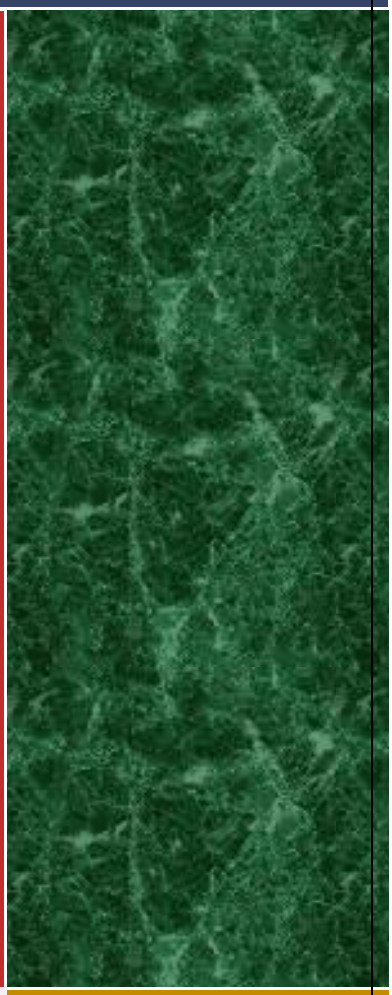
DIWAKAR EDUCATION HUB

Computer System Architecture Unit - 2

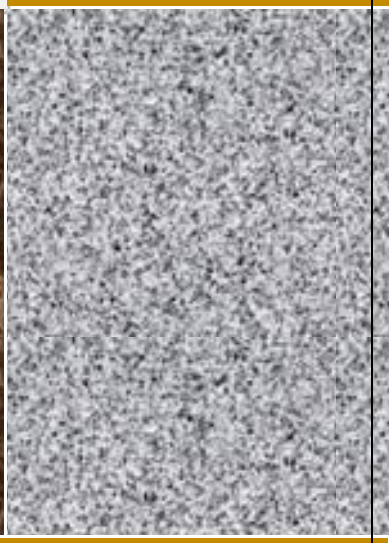
As per updated syllabus

DIWAKAR EDUCATION HUB



2020





THE LEARN WITH EXPERTIES

Digital Logic Circuits

The digital logic circuits are basic building blocks of the digital systems (digital computers). These digital logic circuits can be classified into two categories such as combinational logic circuits and sequential logic circuits. Before studying about the difference between combinational and sequential logic circuits, primarily, we must know what is combinational logic circuit and what are sequential logic circuits.

Digital Computers

A Digital computer can be considered as a digital system that performs various computational tasks.

The first electronic digital computer was developed in the late 1940s and was used primarily for numerical computations.

By convention, the digital computers use the binary number system, which has two digits: 0 and 1. A binary digit is called a bit.

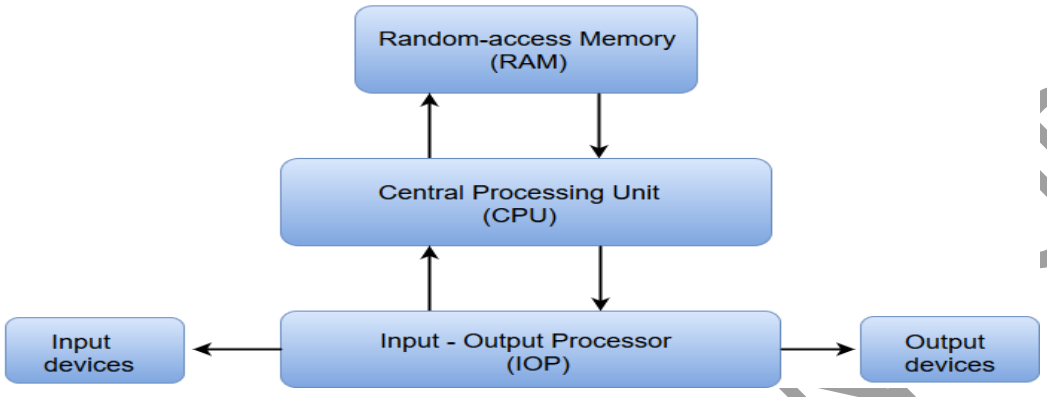
A computer system is subdivided into two functional entities: Hardware and Software.

The hardware consists of all the electronic components and electromechanical devices that comprise the physical entity of the device.

The software of the computer consists of the instructions and data that the computer manipulates to perform various data-processing tasks.

DIWAKAR EDUCATION HUB

Block diagram of a digital computer:



- The Central Processing Unit (CPU) contains an arithmetic and logic unit for manipulating data, a number of registers for storing data, and a control circuit for fetching and executing instructions.
- The memory unit of a digital computer contains storage for instructions and data.
- The Random Access Memory (RAM) for real-time processing of the data.
- The Input-Output devices for generating inputs from the user and displaying the final results to the user.
- The Input-Output devices connected to the computer include the keyboard, mouse, terminals, magnetic disk drives, and other communication devices.

