It can also act as a T flip-flop to accomplish toggling action if J and K are tied together. This toggle application finds extensive use in binary counters.

The race condition in RS flip-flop, when R=S=1 is eliminated in J-K flip-flop. There is a feedback from the output to the inputs.



the outputs feed back to the enabling NAND gates. This is what gives the toggling action when J=K=1.

Truth table

J	K	Q (t+1)	Comments
0	0	Q t	No change
0	1	0	Reset / clear
1	0	1	Set
1	1	Q' t	Complement/ toggle.

The J and K are called control inputs, because they determine what the flip-flop does when a positive clock edge arrives.

Operation:

1. When J=0, K=0 then both N3 and N4 will produce high output and the previous value of Q and Q' retained as it is.

2. When J=0, K=1, N3 will get an output as 1 and output of N4 depends on the value of Q. The final output is Q=0, Q'=1 i.e., reset state

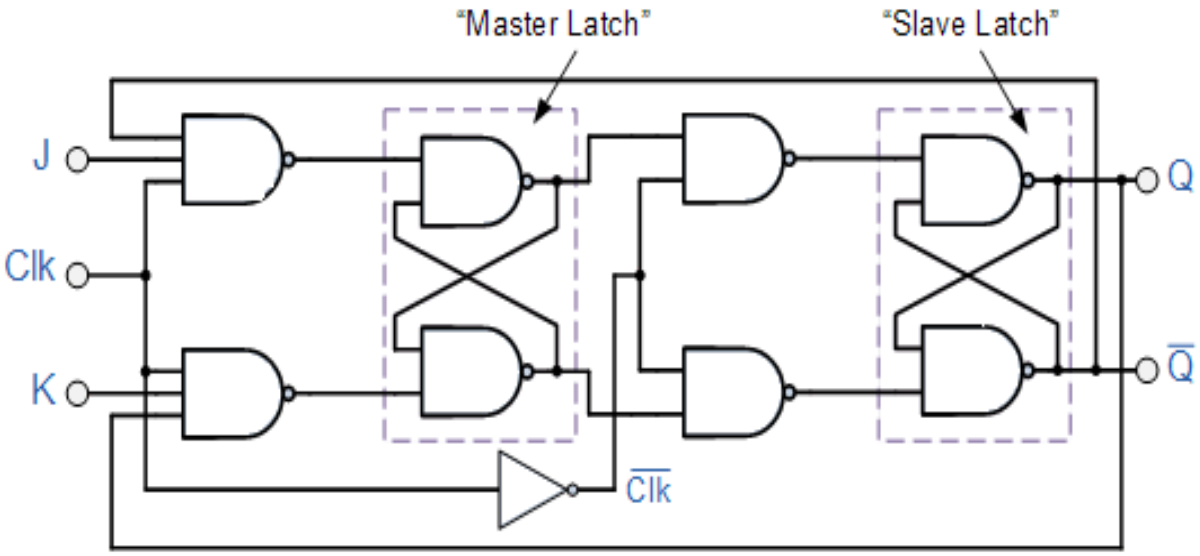
3. When J=1, K=0 the output of N 4 is 1 and N 3 depends on the value of Q'. The final output is Q=1 and Q'=0 i.e., set state

4. When J=1, K=1 it is possible to set (or) reset the flip-flop depending on the current state of output. If Q=1, Q'=0 then N4 passes '0'to N2 which produces Q'=1, Q=0 which is reset state. When J=1, K=1, Q changes to the complement of the last state. The flip-flop is said to be in the toggle state.

Q7. Explain The Master-Slave JK Flip-flop.

Ans. The **Master-Slave Flip-Flop** is basically two gated SR flip-flops connected together in a series configuration with the slave having an inverted clock pulse. The outputs from Q and Q from the "Slave" flip-flop are fed back to the inputs of the "Master" with the outputs of the "Master" flip-flop being connected to the two inputs of the "Slave" flip-flop. This feedback configuration from the slave's output to the master's input gives the characteristic toggle of the JK flip-flop as shown below.

The Master-Slave JK Flip-Flop



The input signals J and K are connected to the gated "master" SR flip-flop which "locks" the input condition while the clock (Clk) input is "HIGH" at logic level "1". As the clock input of the "slave" flip-flop is the inverse (complement) of the "master" clock input, the "slave" SR flip-flop does not toggle. The outputs from the "master" flip-flop are only "seen" by the gated "slave" flip-flop when the clock input goes "LOW" to logic level "0". When the clock is "LOW", the outputs from the "master" flip-flop are latched and any additional changes to its inputs are ignored. The gated "slave" flip-flop now responds to the state of its inputs passed over by the "master" section. Then on the "Low-to-High" transition of the clock pulse the inputs of the "master" flip-flop are fed through to the gated inputs of the "slave" flip-flop and on the "High-to-Low" transition the same inputs are reflected on the output of the "slave" making this type of flip-flop edge or pulse-triggered.

Then, the circuit accepts input data when the clock signal is "HIGH", and passes the data to the output on the falling-edge of the clock signal. In other words, the **Master-Slave JK Flip-flop** is a "Synchronous" device as it only passes data with the timing of the clock signal.

Chapter 3

Register

Q1. Explain registers.

Ans. It is group of flip flop suitable for storing binary information. Each flip flop is a binary cell capable of storing one bit of information. An n –bit register has a group of n flip flops and is capable of storing any binary information containing n bits. Register mainly used for storing and shifting data entered into it from an external source.

A register capable of shifting its binary information either to the right or to the left is called a "Shift register".

Q2. Define the types of Shift Registers.

Ans. Shift registers are classified into the following types depending on the way in which the data is entered and retrieved.

Serial-in to Parallel-out (SIPO) - the register is loaded with serial data, one bit at a time, with the stored data being available in parallel form.

Serial-in to Serial-out (SISO) - the data is shifted serially "IN" and "OUT" of the register, one bit at a time in either a left or right direction under clock control.

Parallel-in to Serial-out (PISO) - the parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.

Parallel-in to Parallel-out (PIPO) - the parallel data is loaded simultaneously into the register, and transferred together to their respective outputs by the same clock pulse.

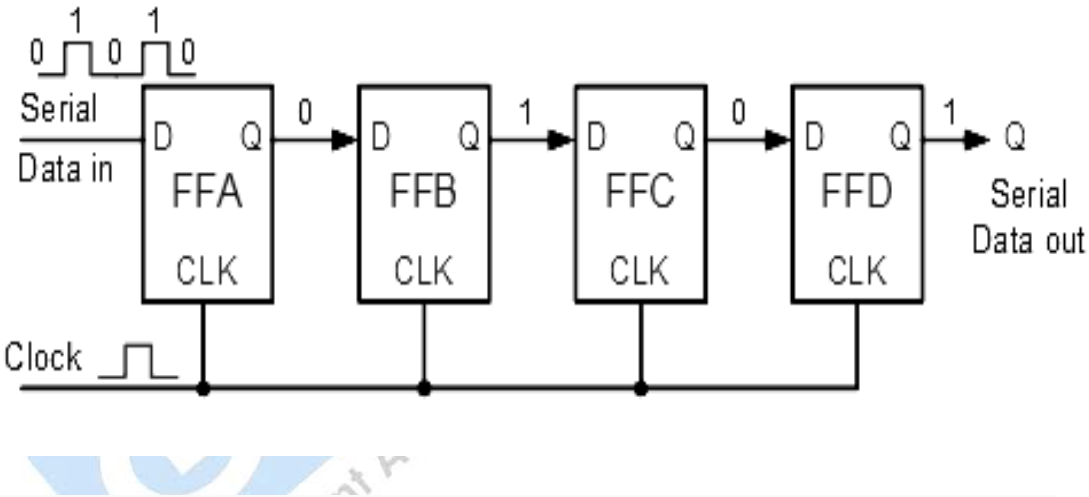
Q3. Explain Serial-in to Serial-out (SISO).

Ans. This **shift register** is very similar to the SIPO above, except were before the data was read directly in a parallel form from the outputs Q_A to Q_D, this time the data is allowed to flow straight through the register and out

of the other end. Since there is only one output, the DATA leaves the shift register one bit at a time in a serial pattern, hence the name **Serial-in to Serial-Out Shift Register** or **SISO**.

The SISO shift register is one of the simplest of the four configurations as it has only three connections, the serial input (SI) which determines what enters the left hand flip-flop, the serial output (SO) which is taken from the output of the right hand flip-flop and the sequencing clock signal (Clk). The logic circuit diagram below shows a generalized serial-in serial-out shift register.

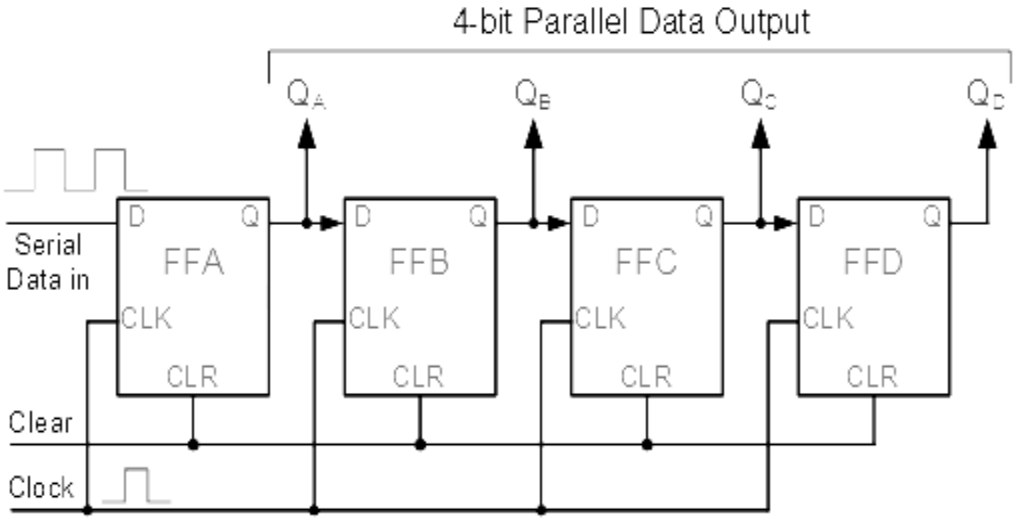
4-bit Serial-in to Serial-out Shift Register



This type of **Shift Register** also acts as a temporary storage device or as a time delay device for the data, with the amount of time delay being controlled by the number of stages in the register, 4, 8, 16 etc or by varying the application of the clock pulses. Commonly available IC's include the 74HC595 8-bit Serial-in/Serial-out Shift Register all with 3-state outputs.

Q4. Explain Serial-in to Parallel-out (SIPO).

Ans. 4-bit Serial-in to Parallel-out Shift Register



The operation is as follows. Assume that all the flip-flops (FFA to FFD) have just been RESET (CLEAR input) and that all the outputs Q_A to Q_D are at logic level "0" i.e, no parallel data output. If a logic "1" is connected to the DATA input pin of FFA then on the first clock pulse the output of FFA and therefore the resulting Q_A will be set HIGH to logic "1" with all the other outputs still remaining LOW at logic "0". Assume now that the DATA input pin of FFA has returned LOW again to logic "0" giving us one data pulse or 0-1-0.

The second clock pulse will change the output of FFA to logic "0" and the output of FFB and Q_B HIGH to logic "1" as its input D has the logic "1" level on it from Q_A. The logic "1" has now moved or been "shifted" one place along the register to the right as it is now at Q_A. When the third clock pulse arrives this logic "1" value moves to the output of FFC (Q_C) and so on until the arrival of the fifth clock pulse which sets all the outputs Q_A to Q_D back again to logic level "0" because the input to FFA has remained constant at logic level "0".

The effect of each clock pulse is to shift the data contents of each stage one place to the right, and this is shown in the following table until the

complete data value of 0-0-0-1 is stored in the register. This data value can now be read directly from the outputs of Q_A to Q_D. Then the data has been converted from a serial data input signal to a parallel data output. The truth table and following waveforms show the propagation of the logic "1" through the register from left to right as follows.

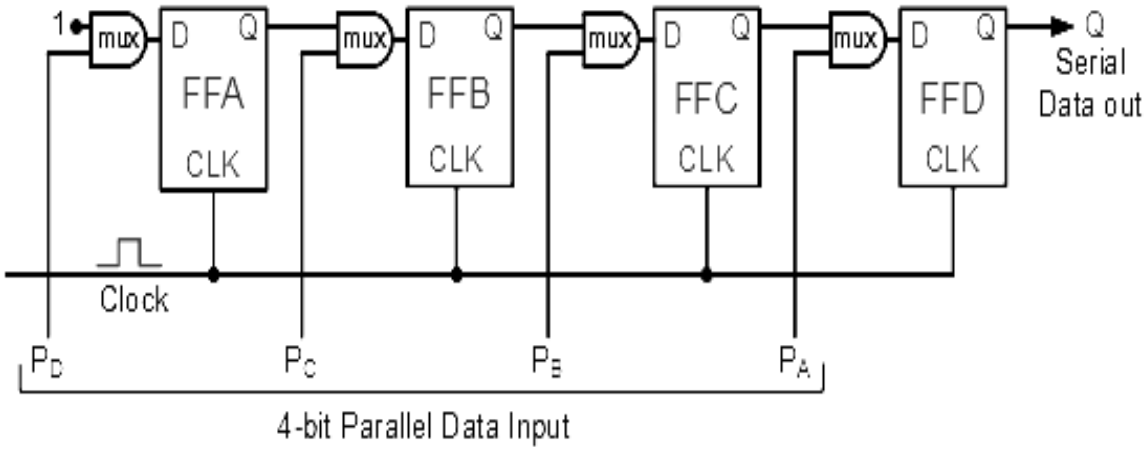
Basic Movement of Data through a Shift Register

Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0

Q5. Explain Parallel-in to Serial-out (PISO).

Ans. The Parallel-in to Serial-out shift register acts in the opposite way to the serial-in to parallel-out one above. The data is loaded into the register in a parallel format i.e. all the data bits enter their inputs simultaneously, to the parallel input pins P_A to P_D of the register. The data is then read out sequentially in the normal shift-right mode from the register at Q representing the data present at P_A to P_D. This data is outputted one bit at a time on each clock cycle in a serial format. It is important to note that with this system a clock pulse is not required to parallel load the register as it is already present, but four clock pulses are required to unload the data.

4-bit Parallel-in to Serial-out Shift Register

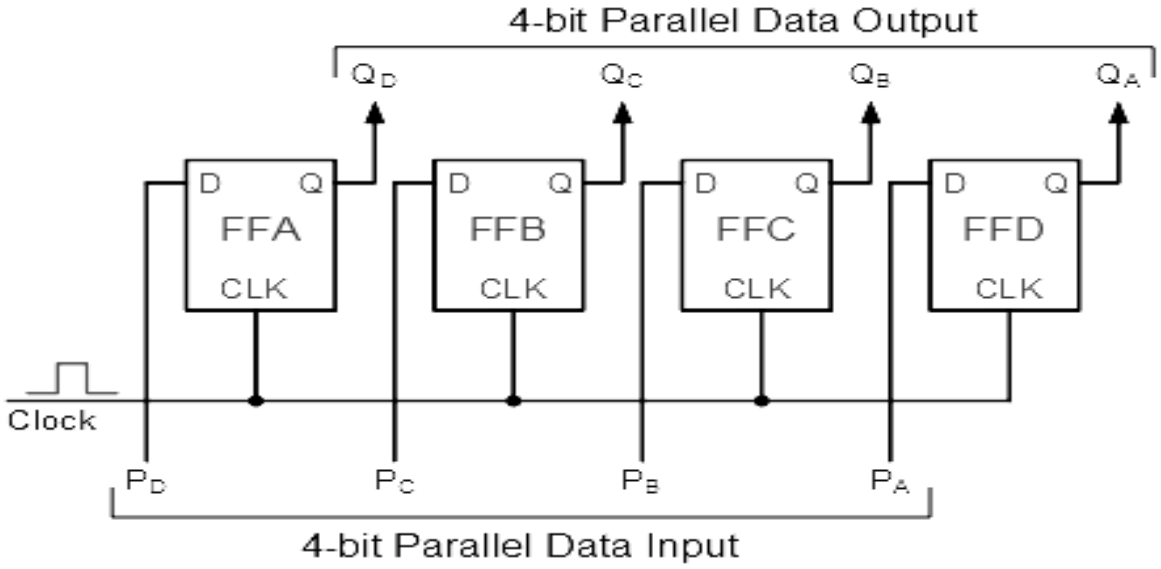


As this type of shift register converts parallel data, such as an 8-bit data word into serial format, it can be used to multiplex many different input lines into a single serial DATA stream which can be sent directly to a computer or transmitted over a communications line. Commonly available IC's include the 74HC166 8-bit Parallel-in/Serial-out Shift Registers.

Q6. Explain Parallel-in to Parallel-out (PIPO).

Ans. The final mode of operation is the Parallel-in to Parallel-out Shift Register. This type of register also acts as a temporary storage device or as a time delay device similar to the SISO configuration above. The data is presented in a parallel format to the parallel input pins P_A to P_D and then transferred together directly to their respective output pins Q_A to Q_A by the same clock pulse. Then one clock pulse loads and unloads the register. This arrangement for parallel loading and unloading is shown below.

4-bit Parallel-in to Parallel-out Shift Register



The PIPO shift register is the simplest of the four configurations as it has only three connections, the parallel input (PI) which determines what enters the flip-flop, the parallel output (PO) and the sequencing clock signal (Clk).

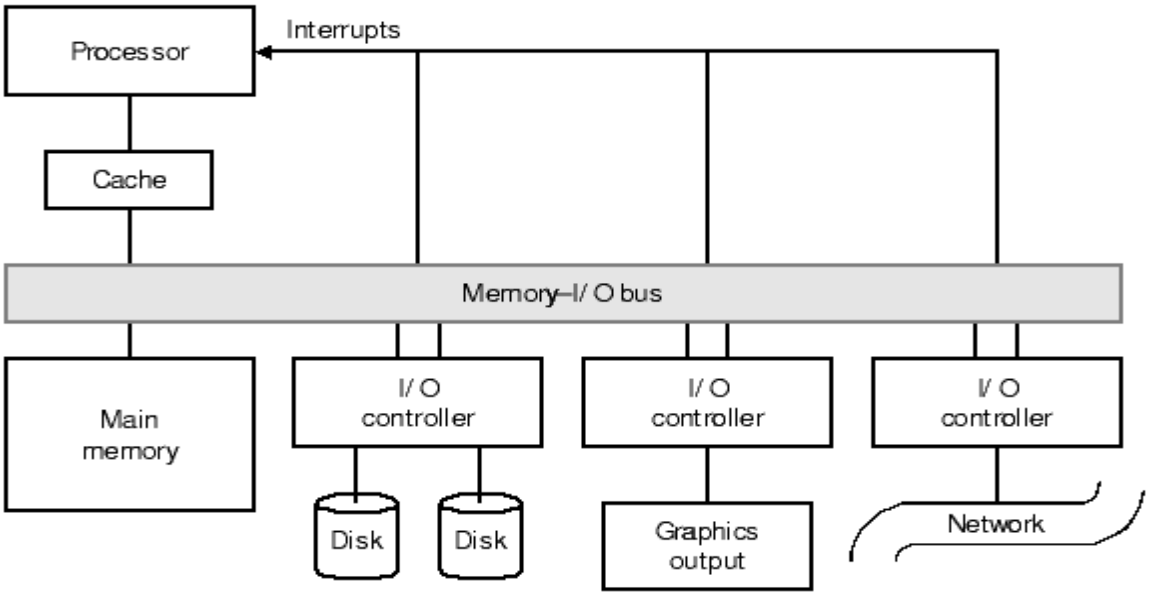
Similar to the Serial-in to Serial-out shift register, this type of register also acts as a temporary storage device or as a time delay device, with the amount of time delay being varied by the frequency of the clock pulses. Also, in this type of register there are no interconnections between the individual flip-flops since no serial shifting of the data is required.

Chapter 4

I/O Interface

Q1. Explain with diagram Memory I/O bus.

Ans. All components of a computer, including CPU, memory, and I/O devices (hard drive, floppy drive, keyboard, mouse, display, etc.) need to be connected by various buses.



Three types of information need to be transmitted between two devices in different lines of the bus: control, data and address. Typically, the control lines include request, grant, release, etc., for the proper communication and coordination of the bus usage by multiple devices.

Q2. Explain the ways how peripheral device communicate and Interface.

Ans. The operations in I/O devices are much slower than those in the CPU, various methods are used to facilitate the communications.

1. **Two ways for CPU to communicate with the I/O devices:**
   - Memory-mapped I/O: portions of the address space are assigned to the I/O devices, so that the read and write operations involving an I/O device are treated by the CPU in the same way as reading and writing the memory.
   - Special I/O instructions: to specify both the device number and the command.
2. **Two ways for I/O devices to communicate with the CPU:**
   - Polling: The I/O device behaves passively.

     After receiving command from the CPU, the I/O device carries out the command, e.g., to get some data ready for the CPU to get, and sets a status register. The CPU polls the device periodically by checking this register, and get the data when it indicates the availability of the data.

   - Interrupt: The I/O device behaves actively.

     When the I/O needs CPU's attention, e.g., some data are ready for CPU to get, it sends an interrupt signal to CPU. The process in CPU is interrupted so that it can attend the I/O's needs. When multiple I/O devices send out interrupts, the one with higher priority will be attended first.

     The CPU is interrupted by the mouse controller only when a mouse move is detected.

3. **Two ways for data transfer between the memory and I/O:**
   - CPU controlled: The communication and data transfer between memory and I/O is controlled by the CPU by either polling or interrupt;
   - Direct Memory Access (DMA): The communication and data transfer between memory and I/O is controlled by a special

controller. CPU is only interrupted by the I/O device when the transfer is complete or error occurs.

Q3. Explain CISC .

Ans. A Complex Instruction Set Computer (CISC) supplies a large number of complex instructions at the assembly language level. *Assembly language* is a low-level computer programming language in which each statement corresponds to a single machine instruction. CISC instructions facilitate the extensive manipulation of low-level computational elements and events such as memory, binary arithmetic, and addressing. This particular architectural methodology requires smaller binary files (because each CISC command accomplishes so much, relatively speaking) but involves relatively slow execution of each individual instruction (because the processor must perform more binary manipulations to fulfill each instruction). The goal of the CISC architectural philosophy is to make microprocessors easy and flexible to program and to provide for more efficient memory use.

CISC Problem

1.Performance tuning unsuccessful

- Rarely used high-level instructions
- Sometimes slower than equivalent sequence

2.High complexity

- Pipelining bottlenecks → lower clock rates
- Interrupt handling can complicate even more

3.Marketing

- Prolonged design time and frequent microcode errors hurt competitiveness

Q4. Explain RISC.

Ans. Reduced instruction set computing, or RISC ,is a CPU design strategy based on the insight that simplified (as opposed to complex) instructions can provide higher performance if this simplicity enables much faster execution of each instruction. A computer based on this strategy is a reduced instruction set computer .

1.Low complexity

- Generally results in overall speedup
- Less error-prone implementation by hardwired logic or simple microcode.

2.VLSI implementation advantages

- Less transistors
- Extra space: more registers, cache

3.Marketing

- Reduced design time, less errors, and more options increase competitiveness

Q5. Define instruction cycle.

Ans. The time period during which one instruction is fetched from memory and executed when a computer is given an instruction in machine language.

There are typically four stages of an instruction cycle that the CPU carries out:

1. Fetch the instruction from memory. This step brings the instruction into the *instruction register*, a circuit that holds the instruction so that it can be decoded and executed.
2. Decode the instruction.
3. Read the effective address from memory if the instruction has an indirect address.
4. Execute the instruction.

Steps 1 and 2 are called the *fetch cycle* and are the same for each instruction. Steps 3 and 4 are called the *execute cycle* and will change with each instruction.

The term refers to both the series of four steps and also the amount of time that it takes to carry out the four steps.

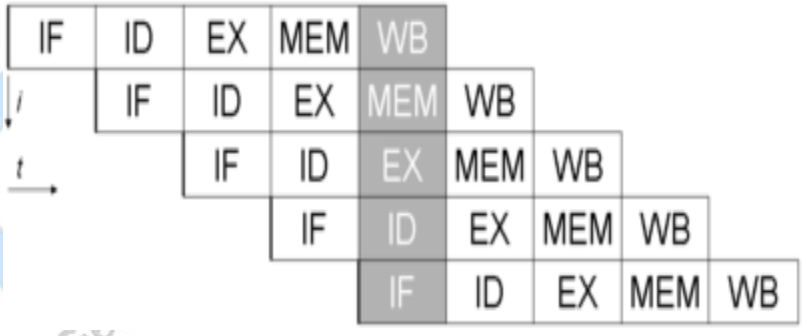
An instruction cycle also is called *machine cycle*.

Q6. Define Instruction Pipeline.

Ans. An instruction pipeline is a technique used in the design of modern microprocessors, microcontrollers and CPUs to increase their instruction throughput (the number of instructions that can be executed in a unit of time).

The main idea is to divide the processing of a CPU instruction, as defined by the instruction **microcode**, into a series of independent steps of micro-operations , with storage at the end of each step. This allows the CPUs control logic to handle instructions at the processing rate of the slowest step, which is much faster than the time needed to process the instruction as a single step.

The term pipeline refers to the fact that each step is carrying a single microinstruction and each step is linked to another step .



Basic five-stage pipeline in a RISC machine (IF = Instruction Fetch, ID = Instruction Decode, EX = Execute, MEM = Memory access, WB = Register write back). The vertical axis is successive instructions, the horizontal axis is time. So in the green column, the earliest instruction is in WB stage, and the latest instruction is undergoing instruction fetch.

Advantages of Pipelining:

1. The cycle time of the processor is reduced; increasing the instruction throughput.
2. If pipelining is used, the CPU Arithmetic logic unit can be designed faster, but more complex.
3. Pipelining in theory increases performance over an un-pipelined core by a factor of the number of stages and the code is ideal for pipeline execution.
4. Pipelined CPUs generally work at a higher clock frequency than the RAM clock frequency, increasing computers overall performance.

Disadvantages of Pipelining:

Pipelining has many disadvantages though there are a lot of techniques used by CPUs and compilers designers to overcome most of them of them; following is a list of common drawbacks:

1. The design of a non-pipelined processor simpler and cheaper to manufacture, non-pipelined processor executes only a single instruction at a time. This prevents branch delays (in Pipelining, every branch is delayed) as well as problems when serial instructions being executed concurrently.
2. In pipelined processor, insertion of flip flops between modules increases the instruction latency compared to a non-pipelined processor.
3. A non-pipelined processor will have a defined instruction throughput. The performance of a pipelined processor is much harder to predict and may vary widely for different programs.

Chapter 5

Instruction and Addressing

Q1. Explain instruction formats .

Ans. An instruction format defines the layout of the bits of ab instruction , in terms of its constituent parts.
The bits of the instruction are divided into groups called fields. The most common fields are:

- An operation code that specifies the operation to be performed.
- An address field that specifies a memory address or register.
- A mode field that tells us how the operand or the effective address of the operand is to be found out.

Opcode-Field	Address-Field

- Op-field: specifies the operation to be performed;
- Address-field: provides operands or the CPU register/MM addresses of the operands.

Q2. What are the types of instruction formats. Explain with e.g.

Ans. 1. Three address instructions
2. Two address instructions
3. One address instructions
4. Zero address instruction

Three address instructions

Computers with three address instructions use three address fields to specify either a processor register or a memory operand

Example: X= (A+B)*C+D)

where A, B, C, D and X are five main memory locations representing five variables;

3-address format:

Assume variables A, B, C, D, and X are stored in MM locations labeled by their names.

ADD R1, A, B R1←M[A]+M[B]
ADD R2,C,D R2←M[C]+M[D]
MUL X,R1,R2 M[X]←R1*R2

Note: here we assume an instruction OP dst src1 src2 means :-

dst←[src1]*[src2]

where src1 and src2 are the *source operand*, dst is the *destination operand*, and * represents the operation specified in Op-code field OP.

Two address instructions

Computers that use this type of instruction have two addresses specified in their instructions

Example: X= (A+B)*C+D)

MOV R1, A R1 ← [A]
ADD R1, B R1 ← [B] + [R1]
MOV R2, C R2 ← [C]
ADD R2 ,D R2 ← [D] + [R2]
MUL R2 ,R1 R2 ← [R1] * [R2]

MOV X ,R2 X ← [R2]

Note: here we assume an instruction OP dst src means :-

dst←[dst]*[src]

where src is the *source operand*, dst is the *destination operand*, and * represents the operation specified in Op-code field OP.

One address instructions

Always use an implied accumulator (AC).

LOAD	A	#	AC ← [A]
ADD	B	#	AC ← [AC] + [B]
STORE	R	#	R ← [AC]
LOAD	C	#	AC ← [C]
ADD	D	#	AC ← [AC] + [D]
MUL	R	#	AC ← [AC] × [R]
STORE	X	#	X ← [AC]

Zero address instructions

used in stack-organized computer.

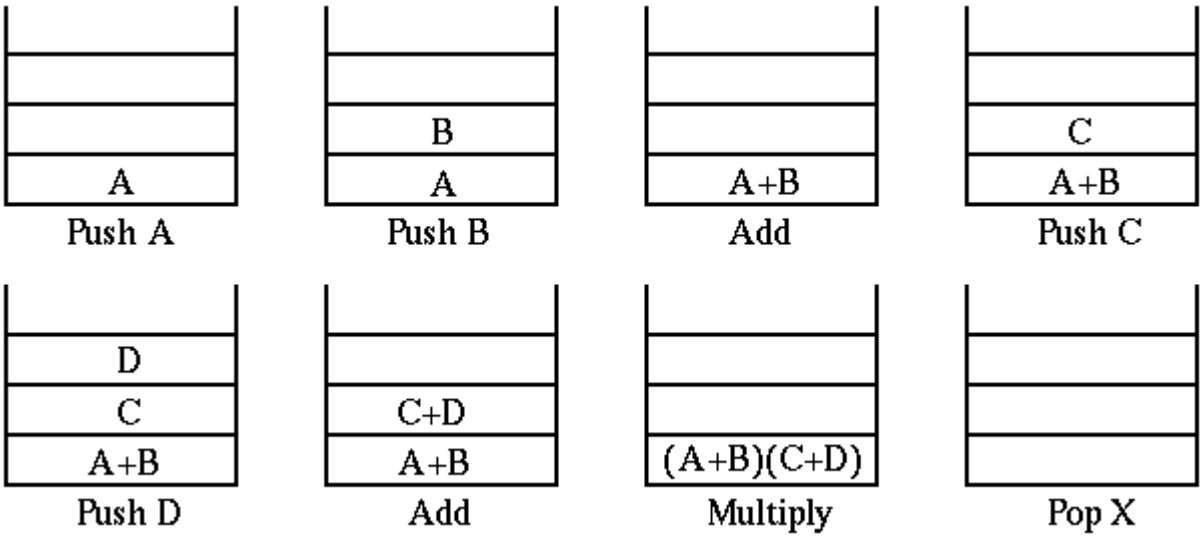
First, the given notation of the operation is converted into ``reversed Polish notation (RPN)''

$$(A + B) \times (C + D) \Longrightarrow AB + CD + \times$$

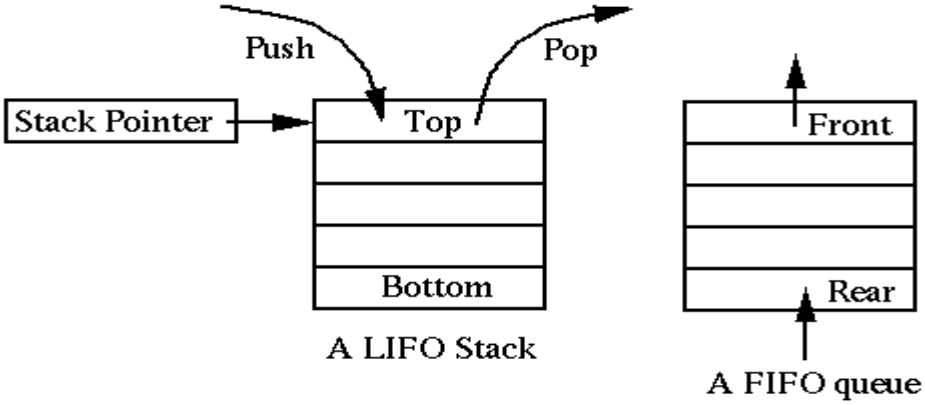
then execute this program:

```
PUSH    A
PUSH    B
ADD
```

```
PUSH    C
PUSH    D
ADD
MUL
POP     X
```



- **Stack** A last-in, first-out (LIFO) data structure.
- **Queue** A first-in, first-out (FIFO) data structure.



Q3. What are the Addressing modes.

Ans. The addressing mode specifies a rule for interpreting or modifying the address field of the instruction.

Types of Addressing Modes:-

- Implied addressing mode
- Immediate addressing mode
- Direct addressing mode
- Indirect addressing mode
- Register addressing mode
- Register Indirect addressing mode
- Auto increment or Auto decrement addressing mode
- Relative addressing mode
- Indexed addressing mode
- Base register addressing mode

Addressing Modes	**Example Instruction**	**Meaning**	**When used**
Register	Add R4,R3	R4 <- R4 + R3	When a value is in a register
Immediate	Add R4, #3	R4 <- R4 + 3	For constants
Displacement	Add R4, 100(R1)	R4 <- R4 + M[100+R1]	Accessing local variables
Register deffered	Add R4,(R1)	R4 <- R4 + M[R1]	Accessing using a pointer or a computed address
Indexed	Add R3,	R3 <- R3 +	Useful in array

	(R1 + R2)	M[R1+R2]	addressing: R1 - base of array R2 - index amount
Direct	Add R1, (1001)	R1 <- R1 + M[1001]	Useful in accessing static data
Memory deferred	Add R1, @(R3)	R1 <- R1 + M[M[R3]]	If R3 is the address of a pointer *p*, then mode yields **p*
Auto-increment	Add R1, (R2)+	R1 <- R1 +M[R2] R2 <- R2 + *d*	Useful for stepping through arrays in a loop. R2 - start of array *d* - size of an element
Auto-decrement	Add R1,- (R2)	R2 <-R2-*d* R1 <- R1 + M[R2]	Same as autoincrement. Both can also be used to implement a stack as push and pop
Scaled	Add R1, 100(R2)[R3]	R1<- R1+M[100+R2+R3**d*]	Used to index arrays. May be applied to any base addressing mode in some machines.