Logic Gates

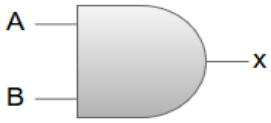
- The logic gates are the main structural part of a digital system.
- Logic Gates are a block of hardware that produces signals of binary 1 or 0 when input logic requirements are satisfied.
- Each gate has a distinct graphic symbol, and its operation can be described by means of algebraic expressions.
- The seven basic logic gates includes: AND, OR, XOR, NOT, NAND, NOR, and XNOR.
- The relationship between the input-output binary variables for each gate can be represented in tabular form by a truth table.
- Each gate has one or two binary input variables designated by A and B and one binary output variable designated by x.

AND GATE:

DIWAKAR EDUCATION HUB

The AND gate is an electronic circuit which gives a high output only if all its inputs are high. The AND operation is represented by a dot (.) sign.

AND Gate:



Algebraic Function: x = AB

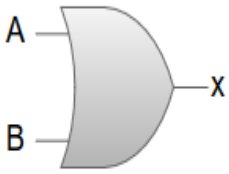
Truth Table:

A	B	x
0	0	0
0	1	0
1	0	0
1	1	1

OR GATE:

The OR gate is an electronic circuit which gives a high output if one or more of its inputs are high. The operation performed by an OR gate is represented by a plus (+) sign.

OR Gate:



Algebraic Function: x = A + B

Truth Table:

A	B	x
0	0	0
0	1	1
1	0	1
1	1	1

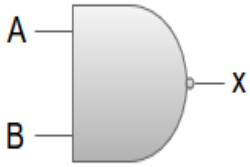
NOT GATE:

The NOT gate is an electronic circuit which produces an inverted version of the input at its output. It is also known as an **Inverter**.

NAND GATE:

The NOT-AND (NAND) gate which is equal to an AND gate followed by a NOT gate. The NAND gate gives a high output if any of the inputs are low. The NAND gate is represented by a AND gate with a small circle on the output. The small circle represents inversion.

NAND Gate:



Algebraic Function: x = (AB)'

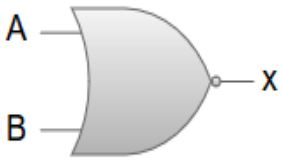
Truth Table:

A	B	x
0	0	1
0	1	1
1	0	1
1	1	0

NOR GATE:

The NOT-OR (NOR) gate which is equal to an OR gate followed by a NOT gate. The NOR gate gives a low output if any of the inputs are high. The NOR gate is represented by an OR gate with a small circle on the output. The small circle represents inversion.

NOR Gate:



Algebraic Function: x = (A+B)'

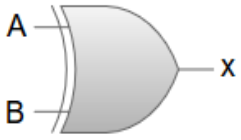
Truth Table:

A	B	x
0	0	1
0	1	0
1	0	0
1	1	0

Exclusive-OR/ XOR GATE:

The 'Exclusive-OR' gate is a circuit which will give a high output if one of its inputs is high but not both of them. The XOR operation is represented by an encircled plus sign.

XOR Gate:



Algebraic Function: $x = A \oplus B$

or

$x = A'B + AB'$

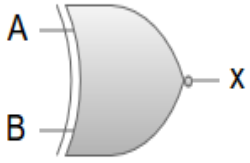
Truth Table:

A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

EXCLUSIVE-NOR/Equivalence GATE:

The 'Exclusive-NOR' gate is a circuit that does the inverse operation to the XOR gate. It will give a low output if one of its inputs is high but not both of them. The small circle represents inversion.

Exclusive-NOR Gate:



Algebraic Function: $x = (A \oplus B)'$

or

$x = A'B' + AB$

Truth Table:

A	B	x
0	0	1
0	1	0
1	0	0
1	1	1

Boolean Algebra

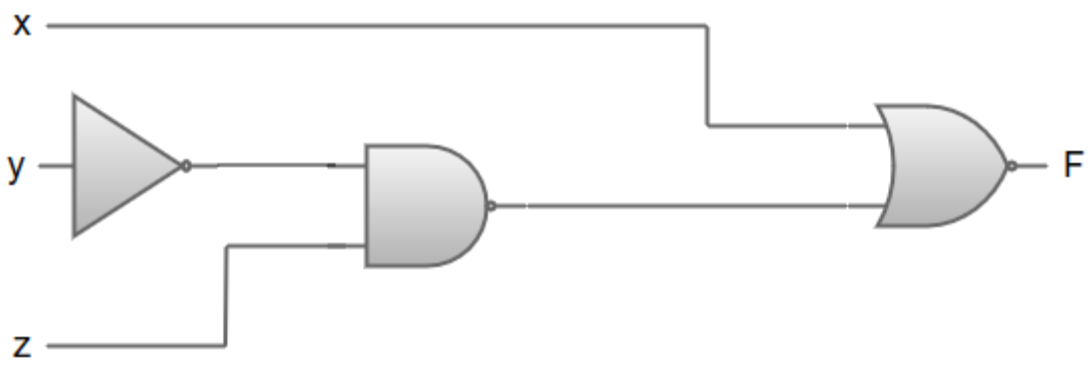
Boolean algebra can be considered as an algebra that deals with binary variables and logic operations. Boolean algebraic variables are designated by letters such as A, B, x, and y. The basic operations performed are AND, OR, and complement.