Q4. What is assembly language?

Ans. An assembly language is a low-level programming language for computers, microprocessors, microcontrollers, and other programmable devices. It implements a symbolic representation of the machine codes and other constants needed to program a given CPU architecture. This representation is usually defined by the hardware manufacturer, and is based on mnemonics that symbolize processing steps (instructions), processor registers, memory locations, and other language features. An assembly language is thus specific to a certain physical (or virtual) computer architecture. This is in contrast to most high-level programming languages.

Q5. Define Assembler

Ans. Assembler creates object code by translating assembly instruction mnemonics into opcodes, and by resolving symbolic names for memory locations and other entities The use of symbolic references is a key feature of assemblers, saving tedious calculations and manual address updates after program modifications. Most assemblers also include macro facilities for performing textual substitution—e.g., to generate common short sequences of instructions as inline, instead of called subroutines.

Q6. Explain the Number of passes of an assembler.

Ans. There are two types of assemblers based on how many passes through the source are needed to produce the executable program.

- One-pass assemblers go through the source code once. Any symbol used before it is defined will require "errata" at the end of the object code telling the linker or the loader to "go back" and overwrite a placeholder which had been left where the as yet undefined symbol was used.
- Two-pass assemblers create a table with all symbols and their values in the first pass, then use the table in a second pass to generate code.
- In both cases, the assembler must be able to determine the size of each instruction on the first or only pass in order to calculate the addresses of symbols. This means that if the size of an operation referring to an

operand defined later depends on the type or distance of the operand, the assembler will make a pessimistic estimate when first encountering the operation, and if necessary pad it with one or more "no-operation" instructions in the second pass or the errata.

The original reason for the use of one-pass assemblers was speed of assembly; however, modern computers perform two-pass assembly without unacceptable delay. The advantage of the two-pass assembler is that the absence of a need for errata makes the linker (or the loader if the assembler directly produces executable code) simpler and faster

Q7. Explain Assembly directives.

Ans. Assembly directives, also called pseudo opcodes, pseudo-operations or pseudo-ops, are instructions that are executed by an assembler at assembly time, not by a CPU at run time. They can make the assembly of the program dependent on parameters input by a programmer, so that one program can be assembled different ways, perhaps for different applications. They also can be used to manipulate presentation of a program to make it easier to read and maintain.

Q8. Explain Pseudo instructions.

Ans. Pseudo instructions means "fake instruction". When designing a modern ISA, one criteria is to decide whether an instruction should be part of the ISA or not. Initially, a use for the instruction, it was often added to their instruction set. This lead to bloated and slow implementations.

By taking out some instructions, assembly language programmers would find it a little harder to write code. To make it easier for them, pseudo instructions were added. Pseudo instructions do not correspond to real MIPS instructions.

Instead, the assembler, a program that converts assembly language programs to machine code, would then translate pseudo instructions to real instructions, usually requiring at least one on more instructions.

Pseudo instructions not only make it easier to program, it can also add clarity to the program, by making the intention of the programmer more clear.

Q 9. Explain Macro Instruction .

Ans. A macro instruction is a line of computer program coding that results in one or more lines of program coding in the target programming language, sets variables for use by other statements, etc.. In the mid 1950s, when assembly language programming was commonly used to write programs for digital computers, the use of macro instructions was initiated for two main purposes: to reduce the amount of program coding that had to be written by generating several assembly language statements from one macro instruction and to enforce program writing standards, e.g. specifying input/output commands in standard ways. Macro instructions were effectively a middle step between assembly language programming and the high-level programming languages that followed, such as FORTRAN and COBOL.

Q.10 Explain linking and loading

Ans. Linking and Loading

Loading is the process of copying an executable image into memory.

* more sophisticated loaders are able to relocate images to fit into available memory

* must readjust branch targets, load/store addresses

Linking is the process of resolving symbols between independent object files.

* suppose we define a symbol in one module, and want to use it in another

* some notation, such as .EXTERNAL, is used to tell assembler that a symbol is defined in another module

* linker will search symbol tables of other modules to resolve symbols and complete code generation before loading

Chapter 6

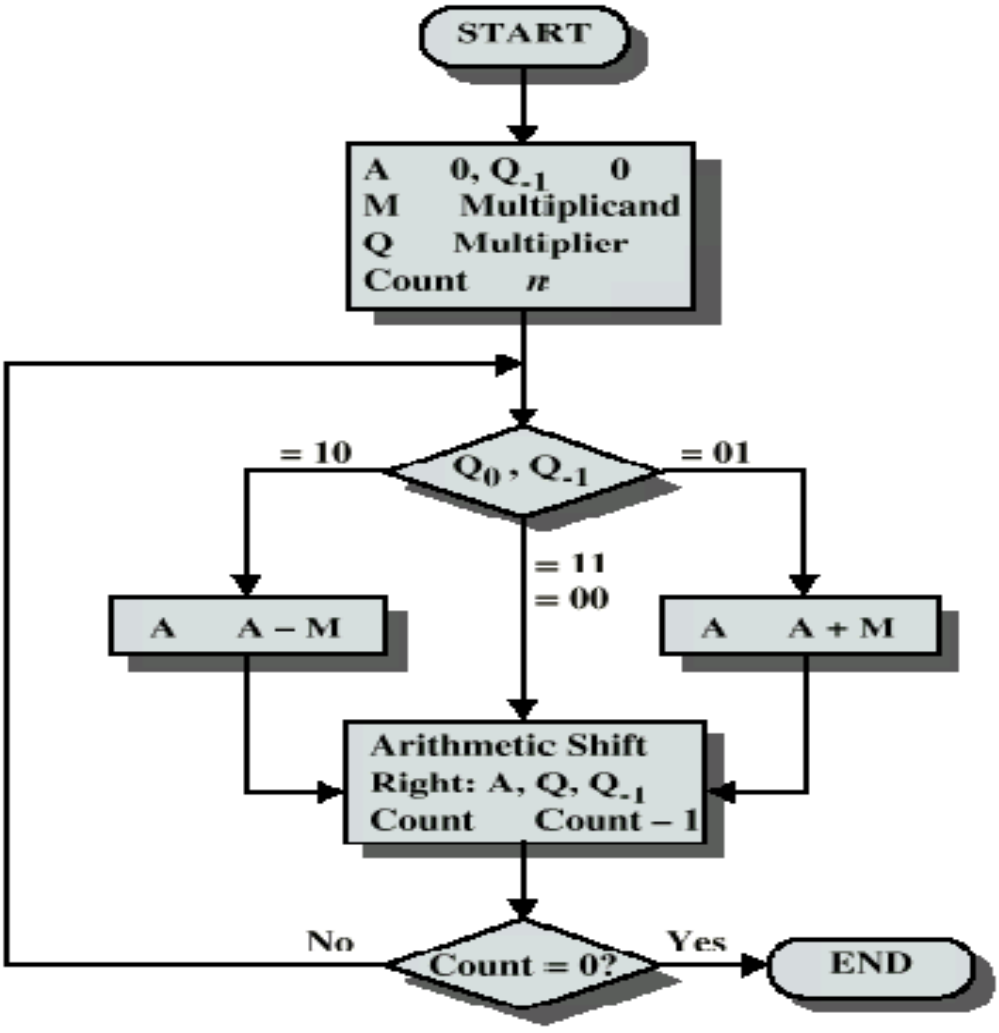
Arithmetic /Logic unit

Q1. Explain Booth's Algo.

Ans. Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation. Booth's algorithm involves repeatedly adding one of two predetermined values *A* and *S* to a product *P*, then performing a rightward arithmetic shift on *P*. Let **m** and **r** be the multiplicand and multiplier, respectively; and let *x* and *y* represent the number of bits in **m** and **r**.

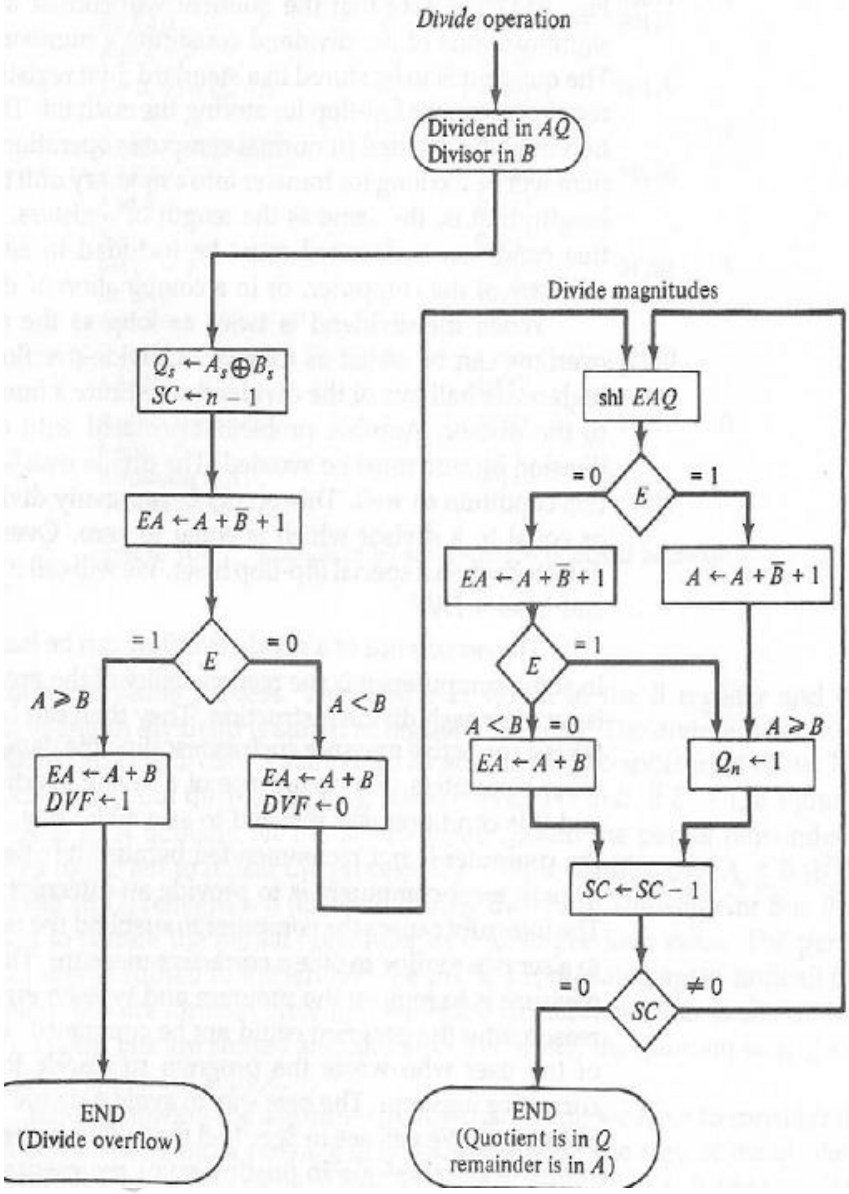
1. Determine the values of *A* and *S*, and the initial value of *P*. All of these numbers should have a length equal to (*x* + *y* + 1).
    1. A: Fill the most significant (leftmost) bits with the value of **m**. Fill the remaining (*y* + 1) bits with zeros.
    2. S: Fill the most significant bits with the value of (−**m**) in two's complement notation. Fill the remaining (*y* + 1) bits with zeros.
    3. P: Fill the most significant *x* bits with zeros. To the right of this, append the value of **r**. Fill the least significant (rightmost) bit with a zero.
2. Determine the two least significant (rightmost) bits of *P*.
    1. If they are 01, find the value of *P* + *A*. Ignore any overflow.
    2. If they are 10, find the value of *P* + *S*. Ignore any overflow.
    3. If they are 00, do nothing. Use *P* directly in the next step.
    4. If they are 11, do nothing. Use *P* directly in the next step.
3. Arithmetically shift the value obtained in the 2nd step by a single place to the right. Let *P* now equal this new value.
4. Repeat steps 2 and 3 until they have been done *y* times.

5. Drop the least significant (rightmost) bit from *P*. This is the product of **m** and **r**.



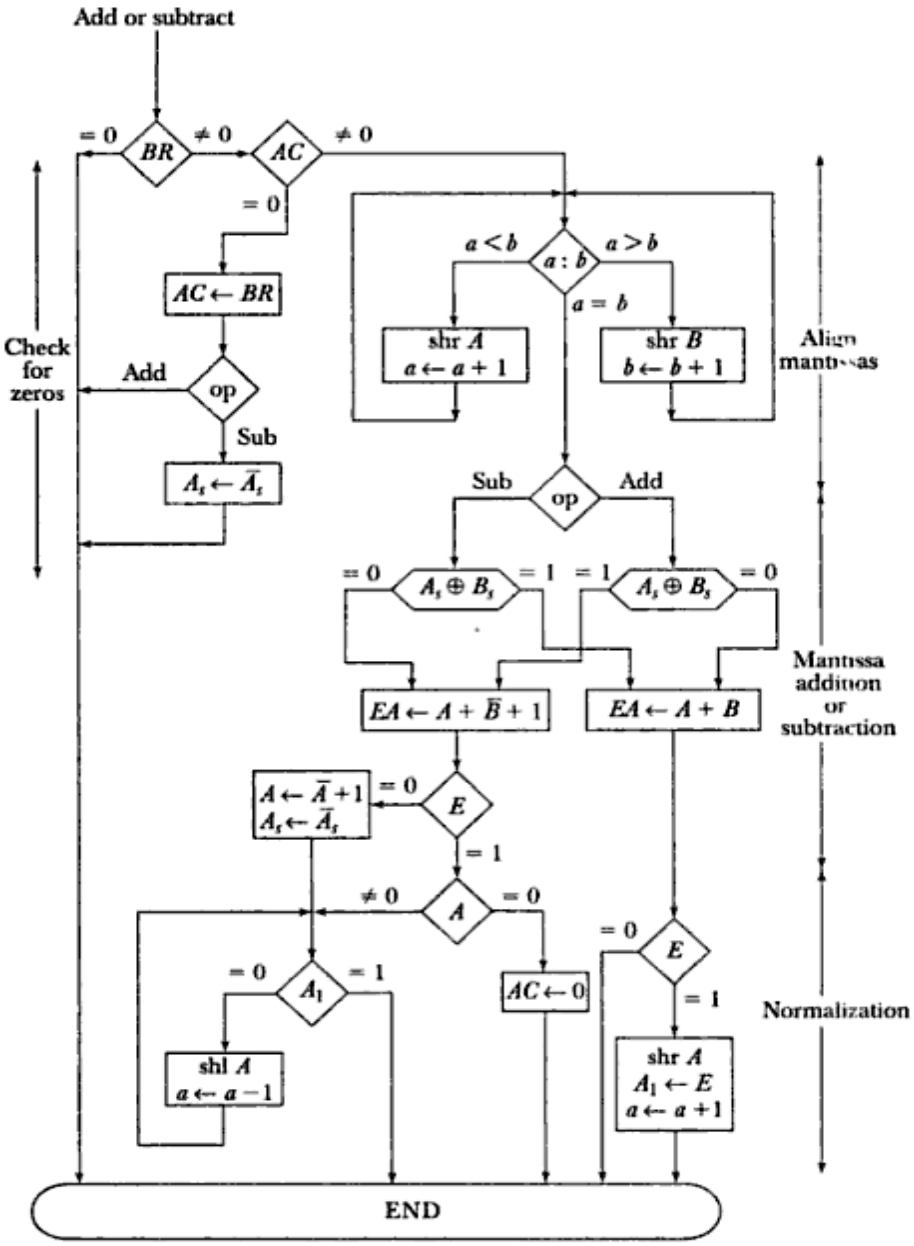
Booth's algo for two's complement multiplication

Q2. Explain division algorithm.
Ans



Q3. Explain with flowchart addition and subtraction of floating point numbers.

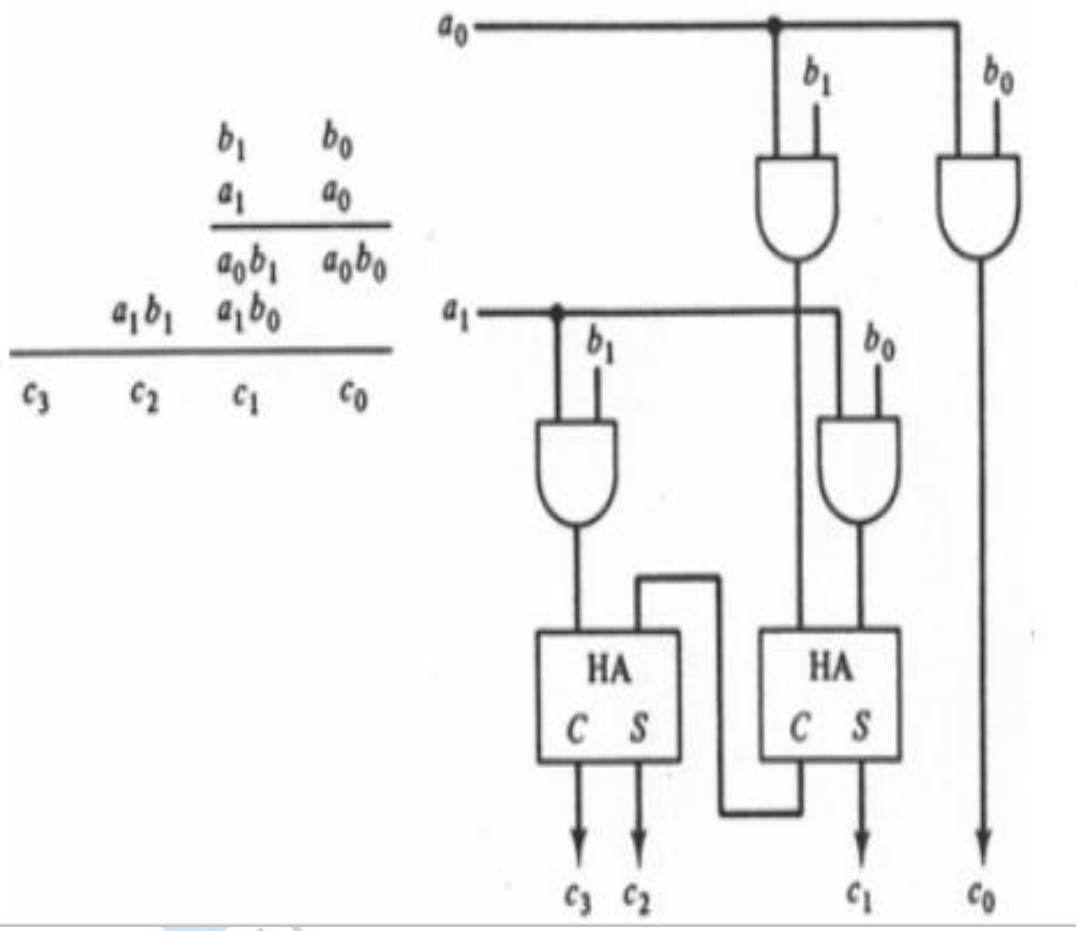
Ans.



Q4. Explain array multiplier.

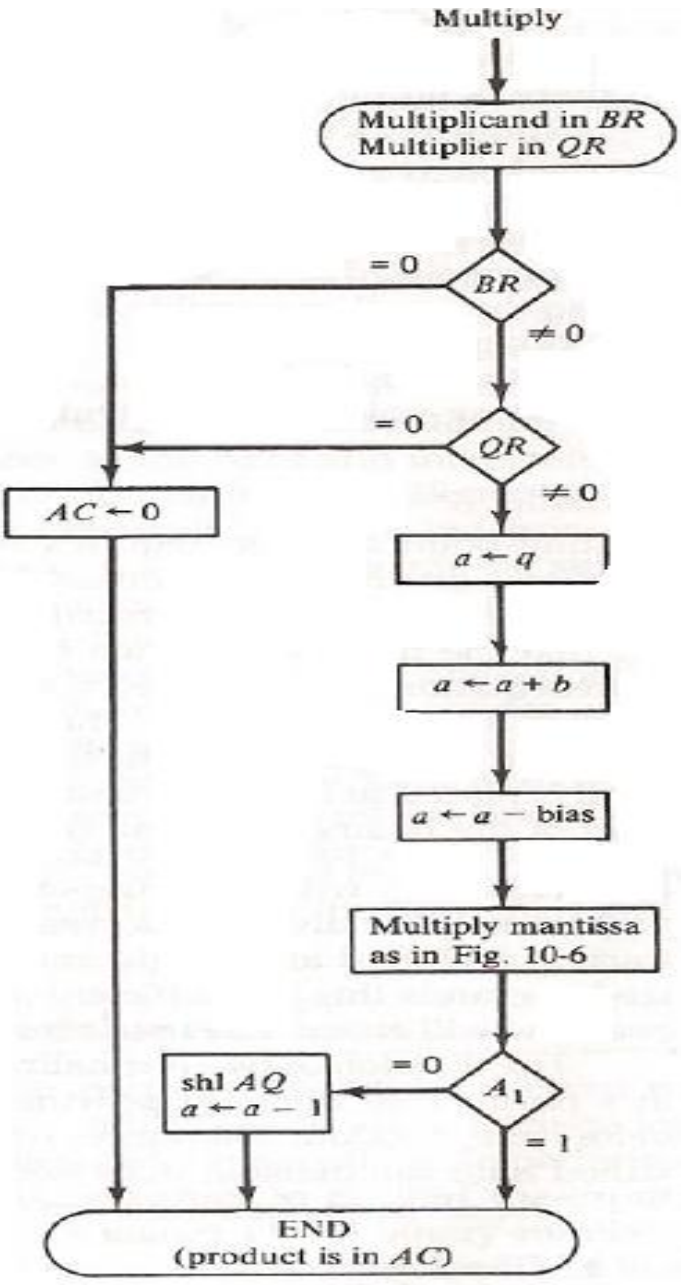
Ans. It is a Combination circuit , Requires large number of gates. Its product generated in one micro operation

- Became feasible after integrated circuits developed
- Needed for j multiplier and k multiplicand bits
„ j x k AND gates
„ j – 1 k-bit adders to produce product of j + k bit



2-bit by 2-bit Array Multiplier

Q5. Explain Multiplication of floating point numbers.
Ans.


Multiply
Multiplicand in BR
Multiplier in QR
BR
= 0
≠ 0
QR
= 0
≠ 0
AC ← 0
a ← q
a ← a + b
a ← a − bias
Multiply mantissa
as in Fig. 10-6
A1
= 0
= 1
shl AQ
a ← a − 1
END
(product is in AC)

Chapter 7

Memory

Q1. Explain Ram and Rom Chip

Ans. ***R****andom* ***a****ccess* ***m****emory*, a type of computer memory that can be accessed randomly; that is, any byte of memory can be accessed without touching the preceding bytes. RAM is the most common type of memory found in computers and other devices, such as printers.

There are two different types of RAM: DRAM (dynamic random access memory) and SRAM (static random access memory).

The two types differ in the technology they use to hold data, with dram being the more common type. In terms of speed, SRAM is faster. DRAM needs to be refreshed thousands of times per second while SRAM does not need to be refreshed, which is what makes it faster than DRAM. DRAM supports access times of about 60 nanoseconds, SRAM can give access times as low as 10 nanoseconds. Despite SRAM being faster, it's not as commonly used as dram because it's so much more expensive. Both types of RAM are *volatile*, meaning that they lose their contents when the power is turned off.

R*ead-****o****nly* ***m****emory*, computer memory on which data has been prerecorded. Once data has been written onto a ROM chip, it cannot be removed and can only be read. Unlike main memory (RAM), ROM retains its contents even when the computer is turned off. ROM is referred to as being *nonvolatile*, whereas RAM is *volatile.*

Q2. Explain auxiliary memory.

Ans. Auxiliary memory units are among computer peripheral equipment. They trade slower access rates for greater storage capacity and data stability. Auxiliary memory holds programs and data for future use, and, because it is nonvolatile (like ROM), it is used to store inactive programs and to archive data. Early forms of auxiliary storage included punched

paper tape, punched cards, and magnetic drums. the most common forms of auxiliary storage have been magnetic disks, magnetic tapes, and optical discs.

Magnetic tape

Magnetic tape, similar to the tape used in tape recorders, has also been used for auxiliary storage, primarily for archiving data. Tape is cheap, but access time is far slower than that of a magnetic disk because it is sequential-access memory—i.e., data must be sequentially read and written as a tape is unwound, rather than retrieved directly from the desired point on the tape.

Optical discs

Another form of largely read-only memory is the optical compact disc, developed from videodisc technology during the early 1980s. Data are recorded as tiny pits in a single spiral track on plastic discs that range from 3 to 12 inches (7.6 to 30 cm) in diameter, though a diameter of 4.8 inches (12 cm) is most common.

Q3. Explain cache memory.

Ans. The *cache* is a small amount of high-speed memory, usually with a memory cycle time comparable to the time required by the CPU to fetch one instruction. The cache is usually filled from main memory when instructions or data are fetched into the CPU. Often the main memory will supply a wider data word to the cache than the CPU requires, to fill the cache more rapidly. The amount of information which is replaces at one time in the cache is called the *line size* for the cache. This is normally the width of the data bus between the cache memory and the main memory.

Q4. Explain the working of cache memory

Ans. When the *processor* needs to read or write a location in main memory, it first checks whether that memory location is in the cache. This is accomplished by comparing the address of the memory location to all tags in the cache that might contain that address. If the processor finds that the memory location is in the cache, we say that a *cache hit* has occurred; otherwise, we speak of a *cache miss*. In the case of a cache hit, the processor immediately reads or writes the data in the cache line. The proportion of accesses that result in a cache hit is known as the *hit rate*, and is a measure of the effectiveness of the cache for a given program or algorithm.

In the case of a miss, the cache allocates a new entry, which comprises the tag just missed and a copy of the data. The reference can then be applied to the new entry just as in the case of a hit. Read misses delay execution because they require data to be transferred from a much slower memory than the cache itself. Write misses may occur without such penalty since the data can be copied in the background.

In order to make room for the new entry on a cache miss, the cache has to *evict* one of the existing entries. The heuristic that it uses to choose the entry to evict is called the *replacement policy*. The fundamental problem with any replacement policy is that it must predict which existing cache entry is least likely to be used in the future. One popular replacement policy, LRU, replaces the least recently used entry. Defining some memory ranges non *cacheable* avoids affecting performance by storing in caches information which are never re-used or seldom used. Cache misses are simply ignored for not cacheable data. Cache entries may also be disabled or locked depending on the context.

If data are written to the cache, they must at some point be written to main memory as well. The timing of this write is controlled by what is known as the *write policy*. In a *write-through* cache, every write to the cache causes a write to main memory. Alternatively, in a *write-back* or *copy-back* cache, writes are not immediately mirrored to the main memory. Instead, the cache tracks which locations have been written over (these locations are marked *dirty*). The data in these locations are written back to the main memory when that data is evicted from the cache. For this reason, a miss in a write-back cache may sometimes require *two* memory accesses to service: one to first write the dirty location to memory and then another to read the new location from memory.

There are intermediate policies as well. The cache may be write-through, but the writes may be held in a store data queue temporarily, usually so that multiple stores can be processed together (which can reduce bus turnarounds and so improve bus utilization).

Q5. Explain Virtual memory.

Ans. In computing, virtual memory is a memory management technique developed for multitasking kernels. This technique virtualizes a computer architecture's various forms of computer data storage (such as random-access memory and disk storage), allowing a program to be *designed as*

though there is only one kind of memory, "virtual" memory, which behaves like directly addressable read/write memory (RAM).

Most modern operating systems that support virtual memory also run each process in its own dedicated address space, allowing a program to be *designed as though*t has sole access to the virtual memory.

Systems that employ virtual memory:

-use hardware memory more efficiently than do systems without virtual memory.

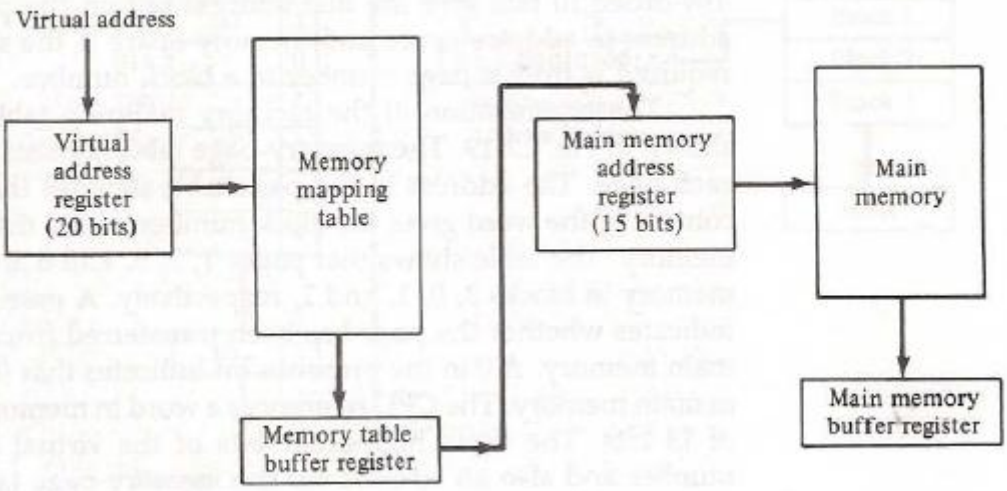
-make the programming of applications easier:

-by hiding fragmentation,

-by delegating to the kernel the burden of managing the memory hierarchy (there is no need for the program to handle overlays explicitly),

-and, when each process is run in its own dedicated address space, by obviating the need to relocate program code or to access memory with relative addressing.

Memory virtualization is a generalization of the concept of virtual memory.



Memory table for mapping virtual address

Q6. Explain Paging.

Ans. Paging is a memory management technique which widely uses virtual memory concept. When paging is used, the processor divides the linear address space into fixed-size pages (of 4KBytes, 2 MBytes, or 4 MBytes in length) that can be mapped into physical memory and/or disk storage.

When a program (or task) references a logical address in memory, the processor translates the address into a linear address and then uses its paging mechanism to translate the linear address into a corresponding physical address

The main functions of paging are performed when a program tries to access pages that are not currently mapped to physical memory (RAM). This situation is known as a page fault. The operating system must then take control and handle the page fault, in a manner invisible to the program. Therefore, the operating system must:

-Determine the location of the data in auxiliary storage.

-Obtain an empty page frame in RAM to use as a container for the data.

-Load the requested data into the available page frame.

-Update the page table to show the new data.

-Return control to the program, transparently retrying the instruction that caused the page fault.

Until there is not enough RAM to store all the data needed, the process of obtaining an empty page frame does not involve removing another page from RAM. If all page frames are non-empty, obtaining an empty page frame requires choosing a page frame containing data to empty. If the data in that page frame has been modified since it was read into RAM (i.e., if it has become "dirty"), it must be written back to its location in secondary storage before being freed; otherwise, the contents of the page's page frame in RAM are the same as the contents of the page in secondary storage, so it does not need to be written back to secondary storage. If a reference is then made to that page, a page fault will occur, and an empty page frame must be obtained and the contents of the page in secondary storage again read into that page frame.

Efficient paging systems must determine the page frame to empty by choosing one that is least likely to be needed within a short time. There are various page replacement algorithms that try to do this. Most operating systems use some approximation of the least recently used (LRU) page replacement algorithm (the LRU itself cannot be implemented on the current hardware) or a working set-based algorithm.

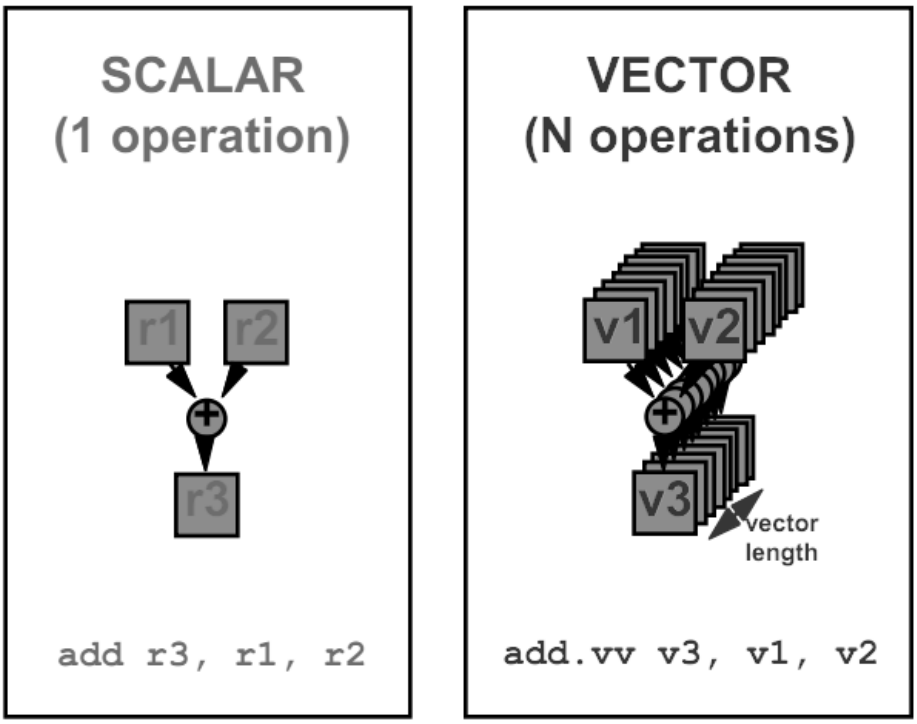
To further increase responsiveness, paging systems may employ various strategies to predict which pages will be needed soon. Such systems will attempt to load pages into main memory preemptively, before a program references them.

Chapter 8

Processor

Q1. Explain vector processors.

Ans. Vector processors have high-level operations that work on linear arrays of numbers: "vectors"



Properties of Vector Processors:-

- Each result independent of previous result

=> long pipeline, compiler ensures no dependencies

=> high clock rate

- Vector instructions access memory with known pattern

=> highly interleaved memory
=> amortize memory latency of over » 64 elements
=> no (data) caches required! (Do use instruction cache)
• Reduces branches and branch problems in pipelines
• Single vector instruction implies lots of work (» loop)
=> fewer instruction fetches
Styles of Vector Architectures:-
• memory-memory vector processors: all vector operations are memory to memory
• vector-register processors: all vector operations between vector registers.

Q2. Explain array processing.

Ans. Array processing is signal processing of the outputs of an array of sensors to:
Enhance the signal-to-interference-plus-noise ratio (SINR) compared to that of a single sensor using conventional or adaptive beam forming.
Determine the number of emitting sources, the locations of these sources, their waveforms, and other signal parameters. Track multiple moving sources. Array processing is used in radar, sonar, seismic exploration, anti-jamming and wireless communications. One of the main advantages of using array processing along with an array of sensors is a smaller foot-print. The problems associated with array processing include the number of sources used, their direction of arrivals, and their signal waveforms.[1]
There are four assumptions in array processing. The first assumption is that there is uniform propagation in all directions of isotropic and no dispersive medium. The second assumption is that for far field array processing, the radius of propagation is much greater than size of the array and that there is plane wave propagation. The third assumption is that there is a zero mean white noise and signal, which shows uncorrelation. Finally, the last assumption is that there is no coupling and the calibration is perfect.

Q3. What is multiprocessing?

Ans. Multiprocessing is the use of two or more central processing units (CPUs) within a single computer system. The term also refers to the ability of a system to support more than one processor and/or the ability to allocate tasks between them.[1] There are many variations on this basic theme, and

the definition of multiprocessing can vary with context, mostly as a function of how CPUs are defined (multiple cores on one die, multiple dies in one package, multiple packages in one system unit, etc.).

Multiprocessing sometimes refers to the execution of multiple concurrent software processes in a system as opposed to a single process at any one instant. However, the terms multitasking or multiprogramming are more appropriate to describe this concept, which is implemented mostly in software, whereas multiprocessing is more appropriate to describe the use of multiple hardware CPUs. A system can be both multiprocessing and multiprogramming, only one of the two, or neither of the two of them.

Software implementation issues:-

In a single instruction stream, single data stream computer one processor sequentially processes instructions, each instruction processes one data item. One example is the "von Neumann" architecture with RISC.

1. SIMD multiprocessing

In a single instruction stream, multiple data stream computer one processor handles a stream of instructions, each one of which can perform calculations in parallel on multiple data locations.

SIMD multiprocessing is well suited to parallel or vector processing, in which a very large set of data can be divided into parts that are individually subjected to identical but independent operations. A single instruction stream directs the operation of multiple processing units to perform the same manipulations simultaneously on potentially large amounts of data.

For certain types of computing applications, this type of architecture can produce enormous increases in performance, in terms of the elapsed time required to complete a given task. However, a drawback to this architecture is that a large part of the system falls idle when programs or system tasks are executed that cannot be divided into units that can be processed in parallel.

SIMD multiprocessing finds wide use in certain domains such as computer simulation, but is of little use in general-purpose desktop and business computing environments.

2. MISD multiprocessing

MISD multiprocessing offers mainly the advantage of redundancy, since multiple processing units perform the same tasks on the same data, reducing the chances of incorrect results if one of the units fails. MISD

architectures may involve comparisons between processing units to detect failures. Apart from the redundant and fail-safe character of this type of multiprocessing, it has few advantages, and it is very expensive. It does not improve performance. It can be implemented in a way that is transparent to software. It is used in array processors and is implemented in fault tolerant machines.

3. MIMD multiprocessing

MIMD multiprocessing architecture is suitable for a wide variety of tasks in which completely independent and parallel execution of instructions touching different sets of data can be put to productive use. For this reason, and because it is easy to implement, MIMD predominates in multiprocessing.

Processing is divided into multiple threads, each with its own hardware processor state, within a single software-defined process or within multiple processes. Insofar as a system has multiple threads awaiting dispatch (either system or user threads), this architecture makes good use of hardware resources.

MIMD does raise issues of deadlock and resource contention, however, since threads may collide in their access to resources in an unpredictable way that is difficult to manage efficiently. MIMD requires special coding in the operating system of a computer but does not require application changes unless the programs themselves use multiple threads (MIMD is transparent to single-threaded programs under most operating systems, if the programs do not voluntarily relinquish control to the OS). Both system and user software may need to use software constructs such as *semaphores* (also called *locks* or *gates*) to prevent one thread from interfering with another if they should happen to cross paths in referencing the same data. This gating or locking process increases code complexity, lowers performance, and greatly increases the amount of testing required, although not usually enough to negate the advantages of multiprocessing.

Q4. Explain distributed computing.

Ans. Distributed computing is a field of computer science that studies distributed systems. A distributed system consists of multiple autonomous computers that communicate through a computer network. The computers interact with each other in order to achieve a common

goal. A computer program that runs in a distributed system is called a distributed program, and distributed programming is the process of writing such programs.

Distributed computing also refers to the use of distributed systems to solve computational problems. In distributed computing, a problem is divided into many tasks, each of which is solved by one or more computers.

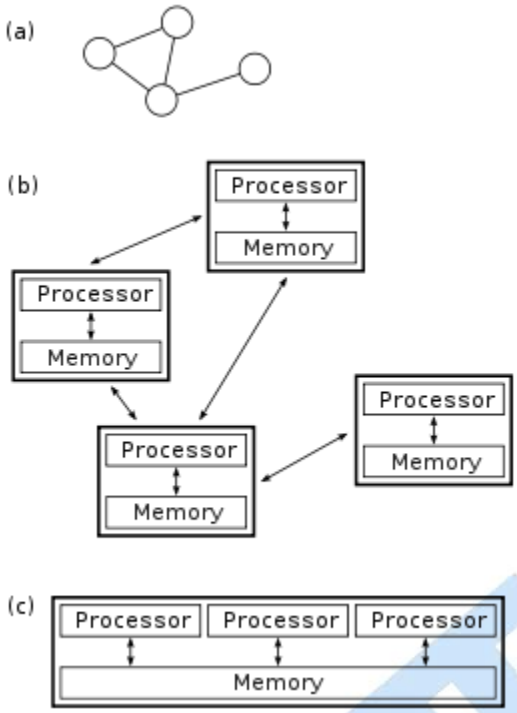
The word *distributed* in terms such as "distributed system", "distributed programming", and "distributed algorithm" originally referred to computer networks where individual computers were physically distributed within some geographical area. Distributed systems are groups of networked computers, which have the same goal for their work.

The terms "concurrent computing", "parallel computing", and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly-coupled form of distributed computing, and distributed computing may be seen as a loosely-coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria:

In parallel computing, all processors have access to a shared memory. Shared memory can be used to exchange information between processors.

In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.

The figure illustrates the difference between distributed and parallel systems. Figure (a) is a schematic view of a typical distributed system; as usual, the system is represented as a network topology in which each node is a computer and each line connecting the nodes is a communication link. Figure (b) shows the same distributed system in more detail: each computer has its own local memory, and information can be exchanged only by passing messages from one node to another by using the available communication links. Figure (c) shows a parallel system in which each processor has a direct access to a shared memory.



(a)–(b) A distributed system.
(c) A parallel system.

-

Chapter 9

Microprocessor

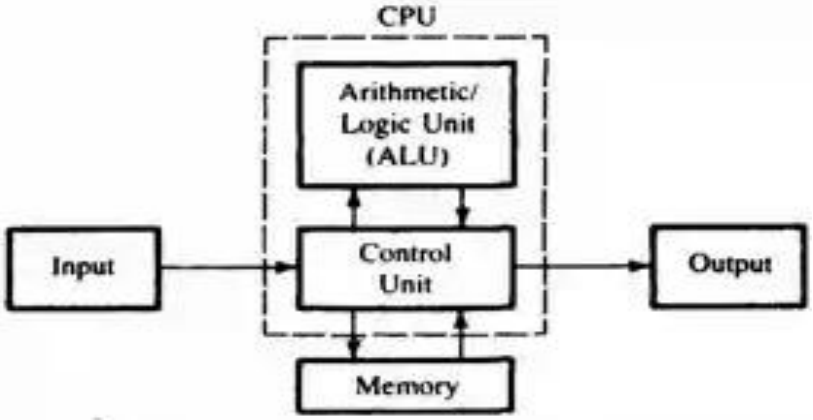
Q1. What is microprocessor?

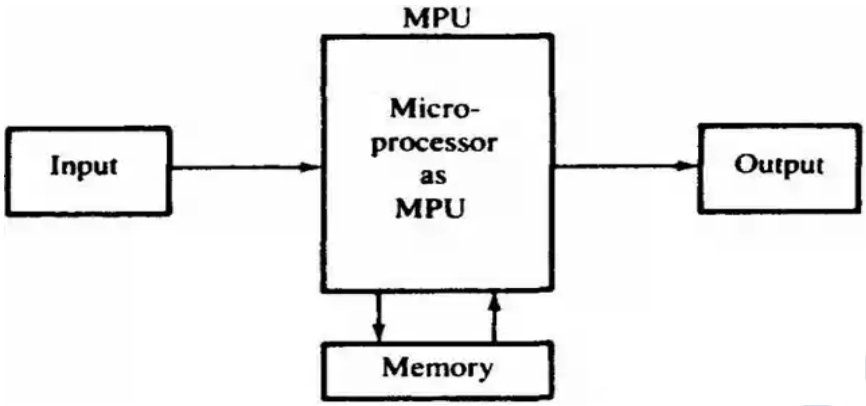
Ans. A microprocessor is a clock-driven semiconductor device consisting of electronic logic circuits manufactured by using either a large-scale integration (LSI) or very-large-scale integration (VLSI) technique.

The microprocessor is capable of performing various computing functions and making decisions to change the sequence of program execution.

The microprocessor can be divided into three segments for the sake of clarity. – They are: arithmetic/logic unit (ALU), register array, and control unit.

A comparison between a microprocessor, and a computer is shown below:





Arithmetic/Logic Unit: This is the area of the microprocessor where various computing functions are performed on data. The ALU unit performs such arithmetic operations as addition and subtraction, and such logic operations as AND, OR, and exclusive OR.

Register Array: This area of the microprocessor consists of various registers identified by letters such as B, C, D, E, H, and L. These registers are primarily used to store data temporarily during the execution of a program and are accessible to the user through instructions.

Control Unit: The control unit provides the necessary timing and control signals to all the operations in the microcomputer. It controls the flow of data between the microprocessor and memory and peripherals.

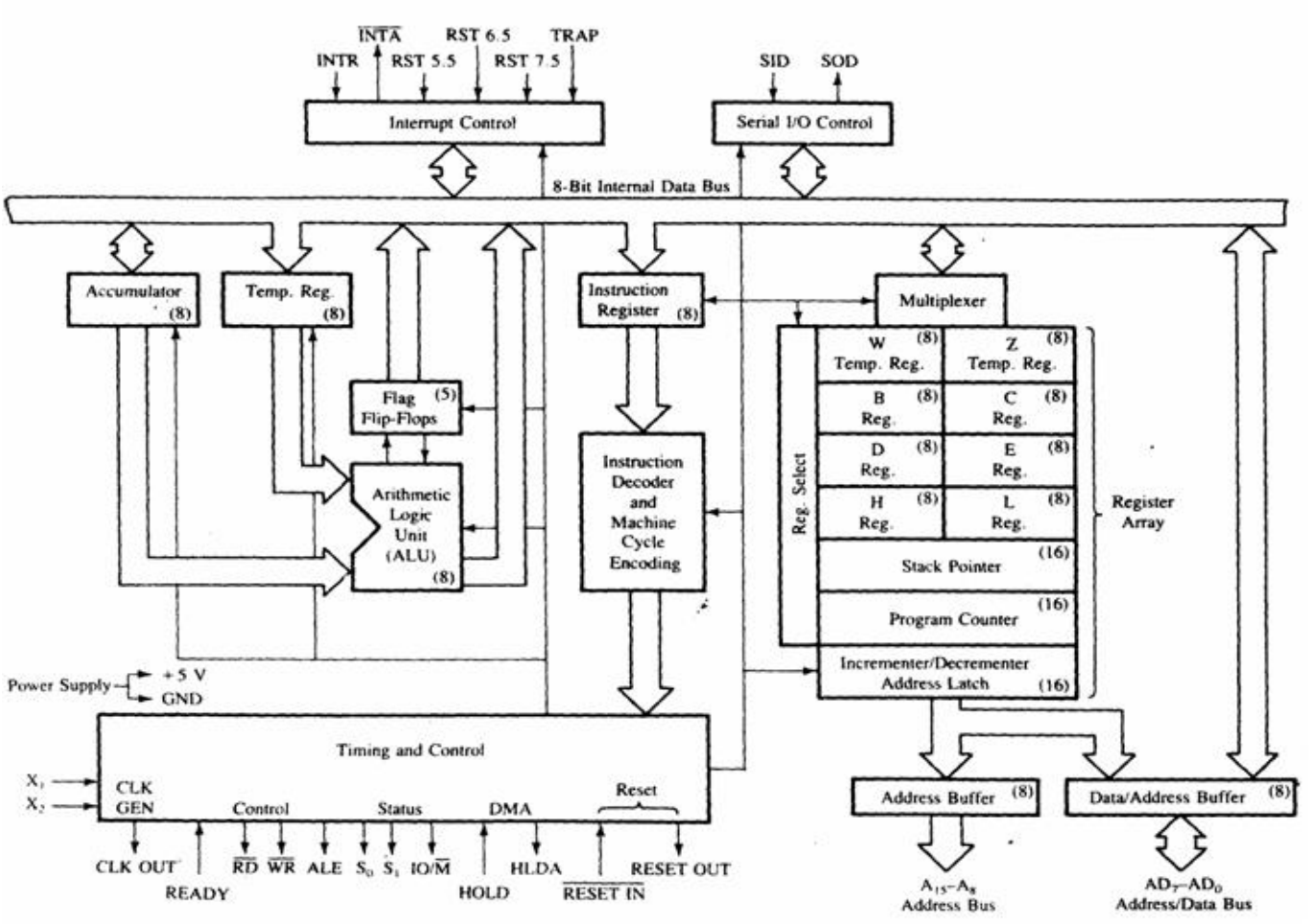
Memory: Memory stores such binary information as instructions and data, and provides that information to the microprocessor whenever necessary.

I/O (Input/output): It communicates with the outside world. I/O includes two types of devices: input and output; these I/O devices are also known as peripherals.

System Bus: The system bus is a communication path between the microprocessor and peripherals: it is nothing but a group of wires to carry bits.

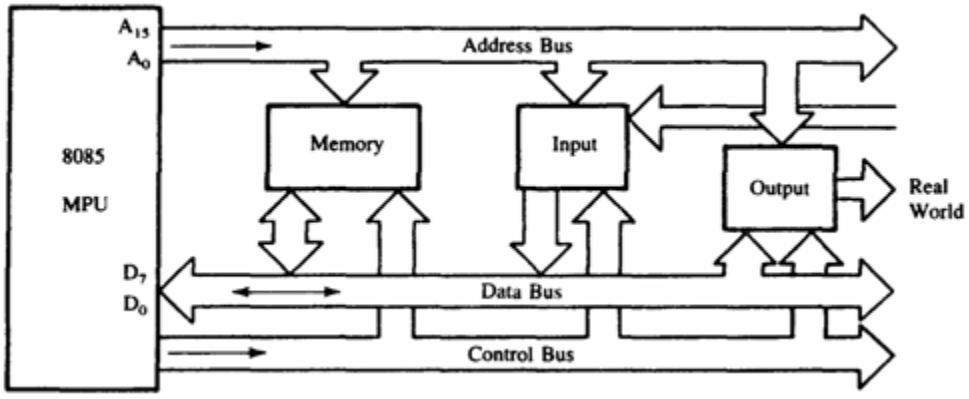
Q2. Explain the functional block diagram of 8085 microprocessor.

Ans. The functional block diagram or architecture of 8085 Microprocessor is very important as it gives the complete details about a Microprocessor. Fig. shows the Block diagram of a Microprocessor.



Q3. Explain 8085 bus structure.

Ans. 8085 Bus Structure:



Address Bus:
The address bus is a group of 16 lines generally identified as A0 to A15.
The address bus is unidirectional: bits flow in one direction-from the MPU to peripheral devices.
The MPU uses the address bus to perform the first function: identifying a peripheral or a memory location.

Data Bus:
The data bus is a group of eight lines used for data flow.
These lines are bi-directional - data flow in both directions between the MPU and memory and peripheral devices.
The MPU uses the data bus to perform the second function: transferring binary information.
The eight data lines enable the MPU to manipulate 8-bit data ranging from 00 to FF (28 = 256 numbers).
The largest number that can appear on the data bus is 11111111.

Control Bus:
The control bus carries synchronization signals and providing timing signals.
The MPU generates specific control signals for every operation it performs. These signals are used to identify a device type with which the MPU wants to communicate.

Q4. What kind of instruction are available in 8085 instruction set how will you classify . give e.g from each group.

Ans. Data Transfer Group:
The data transfer instructions move data between registers or between memory and registers.

MOV Move
MVI Move Immediate

Arithmetic Group:
The arithmetic instructions add, subtract, increment, or decrement data in registers or memory.

ADD Add to Accumulator

ADI Add Immediate Data to Accumulator

Logical Group:
This group performs logical (Boolean) operations on data in registers and memory and on condition flags.

OR Logical OR with Accumulator Using Immediate Data
XRA Exclusive Logical OR with Accumulator
XRI Exclusive OR Using Immediate Data

The Compare instructions compare the content of an 8-bit value with the contents of the accumulator;

CMP Compare
CPI Compare Using Immediate Data

The rotate instructions shift the contents of the accumulator one bit position to the left or right:

RLC Rotate Accumulator Left
RRC Rotate Accumulator Right
RAL Rotate Left Through Carry
RAR Rotate Right Through Carry

Complement and carry flag instructions:

CMA Complement Accumulator
CMC Complement Carry Flag
STC Set Carry Flag

Branch Group:
The branching instructions alter normal sequential program flow, either unconditionally or conditionally. The unconditional branching instructions are as follows:

JMP Jump
CALL Call

RET Return

Stack I/O, and Machine Control Instructions:
The following instructions affect the Stack and/or Stack Pointer:

PUSH Push Two bytes of Data onto the Stack
POP Pop Two Bytes of Data off the Stack

The I/0 instructions are as follows:

IN Initiate Input Operation
OUT Initiate Output Operation

The Machine Control instructions are as follows:
EI Enable Interrupt System
DI Disable Interrupt System
HLT Halt
NOP No Operation

Chapter 10

Microprocessor program

Q1. Add the contents of memory locations 4000H and 4001H and place the result in memory location 4002H.

Ans. Sample problem

(4000H) = 14H
(4001H) = 89H
Result = 14H + 89H = 9DH

Source program

LXI H 4000H : HL points 4000H
MOV A, M : Get first operand
INX H : HL points 4001H
ADD M : Add second operand
INX H : HL points 4002H
MOV M, A : Store result at 4002H
HLT : Terminate program execution

Q2. Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit number in memory locations 4000H and 4001H. The most significant eight bits of the two numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.

Ans. Sample problem:

(4000H) = 19H
(400IH) = 6AH
(4004H) = I5H (4003H) = 5CH
Result = 6A19H - 5C15H = OE04H
(4004H) = 04H
(4005H) = OEH

Source program:
LHLD 4000H : Get first 16-bit number in HL
XCHG : Save first 16-bit number in DE
LHLD 4002H : Get second 16-bit number in HL
MOV A, E : Get lower byte of the first number
SUB L : Subtract lower byte of the second number
MOV L, A : Store the result in L register
MOV A, D : Get higher byte of the first number
SBB H : Subtract higher byte of second number with borrow
MOV H, A : Store l6-bit result in memory locations 4004H and 4005H.
SHLD 4004H : Store l6-bit result in memory locations 4004H and 4005H.
HLT : Terminate program execution.

Q3. Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.

Ans. Sample problem:
(4000H) = 51H
(4001H) = 19H
Result = 51H - 19H = 38H
Source program:
LXI H, 4000H : HL points 4000H
MOV A, M : Get first operand
INX H : HL points 4001H
SUB M : Subtract second operand
INX H : HL points 4002H
MOV M, A : Store result at 4002H.
HLT : Terminate program execution

Q4. Add two 16-bit numbers

Ans. Sample problem:
(4000H) = 15H
(4001H) = 1CH
(4002H) = B7H
(4003H) = 5AH
Result = 1C15 + 5AB7H = 76CCH

(4004H) = CCH
(4005H) = 76H
Source Program :
LHLD 4000H : Get first I6-bit number in HL
XCHG : Save first I6-bit number in DE
LHLD 4002H : Get second I6-bit number in HL
MOV A, E : Get lower byte of the first number
ADD L : Add lower byte of the second number
MOV L, A : Store result in L register
MOV A, D : Get higher byte of the first number
ADC H : Add higher byte of the second number with CARRY
MOV H, A : Store result in H register
SHLD 4004H : Store I6-bit result in memory locations 4004H and 4005H.
HLT : Terminate program execution

Q5. Add the contents of memory locations 40001H and 4001H and place the result in the memory locations 4002Hand 4003H.

Ans. Sample problem:
(4000H) = 7FH
(400lH) = 89H
Result = 7FH + 89H = lO8H
(4002H) = 08H
(4003H) = 0lH
Source program:
LXI H, 4000H :HL Points 4000H
MOV A, M :Get first operand
INX H :HL Points 4001H
ADD M :Add second operand
INX H :HL Points 4002H
MOV M, A :Store the lower byte of result at 4002H
MVIA, 00 :Initialize higher byte result with 00H
ADC A :Add carry in the high byte result
INX H :HL Points 4003H
MOV M, A :Store the higher byte of result at 4003H
HLT :Terminate program execution

Q6. Multiply two 8-bit numbers stored in memory locations 2200H and 2201H by repetitive addition and store the result in memory locations 2300H and 2301H.

Ans. Sample problem

(2200H) = 03H
(2201H) = B2H
Result = B2H + B2H + B2H = 216H

(2300H) = 16H
(2301H) = 02H

Source program :

```
LDA 2200H
MOV E, A
MVI D, 00 : Get the first number in DE register pair
LDA 2201H
MOV C, A : Initialize counter
LX I H, 0000 H : Result = 0
BACK: DAD D : Result = result + first number
DCR C : Decrement count
JNZ BACK : If count  0 repeat
SHLD 2300H : Store result
HLT : Terminate program execution
```

Multiple Choice Questions

1. A byte corresponds to:
   (a) 4 bits (b) 8 bits
   (c) 16 bits (d) 32 bits ( )

2. The system bus is made up of:
   (a) Data bus
   (b) Data bus and address bus
   (c) Data bus and control bus
   (d) Data bus, control bus and address bus ( )

3. Cache memory enhances:
   (a) Memory capacity
   (b) Memory access time
   (c) Secondary storage capacity
   (d) Secondary storage access time ( )

4. An XOR operation is realized using the following expression:
   (a) F = A'B' = AB' (b) F = A'B' + AB
   (c) F = (A'+B'). (A+B') (d) F = (A'+B').(A+B) ( )

5. VGA stands for :
   (a) Video Graphics Array (b) Virtual Graphics Adaptor
   (c) Video Graphics Adaptor (d) Virtual Graphics Array ( )

6. The time required by the read/write head to reach the desired track in Magnetic disk is known as:
   (a) Seek Time (b) Search Time
   (c) Latency Time (d) Track Search Time ( )

7. Cache memory is implemented with using of:
   (a) DRAM
   (b) EEPROM
   (c) EPROM
   (d) None of the above ( )

8. The memory which is programmed at the time it manufactured:
   (a) ROM
   (b) RAM
   (c) PROM
   (d) All of the above ( )

9. Which of the following is/are basic computer registers:
   (a) PC
   (b) DR
   (c) ACC
   (d) All of the above ( )

10. A CPU consist of:
   (a) *ALU Only*
   (b) Control Unit Only
   (c) ALU, Control Unit and Registers
   (d) None of the above ( )

11. The process of accessing informing on a CD-ROM is:
   (a) Random
   (b) Sequential
   (c) Semi-random
   (d) None of the above ( )

12. SIMM sands for:
   (a) Single Instruction Memory Management
   (b) Single In-line Memory Module
   (c) Single Instruction Memory Module
   (d) Single In-line Memory Management ( )

13. The number or memory locations that a CPU with a 16-bit program counter can address:
   (a) 16 k (b) 256 k
   (c) 64 k (d) 32 k ( )

14. Cache memory is:

(a) Temporary memory
(b) Primary
(c) High speed memory
(d) All of the above ( )

15. Compare with secondary storage, primary storage is:
(a) Slow and inexpensive (b) Fast and inexpensive
(c) Fast and expensive (d) Slow and expensive ( )

16. RAM is:
(a) Ream Only Memory
(b) Write Only Memory
(c) Read/Write Memory
(d) None of the above ( )

17. One kilo byte is:
(a) 1024 bits (b) 1064 bits
(c) 1026 bits (d) 1000 bits ( )

18. A 4 – bit ring counter is initially loaded with:
(a) 0000 (b) 0001
(c) 1110 (d) 1111 ( )

19. How many full addresses are needed to add two 4-bit numbers?
(a) 8 (b) 2
(c) 4 (d) 16 ( )

20. An asynchronous counter can be designed using:
(a) 4 flip - flops (b) 5 flip – flops
(c) 10 flip flops (d) 6 flip – flops ( )

21. Which of the following is the internal memory of the computer?
(a) CPU registers
(b) Cache memory
(c) Main memory
(d) All of the above ( )

22. What does CISC stand for?
    (a) Complex instruction set-computer
    (b) Counter instruction set computer
    (c) Complex instruction set counter
    (d) Counter instruction set complex ( )

23. A multiplexer is a circuit with:
    (a) Many inputs and a single output
    (b) Many inputs and many outputs
    (c) One input and many outputs
    (d) None of the above ( )

24. Performance of a cache memory is measured in terms of:
    (a) Hit ratio (b) Miss ration
    (c) Direct ratio (d) Indirect ratio ( )

25. D type of flip – flop can be designed with:
    (a) S-R Flip flop
    (b) $J-K\ flip-flop$
    (c) $T\ flip-flop$
    (d) All of the above ( )

26. The poritiy bit is used for:
    (a) Error checking (b) Coding
    (c) Redudancy (d) Indexing ( )

27. Direct address are as same as:
    (a) Effective address
    (b) Address of operands
    (c) Both (a) and (b)
    (d) One of the above ( )

28. CPU of a computer system does not contain:
    (a) Main storage
    (b) *Arithmetic unit*
    (c) Special register group
    (d) None of the above ( )

29. Which of the following is the slowest in accessing data?
(a) Zip disk (b) *Hard disk*
(c) Floppy disk (d) Magnetic tape ( )

30. Which Gate is known as Universal Gate?
(a) NOT gate (b) AND gate
(c) NAND gate (d) XOR gate ( )

31. Data transfer rate in a modem is measured in:
(a) Bits per sound (b) *bits per minute*
(c) Cycles per second (d) Bits per hour ( )

32. The process of preparing a disk with tracks and sector is called:
(a) Formatting (b) Caching
(c) Surging (d) Crushing ( )

33. A storage device can be:
(a) Direct access
(b) Sequential access
(c) Both (a) and (b)
(d) None of the above ( )

34. The OCR stands for:
(a) Outsized character reader
(b) Optical character recognition
(c) Operational character reader
(d) Only character reader ( )

35. The NOR gate is logically equivalent to an OR gate followed by an:
(a) AND (b) XOR
(c) NOR (d) XNOR ( )

36. Storage device can be:
(a) Sequential
(b) Direct access
(c) Both (a) and (b)

(d) None of the above ( )

37. Effective address in addressing mode is:
(a) Address of an instruction
(b) *Address of the operation code*
(c) Address of the operand to fetch
(d) Contents of program counter ( )

38. The number of select input lines in 1 to 16 demultiplexer is:
(a) 1 (b) 4
(b) 8 (d) 16 ( )

39. The distance between the phosphor dots that make up a single pixel is called:
(a) Resolution (b) Dot pitch
(c) Dot distance (d) Dot rate ( )

40. Which of the following bus is bi- directional?
(a) Address bus
(b) *data bus*
(c) Control bus
(d) All of the above ( )

Answer Key

1. (b)	2. (d)	3. (a)	4. (d)	5. (c)	6. (a)	7. (b)	8. (c)	9. (d)	10. (c)
11. (a)	12. (b)	13. (c)	14. (d)	15. (c)	16. (c)	17. (a)	18. (a)	19. (c)	20. (a)
21. (d)	22. (a)	23. (a)	24. (a)	25. (a)	26. (a)	27. (c)	28. (a)	29. (d)	30. (c)
31. (a)	32. (a)	33. (c)	34. (b)	35. (c)	36. (c)	37. (c)	38. (b)	39. (c)	40. (b)

Set 2

1. The capacity of CD-ROM is around:
   (a) 100 MB (b) 650 MB
   (c) 1 GB (d) 4 GB ( )

2. The faster and most expensive type of storage device is:
   (a) Electronic disk (b) Register
   (c) Cache (d) Magnetic Tape ( )

3. Which is not an output device?
   (a) Printer (b) Monitor
   (c) Scanner (d) Plotter ( )

4. The number of bits in a nibble are:
   (a) 16 (b) 5
   (c) 4 (d) 8 ( )

5. The device used in a data communication network to perform the conversion between analog and digital signal, is called a :
   (a) Front end processor (b) Modem
   (c) Decoder (d) Multiplexor ( )

6. The smallest addressable portion of disk is called a:
   (a) Sector (b) Track
   (c) Inode (d) Bit ( )

7. Which device has one input and many outputs?
   (a) Flip-flop (b) Multiplexor
   (c) Demultiplexer (d) Counter ( )

8. What does RAM and DRAM stand for?
   (a) Remote Access Memory, Dynamic Remote Access Memory
   (b) Random Access Memory, Dynamic Random Access Memory
   (c) Ramote Access Memory, Dependent Random Access Memory
   (d) Random Access Memory, Dependent Random Memory ( )

9. The out put of ……gate is 1, when all of its inputs are 1
   (a) NOR (b) XOR
   (c) *AND* (d) NOT ( )

10. Which part of the computer performs arithmetic calculations?
   (a) ALU
   (b) Registers
   (c) Logic Bus
   (d) None of the above ( )

11. The Parity bit is:
   (a) always 1
   (b) always 0
   (c) 1 or 0
   (d) None of the above ( )

12. Which monitor has a maximum resolution of 1024 x 768?
   (a) SXGA (b) XGA
   (c) SVGA (d) VXGA ( )

13. Magnetic tape is a:
   (a) Serial access medium
   (b) Random access median
   (c) Parallel access medium
   (d) None of the above ( )

14. What does DMA stand for?
   (a) Direct Memory Access
   (b) Direct Memory Accelerator
   (c) Directional Memory Access
   (d) Distributed Multiprogramming Assistant ( )

15. The system is made up of:
   (a) Data bus
   (b) Data bus and address bus
   (c) Data bus and control bus

(d) Data bus, control bus and address bus ( )

16. What is RISC?
(a) Remodeled Interface System Computer
(b) Remote Internet Secured Connection
(c) Runtime Instruction Set Compiler
(d) Reduced Instruction Set Computer ( )

17. Which is a non- volatile memory?
(a) RAM
(b) ROM
(c) Both (a) and (b)
(d) None of the above ( )

18. The program counter:
(a) Stores the address of the instruction that is currently being executed
(b) Stores next instruction to be executed
(c) Stores the address of next instruction to be executed
(d) Stores the instruction that is currently being executed ( )

19. A process is a:
(a) Single thread of execution (b) Program in execution
(c) Program in memory (d) Task ( )

20 What do you need for an Inkjet Printer?
(a) A cartridge
(b) A drum
(c) A ribbon
(d) None of the above ( )

21. In which type of flip-flop the indeterminate condition of the SR flip -flop (When S=R=1 is defined) ?
(a) Edge Triggered Flip Flop (b) J K Flip-Flop
(c) D Flip-Flop (d) T Flip-Flop ( )

22. A binary cell capable of storing one bit of information is called is a:

(a) Flip-Flop (b) Latch
(c) Combinational Circuit (d) Clock ( )

23. The processor …….and I/O devices are interconnected by means of common bus.
(a) Cache memory (b) Auxiliary memory
(c) Virtual memory (d) Main memory ( )

24. A virtual memory is:
(a) A form of ROM
(b) Related to virtual reality
(c) A form of RAM
(d) None of the above ( )

25. Vacuum tube was used in the ……generation of computers.
(a) I (b) II
(c) III (d) IV ( )

26. CISC machines:
(a) Have fewer instruction than RISC machines
(b) Use more RAM than RISC machines
(c) Have medium clock speed
(d) Use variable size instruction ( )

27. Pick and odd one out:
(a) keyboard (b) Touch Screen
(c) Scanner (d) Plotter ( )

28. Interrupt signals generated by a printer are called:
(a) Internal interrupt
(b) External interrupt
(c) Software interrupt
(d) None of the above ( )

29. ……..interrupt is also called trap.
(a) Internal
(b) External

(c) Both A and B
(d) None of the above ( )

30. What is the name of logic circuit which can add two binary digits?
(a) Full adder
(b) Parallel adder
(c) Half adder
(d) None of the above ( )

31. Which of the following is a bus architecture?
(a) ISA
(b) AGP
(c) MCA
(d) All of the above ( )

32. Most computer come with:
(a) Serial Port (b) Parallel port
(c) Both Serial and Parallel Port (d) SCSI port
( )

33. Which memory allows the address space to be larger than the memory space?
(a) Cache memory (b) Main memory
(c) Virtual memory (d) Auxiliary memory ( )

34. Cache memory is used in computer system to:
(a) Ensure fast booting (b) Replace static memory
(c) Replace hard disk (d) Speed up memory access ( )

35. Bit is an abbreviation of:
(a) Binary digits
(b) Birla Institute of Technology
(c) British Institute of Technology
(d) None of the above ( )

36. In a JK flip- flop the function k = j will results into:

(a) T Flip – Flop (b) S R Flip Flop
(c) D Flip Flop (d) Master Slave flip flop ( )

37. Effective address in addressing mode is:
(a) Address of an Instruction
(b) Address of the Operation Code
(c) Address of the Opened to Fetch
(d) Contents to PC ( )

38. Which is the largest unit of storage among the following:
(a) Tera byte (b) Kilo byte
(c) Mega byte (d) Giga byte ( )

39. The minimum number of bits required to represent 128 are:
(a) 8 (b) 5
(c) 16 (d) 7 ( )

40. The output of combinational circuit depends in:
(a) Present input
(b) Previous input
(c) Both present and previous inputs
(d) None of the above ( )

Answer Key

1. (b)	2. (c)	3. (c)	4. (c)	5. (b)	6. (a)	7. (c)	8. (b)	9. (c)	10. (a)
11. (c)	12. (c)	13. (a)	14. (a)	15. (d)	16. (d)	17. (b)	18. (c)	19. (b)	20. (a)
21. (a)	22. (a)	23. (d)	24. (c)	25. (a)	26. (a)	27. (d)	28. (b)	29. (a)	30. (c)
31. (a)	32. (c)	33. (c)	34. (d)	35. (a)	36. (c)	37. (c)	38. (a)	39. (d)	40. (c)

Set 3

1. The storage capacity of 3.5 inch floppy disk is:
   (a) 1.44 kb (b) 1.44 mb
   (c) 650 mb (d) 1.2 mb ( )

2. Which is a sequential storage device:
   (a) Hark Disk (b) Magnetic disk
   (c) magnetic tape (d) none ( )

3. Which is fastest memory:
   (a) Virtual memory (b) Ram
   (c) ROM (d) Cache Memory ( )

4. The input unit of a computer system:
   (a) Feeds the data in CPU
   (b) Retrieve the data from CPU
   (c) Directs all other units
   (d) All of the above ( )

5. An address is the number used by the CPU to specify :
   (a) A location in the memory (b) A location in the flags
   (c) A location in accumulator (d) A location in stack pointer ( )

6. Which of the flip-flops suffer from the race condition problem:
   (a) D flip-flop (b) T flip-flop
   (c) J K flip-flop (d) RS flip-flop ( )

7. A 16 to 1 multiplexer requires how may control signals:
   (a) 3 (b) 4
   (c) 2 (d) 5 ( )

8. The way the operands are chosen during program execution is dependent on the…..of the instruction.
   (a) Instruction format (b) Addressing mode
   (c) Data format (d) None ( )

9. The minimum number of bits required to represent 49 is:
   (a) 8 (b) 5
   (c) 6 (d) 7 ( )

10. Pick odd one out:
    (a) FPM (b) EDO
    (c) SD RAM (d) RS 232 ( )

11. A modem can be of type:
    (a) Fax
    (b) Data
    (c) Voice
    (d) All of the above ( )

12. Which is the smallest unit of storage:
    (a) Tracks (b) Cylinders
    (c) Sector (d) None ( )

13. Which is not an input device:
    (a) Scanner (b) Trackball
    (c) Mouse (d) Plotter ( )

14. An electric circuit that has two stable states is called:
    (a) Encoder (b) Flip-flop
    (c) Decoder (d) Multiplexer ( )

15. RAM is:
    (a) Read only memory
    (b) Write only memory
    (c) Read/write memory
    (d) None of the above ( )

16. Which is called a universal gate:
    (a) XOR (b) XNOR
    (c) NAND (d) OR ( )

17. In which mode the operands are specified implicity in the definition of the instruction:
    (a) Implied mode (b) Register Mode
    (c) Relative Address mode (d) None ( )

18. The ALU of a computer normally contains a number of high speed storage elements called:
    (a) Semiconductor memory (b) Register

(c) Hard Disk (d) Magnetic Disk ( )

19. The computer use addressing mode for the purpose of:
(a) the give programming versatility
(b) to reduce the number of bits in the addressing field
(c) both A & B
(d) None of these ( )

20. Which one use fixed length instruction format:
(a) RISC (b) CISC
(c) Both A & B (d) None ( )

21. Which port is more popular in these days.
(a) Serial (b) Parallel
(c) USB (d) AGP ( )

22. Which type of interrupts normally occur due to I/O divices:
(a) Software Interrupts (b) External Interrupts
(c) Internal Interrupts (d) None ( )

23. Interrupts arise from illegal or erroneous use of an instruction or data:
(a) Internal (b) External
(c) Hardware (d) None ( )

24. A large number of instructions are used in:
(a) CISC (b) RISC
(c) Both (a) and (b) (d) None ( )

25. The purpose of ……is to speed up the computer processing capability and increase its throughput.
(a) Parallel processing (b) Interrupts
(c) RISC (d) None ( )

26. The parity bit is used for:
(a) Error checking (b) Indexing
(c) Coding (d) Controlling ( )

27. Pick odd one put:
(a) VGA (b) SVGA
(c) EISA (d) SXGA ( )

28 Concentric circles in a bard disk are called:
(a) Sector (b) Cylinder
(c) Read/write head (d) Tracks ( )

29. Which one term is not associated with a hard disk:
(a) IDE (b) Reels
(c) EIDE (c) SCSI ( )

30. Which is a bidirectional bus:
(a) Data (b) Address
(c) Both A & B (d) None ( )

31. Which one has the highest storage capacity:
(a) Floppy disk (b) CD
(c) DVD (d) None ( )

32. A ……supplied by one of the units to indicate to the other unit when the transfer has to occur:
(a) Strobe (b) Interrupt
(c) DMA (d) None ( )

33. The……..technique improves the sped of transfer by managing memory buses directly:
(a) RISC (b) DAT
(c) DMA (d) None ( )

34. Which flip – flop have a toggle operation?
(a) RS (b) JK
(c) D (d) T ( )

35. Which combinational logic circuit is used to convert a decimal number to its equivalent binary number?
(a) Multiplexor (b) Decoder
(c) Encoder (d) None ( )

36. The transistor was used ……in the generation of computer:
(a) I (b) II
(c) III (d) IV ( )

37. Which one gives an illusion of unlimited memory?
(a) Virtual (b) Cache

(c) RAM (d) ROM ( )

38. Which memory is used to increase the speed of processing by making current programs and data available to the CPU at a rapid rate?
(a) RAM (b) ROM
(c) DMA (d) Cache ( )

39. A multiplexer is a circuit with
(a) Many inputs but only one output
(b) Many inputs and many outputs
(c) One input and many outputs
(d) None of the above ( )

40 Address of next instruction to be exacted is available in:
(a) Program counter
(b) Index register
(c) Instruction register
(d) None of the above ( )

Answer Key

1. (b)	2. (c)	3. (d)	4. (a)	5. (a)	6. (b)	7. (b)	8. (b)	9. (c)	10. (d)
11. (d)	12. (c)	13. (d)	14. (b)	15. (c)	16. (c)	17. (a)	18. (b)	19. (c)	20. (a)
21. (c)	22. (b)	23. (a)	24. (a)	25. (a)	26. (a)	27. (c)	28. (d)	29. (a)	30. (c)
31. (c)	32. (a)	33. (c)	34. (d)	35. (c)	36. (b)	37. (a)	38. (d)	39. (a)	40. (a)

Set 3

1. A byte is:
   (a) a group of 2 bits (b) a groups of 4 bits
   (c) a group of 8 bits (d) a group of 16 bits ( )

2. An address is the number used by the CPU to specify:
   (a) A location in the memory (b) a location in the flags
   (c) a location in accumulator (d) a location in stack pointer ( )

3. Which of the following is the internal memory of the computer?
   (a) CPU register
   (b) cache
   (c) Main memory
   (d) All of the above ( )

4. The input unit of a computer :
   (a) Feeds the data in CPU
   (b) retrieve the data from CPU
   (c) Directs all other units
   (d) All of the above ( )

5. The control units of a computer :
   (a) Performs arithmetic logical operations on the data
   (b) Controls the operations of output devices
   (c) Is a device for manually operating the computer
   (d) Directs the other units of the computer ( )

6. Dump means:
   (a) Erasing used data
   (b) Storing used data in pushdown stack
   (c) Copying data from internal stage to external stage
   (d) None of the above ( )

7. The minimum number of bits required to represent 34 is:
   (a) 8 (b) 5
   (c) 6 (d) 7 ( )

8. EEPROM stands for:
   (a) Electrical and Electronic Programmable ROM

(b) Electronically Erasable Programmable ROM
(c) Electrically Erasable Programmable ROM
(d) Electronically Equipped Programmable ROM ( )

9. What does SICS stands for:
(a) Complex Instruction Set Computer
(b) Counter Instruction Set Computer
(c) Complex Instruction Set Counter
(d) Counter Instruction Set Complex ( )

10. The following is not an input device:
(a) Data Gloves (b) Scanner
(c) Bar code reader (d) Plotter ( )

11. CRT stands for:
(a) Crystal Ray Tube (b) Cathode Ray Tube
(c) Cathode Ray Terminal (d) Computer Ray Terminal ( )

12. A multiplexer is a circuit with:
(a) Many inputs but only one output
(b) Many inputs and many outputs
(c) One input and many outputs
(d) None of the above ( )

13. A CPU consists of:
(a) ALU only
(b) Control unit only
(c) ALU, control units and register
(d) None of the above ( )

14. Performance of a cache memory is measured in terms of:
(a) Hit ratio (b) Cache ratio
(c) Direct ratio (d) Indirect ratio ( )

15. An electronic circuit that has two stable states is called:
(a) Encoder (b) Flip flop
(c) Decoder (d) Multiplexer ( )

16. An XOR operation is realized using the following expression:
(a) F = A'B+AB (b) F = A'B + AB
(c) F = (A' +B). (A+B') (d) F = (A'+B). (A+B) ( )

17. Effective address in addressing mode is:
    (a) Address of an instruction (b) Address of the operation code
    (c) Address of the operand to fetch (d) Content of PC ( )

18. A D flip flop can be designed with:
    (a) SR flip-flop
    (b) JK flip-flop
    (c) MS flip-flop
    (d) All of the above ( )

19. Sum output of half adder with A and B input is expressed by:
    (a) F (sum) = AB (b) F (sum) = A+B
    (c) F (sum) = AB + A'B' (d) F (sum) = A'B + AB' ( )

20. Which of the following bus is bi-directional?
    (a) Address bus
    (b) Data bus
    (c) Control bus
    (d) All of the above ( )

21. The parity bit is used for:
    (a) Error Checking (b) Indexing
    (c) Coding (d) Controlling ( )

22. Which of the following is a unit of measurement with computer system?
    (a) Byte
    (b) Megabyte
    (c) Kilobyte
    (d) All of the above ( )

23. RAM is:
    (a) Read only memory
    (b) Write only memory
    (c) Read/write memory
    (d) None of the above ( )

24. Direct address are same as:
    (a) Effective address
    (b) Address of operands
    (c) Both A & B

(d) None of the above ( )

25. What is the name of logic circuit which can add two binary digits?
(a) Full adder
(b) Parallel Adder
(c) Half Adder
(d) None of the above ( )

26. Any combination logic circuit can be implemented by using the following universal building block :
(a) AND (b) NAND
(c) OR (d) XOR ( )

27. Address of next instruction to be executed is available in:
(a) Program counter
(b) Address Register
(c) Instruction Register
(d) None of the above ( )

28. Accumulator of the basic computer is:
(a) Instruction Register
(b) Address Register
(c) Data register
(d) None of the above ( )

29. Which of the following is not output device of a computer?
(a) Printer (b) Keyboard
(c) VDU (d) CRT screen ( )

30. Which is secondary memory device?
(a) CPU (b) ALU
(c) Floppy disk (d) Mouse ( )

31. CPU of a computer system does not contain:
(a) main storage
(b) Arithmetic unit
(c) Special register group
(d) None of the above ( )

32. American Standard Code for Information Interchange (ASCII) employ a code character set consisting of :
(a) 7 bits (b) 7 bits with parity check
(c) 8 bits (d) 8 bits with parity check ( )

33. One of the following is a direct entry input device. :
(a) key-to-diskette (b) Punched
(c) Computer Terminal (d) Mouse ( )

34. One of the following is not found on the motherboard of a personal computer:
(a) Direct memory Access (DMA) controller
(b) Programmable timer
(c) Interrupt controller
(d) Video display adapter ( )

35. A CPU generally contain:
(a) Register and ALU
(b) A control and timing section
(c) Instruction and timing section
(d) All of the above ( )

36. Which of the following computers is least powerful?
(a) Minicomputer (b) Microcomputer
(c) Mainframe Computer (d) Supercomputer ( )

37. Which of the following storage device can be carried around?
(a) Floppy disks (b) Main memory
(c) Registers (d) Core memory ( )

38. Which of the following is responsible for coordinating various operations using timing signals?
(a) Arithmetic logic unit (b) Control unit
(c) Memory unit (d) Input/Output unit ( )

39. The ALU of a computer normally contains a number of high speed storage elements called:
(a) Semiconductor Memory (b) Register
(c) Hard Disk (d) Magnetic Disk ( )

40. Computer peripheral is:
(a) A computer device which is not connected to CPU

(b) A device which is connected to CPU
(c) A device for manually operating the computer
(d) None of the above ( )

Answer Key

1. (c)	2. (a)	3. (c)	4. (a)	5. (d)	6. (b)	7. (c)	8. (c)	9. (a)	10. (d)
11. (b)	12. (a)	13. (c)	14. (a)	15. (b)	16. (c)	17. (c)	18. (a)	19. (d)	20. (b)
21. (a)	22. (d)	23. (c)	24. (a)	25. (c)	26. (b)	27. (a)	28. (a)	29. (b)	30. (c)
31. (c)	32. (a)	33. (c)	34. (d)	35. (c)	36. (b)	37. (a)	38. (d)	39. (a)	40. (a)

Set 5

1. Cache memory is:
   (a) Temporary and costly
   (b) Primary
   (c) High speed memory
   (d) All of the above ( )

2. Leser printer is:
   (a) Non-impact Line Printer (b) Non-impact Page Printer
   (c) Impact Line Printer (d) Non-impact Character Printer ( )

3. The major components of a computer are:
   (a) Memory
   (b) CPU
   (c) I/O line printer
   (d) All of the above ( )

4. Registers are part of:
   (a) Control unit and memory (b) Addresses and control unit
   (c) Address and ALU (d) Control unit and ALU( )

5. Instruction is used to store the contents of accumulator into the memory word specified by the effective address:
   (a) LDA (b) BUN
   (c) STA (d) BSA ( )

6. Computer are normally stored in consecutive memory locations and are executed sequentially one at a time:
   (a) Program
   (b) Instruction
   (c) Memory
   (d) None of the above ( )

7. What does RISC stand for:
   (a) Register Instruction Set Counter
   (b) Reduced Instruction Set Counter
   (c) Reduced Instruction Set Computer
   (d) Register Instruction Set Computer ( )

8. What was the data bus width of 68000 Motorola CPU:
   (a) 8
   (b) 16
   (c) 32
   (d) None of the above ( )

9. Information is recorded on a number of concentric circles, called:
   (a) Sectors
   (b) Tracks
   (c) Cylinder
   (d) None of the above ( )

10. Personal computer use a simpler system structure, most of the machines have a card cage with a large printed circuit board at the bottom called:
   (a) CPU
   (b) Expansion slots
   (c) Motherboard
   (d) None of the above ( )

11. A computer system consists of a CPU, a memory, and one or more specialized I/O processors called:
(a) Bandwidth
(b) Data channels
(c) Interrupts
(d) None of the above ( )

12. The most common way of allowing users to point at the screen with a device called:
(a) Mouse (b) Keyboard
(c) Trackball (d) Touch screen ( )

13. A common electrical pathway between multiple devices is:
(a) Clock (b) Bus
(c) Memory (d) Modem ( )

14. Which of the following is the slowest in accessing data?
(a) Zip disk (b) Hard Disk
(c) Floppy Disk (d) Magnetic Tape ( )

15. Which is the re-usable optical disk?
(a) CD-ROM (b) WORM
(c) CD-R (d) CR-RW ( )

16. Which is the technology used in the evaluation of aptitude test?
(a) OCR (b) OMR
(c) MICR (d) MCR ( )

17. What is the measure of the package density of the pixels in a monitor?
(a) Refresh rate
(b) Resolution
(c) Pixel density
(d) None of the above ( )

18. Race (in determinate) condition occurs in the:
(a) JK flip-flop (b) D flip-flop
(c) RS flip-flop (d) T flip-flop ( )

19. Which gate is known as universal gate?
(a) Not Gate (b) AND Gate
(c) NAND Gate (d) XOR Gate ( )

20. Which kind of device allows the user the add components and capabilities to a computer system?
(a) System boards (b) Storage devices
(c) Input devices (d) Expansion slots ( )

21. Capacity of 3 ½ inch floppy disk is:
(a) 20 MB (b) 360 MB
(c) 1.44 MB (d) 1.44 GB ( )

22. Which of the following is an example of volatile memory?
(a) ROM (b) PROM
(c) RAM (d) Hard disk ( )

23. Data transfer rate in modem is measured in:
(a) Bits per second (b) Bits per minute
(c) Cycles per second (d) Bits per hour ( )

24. A computer has a 64 memory. What does letter K stand for?
(a) 1000 (b) Kilometer
(c) 1024 (d) 1064 ( )

25. The full form of MODEM is:
(a) Modifier detoxifier
(b) Modulator demodulator
(c) Both A & B
(d) None of the above ( )

26. The process of preparing a disk with tracks and sector is called :
(a) *Formatting* (b) Caching
(c) Surfing (d) Crushing ( )

27. What is the following is bus architecture?
(a) ISA
(b) AGP
(c) MCA
(d) All of the above ( )

28. Example of display adaptor card is:
(a) CGA
(b) VGA

(c) SUGA
(d) None of the above ( )

29. Storage device can be:
(a) Direct access
(b) Sequential access
(c) Both of the above
(d) None of the above ( )

30. Linkage between the CPU and the users is provided by:
(a) Peripheral devices (b) Storage
(c) control unit (d) Software ( )

31. A combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output liens:
(a) Encoder (b) Multiplayer
(c) Decoder (d) Demultiplexer ( )

32. CISC is a:
(a) Complete Instruction Set Computer
(b) Complex Instruction Set Computer
(c) Complex Inline Store Computer
(d) Complete Instruction, Store Computer ( )

33. A combinational circuit, which performs the addition of two bits, is called:
(a) Full-adder (b) Half-adder
(c) Multiplexer (d) Decoder ( )

34. SIMM is a:
(a) Single Instruction Memory Modules
(b) Single in Line Memory Modules
(c) Single Instruction Memory Manufacture
(d) Single in Line Micro programs Modules ( )

35. A memory having 2^{16} words with each word of 8 bits is referred to as:
(a) 32 K memory (b) 64 k memory
(c) 128 K memory (d) 256 K memory ( )

36. The storage magnetic tape is divided into vertical columns called:
(a) Tracks
(b) Sectors

(c) Frames
(d) None of the above ( )

37. The storage capacity of a disk system depends upon:
(a) Number of recording surfaces
(b) Number of Tracks Per Surface
(c) Number of sectors per track
(d) All of the none ( )

38. Access times for optical disk are typically in the range of:
(a) 10 to 30 milliseconds
(b) 100 to 200 milliseconds
(c) 100 to 300 milliseconds
(d) None of the above ( )

39. Which is the largest unit of storage among the following?
(a) Terabyte (b) Kilobyte
(c) Megabyte (d) Gigabyte ( )

40. Most computers come with:
(a) Serial port (b) Parallel port
(c) Both serial and parallel ports (d) SCSI ports ( )

Answer Key

1. (c)	2. (a)	3. (c)	4. (a)	5. (d)	6. (b)	7. (c)	8. (c)	9. (a)	10. (d)
11. (b)	12. (a)	13. (c)	14. (a)	15. (b)	16. (c)	17. (c)	18. (a)	19. (d)	20. (b)
21. (a)	22. (d)	23. (c)	24. (a)	25. (c)	26. (b)	27. (a)	28. (a)	29. (b)	30. (c)
31. (c)	32. (d)	33. (b)	34. (b)	35. (d)	36. (b)	37. (a)	38. (b)	39. (c)	40. (b)

Key Terms

A

Access Time	Time required to place read/write heads of the disk over a particular track & sector. Rotational delay and seek time should also be considered.
Accumulator	CPU register that has the outcome of operations and occasionally the operands
Address Bus	System bus used to move addresses in RAM or I/O device
ALU	element of the CPU that does arithmetic and logical operations
ASCII (American Standard Code for Information Interchange)	A 7-bit standard character set that signify characters inside the computer.
ASRAM (Asynchronous RAM)	Static RAM is the one for transferring data without using the system clock.
Assembler	A system program that interpret a mnemonic assembly language to low level language.
Associative Memory	Memory whose location are recognized by their contents, rather than their position

B

Binary Operator	An operator that perform on two operands.
Bit (Binary Digit)	A single memory unit that has a single binary value (0 or 1)
Boolean Algebra	Mathematics laws applied by the processor to do logical and shift operations.
Buffer	Memory locations used to hold input or output data. It is required to balance difference in speed amid the CPU and the I/O devices.
Bus	Communication path consisting of a group of lines that carry signals, addresses, or data amid PC's elements. A bus can be used by all computer elements.
Byte	Memory unit that embrace 8 bits.

C

Cache	A small, fast memory - perform like a buffer. It is used to improve performance of CPU.
Chipset	A collection of typical PC functions pooled onto one or more integrated circuit.
Clock	Square wave with equal intervals. Used to harmonize CPU process. Events typically happen at rising or falling rim of the clock.
Combinational (combinatorial) Circuit	Logic circuit who's yield is a function of its input only at any specified time. There is no storage capacity of preceding contents of the circuit.
Control Bus	System bus for transferring control signals among processor and other apparatus.
Control Unit	Part of the CPU accountable for calculating and coordinate computer functions.
CPU (Central Processing Unit)	It is accountable for performing instructions and controlling all other components.

D

Data Bus	System bus for transferring data.
Decoder	Combinational circuit for transferring input signal combination on numerous input lines into one specific 2^n-output lines.
Distributed Memory	Physical memory that is alienated into module every allocated to a processor in a multiprocessor organization.
DMA (Direct Memory Access)	I/O method that permit direct data swap between memory and I/O devices without holding the processor time. CPU only begins the I/O request and is interrupted after the transfer is complete.
DRAM (Dynamic RAM)	RAM put into practice using capacitors and that requires to be sporadically re-energized

E

EEPROM (Electronically Erasable Programmable Read-Only Memory)	usually used in BIOS chip and can be reorganized with a procedure known as **flashing** using dedicated software.
Error-Correcting Code	Code used for sending/receiving signals or characters used to automatically correct errors.
Error-Detecting Code	Code used for sending/receiving signals or characters used to automatically detect errors.
Exclusive-OR Gate (Function)	The logic - produce logical value "True" if both input values are dissimilar else produce "False".

F

Fetch Cycle	It is used for obtaining the instruction to be carried out from memory.
Fixed-Point Representation System	Representation of Real number in which the radix (decimal for radix 10) is kept in a fixed place.
Flip-Flop	A memory unit that include one binary value and in which the output signify the current state. The next state depends on current state and the input.
Floating-Point Representation System	Real number system in which the number is symbolize as two distinct parts- mantissa and exponent.

G

Gate	Combinational circuit component that make an output that look like simple Boolean function (And, OR, or NOT) of the functional input.
GB (Gigabyte)	= 1, 073,741, 824 bytes ≈1 Billion bytes

H

Hexadecimal Numbers	A base-16 number system that characterize 16 values (0 to 9 and A to F). usually signify memory addresses or data.
Hit Ratio	A measure of cache efficiency = (cache hits / cache misses)
Hz (Hertz)	The number of clock cycles per second. by and large specified in KHz (Kilohertz) or MHz (Megahertz).

I

IC (Integrated Circuit / Chip)	A hardware part generally prepared of silicon that hosts dozens to millions of transistors on a tiny region.
Instruction Format	Instruction outline that break up instruction to fields related to element of instruction (opcode, operands,)
Instruction Set	Total collection of instructions used by a machine.
Interrupt	An exception that arrives from outside the processor.
Interrupt Handler	A software program part that is run when an interrupt happens.
ISA (Instruction Set Architecture)	An conceptual medium between the hardware and the software of a device that include essential information to write accurate machine-language program. It comprise the requirement of commands, registers, memory size, instruction, ...etc.

K

K (Kilo)	$2^{10} = 1024$

L

L1 (Level 1) Cache	Cache situated contiguous to the processor. That is Primary cache.
L2 (Level 2) Cache	Cache positioned outside the processor. Known as secondary cache.
LAN (Local Area Network)	Network link that transmit data in a small geographic region, classically within the same building.
Latency	Wait or delay time(milliseconds).
Link Editor (Linker)	A system program that merges separately integrate machine language program and determine indeterminate labels. The consequential code is in executable code form.
Local Variable	A variable is defined and accessed in a particular unit of a program only.
Locality (of Reference) Principle	propensity of a program to access the *same* set of memory locations continually over small phase of time.
LRU (Least Recently Used) Scheme	A substitute method in which the new preferred block replaces the block that has been idle the longest time.

M

M (Mega)	2^{20} = 1,048,576
Mantissa	The part of a floating point number which, when multiply by its base raised to the power of its exponent, provide its value.
MAR (Memory address Register)	A CPU registers for keeping address of memory location being accessed.
Microinstruction	Low-level control instruction in which machine instruction is used to produce control signals.
Micro-operation	Basic CPU operation, carried out for the period of one clock cycle.
Microprocessor	Integrated circuits making the heart of the PC that include ALU, general and special registers, and Control units.
Micro program	Microinstruction sequence.
MIMD (Multiple Instruction stream, Multiple-Data stream)	The categorization under Flynn's nomenclature of a parallel processor where many functional parts carry out different function on different data.
MIPS (Million Instruction Per Second)	Determine execution speed. MIPS = Number of instructions in a program / (program execution time x 10^6)
Miss Rate	Portion of memory access not found in cache.
Motherboard	A large printed-circuit board used to host PC components
Multiprocessor	Computer having more than one processor with common main memory and single address room.
Multiprogramming	Programming mode that permit two or more programs to execute interleaved by a single CPU.

N

Nonvolatile memory	Memory whose data keeps integral even when the power is turned off.

O

Opcode (Operation Code)	Component of an instruction that indicate the operation and format of an instruction.
Operand	Entity on which an operation is carry out. piece of an

	instruction.
Operating System	System software to control program execution, assign and deal with resources, program tasks, control I/O operations, and manage data.

P

Page	A fixed length memory blocks that has virtual address and is swap as a entity amid two memory types (RAM & cache or RAM & secondary memory).
Page Fault	A condition that takes place when referencing a memory word that is not in RAM. It originates interrupt & needs loading the page ongoing the preferred word before the program can go on.
Page Frame	A block in RAM that can keep a page.
Parity Bit	An extra bit attached to a word and basis for sum of all digits to be either odd or even, depending on the type of parity (odd or even parity).
Peer-to-Peer Network	Two or more computers directly linked and directly share the data and hardware resources.
Pipeline	A method of mounting processor efficiency. Instructions are alienated into smaller stages, each stage employ different resource. More than one Instruction can run concurrently, each using different resource.

R

Radix (Base)	Representation of Number system.
Register	High-speed memory constituent exist in in the CPU used to keep data.

S

SCSI (Small Computer System Interface)	A bus used as a standard for I/O devices.
Seek Time	Time required for the head actuator to travel the read/write

	head from one track to the next.
Segmentation	Variable-size address mapping method in which an address is alienated into two parts: a part number and part offset.
Sequential Circuit	Logic circuit in which the next state is a function of both the current state and the input. It works as a memory element.
Sign-Magnitude Representation	Number representation used to represent binary integers. The leftmost bit is used to represent the sign (1 for negative, 0 for positive). rest bits keep the magnitude of the number.
SIMD (Single-Instruction Stream / Single-Data Stream)	Multiprocessor architecture that can do a single function on multiple set of data.
SIMD (Single-Instruction stream, Multiple-Data stream) (Or "data parallel")	The categorization under Flynn's nomenclature for a parallel processor where many processing part do the same action on different data. There is often a central controller that broadcasts the instruction stream to all the processing elements.
SIMM (Single-Inline Memory Module)	Memory unit made up of DRAM chip in special packaging. Soldered on a tiny circuit board with 30- or 72- edge connector.
SISD (Single-Inline, Single-Data Stream)	Computer organization in Flynn's classification that refers to the conservative processor.
Snooping Cache Memory	technique for retain cache coherency in which all cache controllers scrutinize the bus to decide whether or not they have the preferred block.
Spatial Locality	Locality principle that states that data referencing be apt to reference close by addresses.
SRAM (Static RAM)	RAM put into practice with flip-flops. Data keeps as long as the power is on. No periodic revitalizing is required.
Stack	A list that is efforts on LIFO (Last-In First-Out) basis.
Sum-of-Product	A logical expression merge AND terms (Product) and then applies the OR operator (Sum) on them.
Superscalar Pipelining	A system that copies the internal computer components to allow multiple instructions to be run in every pipeline stage.
Superscalar Processor	An advanced pipelining modus operandi in which more than one instruction can execute during one clock cycle, each on a different pipeline stage.
Synchronous Timing	Timing practice in which incidence of events on a bus are

	given by the clock.
System Bus	A bus used to be linked major computer components.

T

Tag	A field in a table having address information to discover a memory block in which a particular word is found.
TB (Terabytes)	= 1, 099,511, 627, 776 bytes ≈1 TB
TLB (Translation Lookaside buffer	Used in virtual memory systems. A cache that keeps track of lately used address mapping to circumvent an access to the page table. It lists the physical address page number related with every virtual address page number.
Transistor	Electronic circuit that holds electrical voltage representing one bit.
Truth table	Table showing logic task by listing all potential input combinations and their subsequent output values.

U

Unary Operator	An operator that perform on one operand only
Underflow	A condition that can take place when the outcome of a floating-point function would be lesser in scale (closer to zero, either positive or negative) than the smallest quantity characterizeable. Underflow is in fact (negative) overflow of the exponent of the floating-point quantity. It happens when a negative exponent is too great to be characterized. That means the number is too small to be represented.
Virtual Address	A memory location accessed in a system by an application program with virtual memory such that intervening hardware and/or software maps the virtual address to real (physical) memory.
Virtual Memory	Address space that can be seen as addressable main memory by the user. They are plotted by the processor into physical address space. Usually the virtual address space is bigger than the physical address space.
Volatile Memory	Memory that lose its contents when the power is off. Example: RAM.

W

Write Back	A cache structural design in which data is only written to main memory when it is enforced out of the cache. Contrary to write-through.
Write Through	A cache structural design in which data is written to main memory at the similar time as it is cached.

References

Websites:

- http://williamstallings.com/COA5e.html
- http://iiusatech.com/murdocca/CAO/
- http://courses.soe.ucsc.edu/courses/cmpe202
- http://www.randelshofer.ch/fhw/gri/float.html#chapterfloatapplet
- http://courses.soe.ucsc.edu/courses/cmpe110
- http://booksite.mkp.com/9780123838728/
- http://inst.eecs.berkeley.edu/~cs152/sp12/
- http://www.randelshofer.ch/fhw/gri/float.html
- http://www.britannica.com/EBchecked/topic/163278/digital-computer
- http://williamstallings.com/COA/COA7e.html
- http://csg.csail.mit.edu/6.823/lecnotes.html
- http://www.cs.berkeley.edu/~pattrsn/252S01/
- http://gs.fanshawec.ca/tlc/math270/
- http://courses.cs.vt.edu/~csonline/NumberSystems/Lessons/BinaryNumbers/index.html
- http://www.math.grin.edu/~stone/courses/fundamentals/IEEE-reals.html

- http://www.randelshofer.ch/fhw/gri/doubleapplet.html

Books:

1. Book Name : Computer Systems Design And Architecture
   Edition: 2/e
   ISBN : 9788177584837
   Author : Vincent P. Heuring,Harry F. Jordan,T.G Venkatesh

2. Book Name : Computer Architecture: Concepts And Evolution
   ISBN : 9788131726143
   Author : Gerrit A. Blaauw, Frederick P. Brooks
   Edition : 1/e

3. Book Name : Computer Systems Organization & Architecture
   ISBN : 9788177587678
   Author : John D. Carpinelli
   Edition : 1/e

4. Book Name : Essentials Of Computer Architecture
   ISBN : 9788177584431
   Author : Douglas E. Comer
   Edition : 1/e

5. Book Name : Principles Of Computer Organization And Assembly Language
   ISBN : 9788131721711
   Author : Patrick Juola
   Edition : 1/e

6. Book Name : Computer System Architecture

ISBN : 9788131700709
Author : M. Morris Mano
Edition : 3/e

7. Book Name : Computer Organization And Architecture
ISBN : 9788131732458
Author : William Stallings
Edition : 8/e

8. Book Name : Computer Architecture: A Quantitative Approach,
Author : John L. Hennessy , David A. Patterson
Edition : 4th Edition

9. Book Name : Parallel Computer Architecture: A Hardware/Software Approach
Author : David Culler and J.P. Singh with Anoop Gupta

10. Book Name : Digital Design And Computer Organisation
ISBN : 978-81-318-0345-5
Author : D. Nasib S. Gill, J.B. Dixit
Edition : 1/e

11. Book Name : Structured Computer Organization (5th edition)
Author : Andrew S. Tannenbaum, Prentice Hall

12. Book Name : Computer Organization and Architecture: Designing for Performance
Edition:7th Ed
Author : William Stallings, Prentice-Hall.

13. Book Name : Guide to Assembly Language Programming in Linux
Author : S. Dandamudi, Springer.