The Boolean algebraic functions are mostly expressed with binary variables, logic operation symbols, parentheses, and equal sign. For a given value of variables, the Boolean function can be either 1 or 0. For instance, consider the Boolean function:

$$F = x + y'z$$

The logic diagram for the Boolean function F = x + y'z can be represented as:

F = x + y'z



- The Boolean function F = x + y'z is transformed from an algebraic expression into a logic diagram composed of AND, OR, and inverter gates.
- Inverter at input 'y' generates its complement y'.
- There is an AND gate for the term y'z, and an OR gate is used to combine the two terms (x and y'z).
- The variables of the function are taken to be the inputs of the circuit, and the variable symbol of the function is taken as the output of the circuit.

DIWAKAR EDUCATION HUB

Note: A truth table can represent the relationship between a function and its binary variables. To represent a function in a truth table, we need a list of the 2^n combinations of n binary variables.

The truth table for the Boolean function F = x + y'z can be represented as:

F = x + y'z

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Laws of Boolean algebra

The basic Laws of Boolean Algebra can be stated as follows:

- Commutative Law states that the interchanging of the order of operands in a Boolean equation does not change its result. For example:
  1. OR operator → A + B = B + A
  2. AND operator → A * B = B * A
- Associative Law of multiplication states that the AND operation are done on two or more than two variables. For example:
  A * (B * C) = (A * B) * C
- Distributive Law states that the multiplication of two variables and adding the result with a variable will result in the same value as multiplication of addition of the variable with individual variables. For example:
  A + BC = (A + B) (A + C).

DIWAKAR EDUCATION HUB

- Annulment law:
  A.0 = 0
  A + 1 = 1
- Identity law:
  A.1 = A
  A + 0 = A
- Idempotent law:
  A + A = A
  A.A = A
- Complement law:
  A + A' = 1
  A.A'= 0
- Double negation law:
  ((A)')' = A
- Absorption law:
  A.(A+B) = A
  A + AB = A

De Morgan's Law is also known as De Morgan's theorem, works depending on the concept of Duality. Duality states that interchanging the operators and variables in a function, such as replacing 0 with 1 and 1 with 0, AND operator with OR operator and OR operator with AND operator.

De Morgan stated 2 theorems, which will help us in solving the algebraic problems in digital electronics. The De Morgan's statements are:

1. "The negation of a conjunction is the disjunction of the negations", which means that the complement of the product of 2 variables is equal to the sum of the compliments of individual variables. For example, (A.B)' = A' + B'.
2. "The negation of disjunction is the conjunction of the negations", which means that compliment of the sum of two variables is equal to the product of the complement of each variable. For example, (A + B)' = A'B'.

Map Simplification

The Map method involves a simple, straightforward procedure for simplifying Boolean expressions.

DIWAKAR EDUCATION HUB

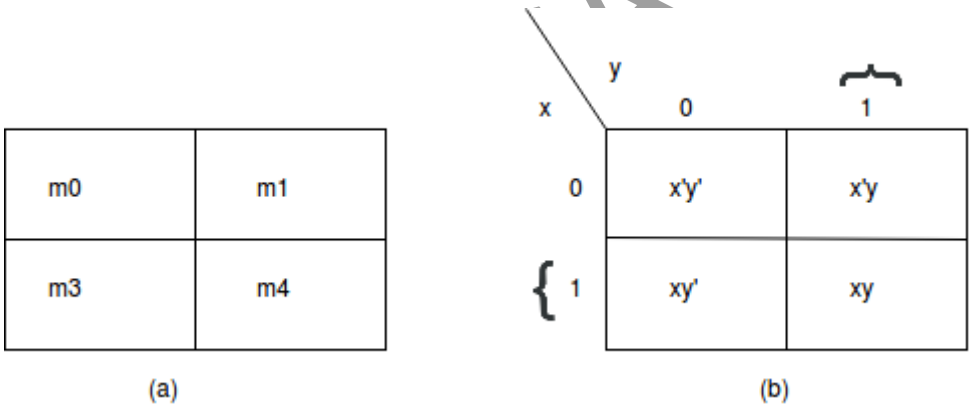
Map simplification may be regarded as a pictorial arrangement of the truth table which allows an easy interpretation for choosing the minimum number of terms needed to express the function algebraically. The map method is also known as Karnaugh map or K-map.

Each combination of the variables in a truth table is called a mid-term.

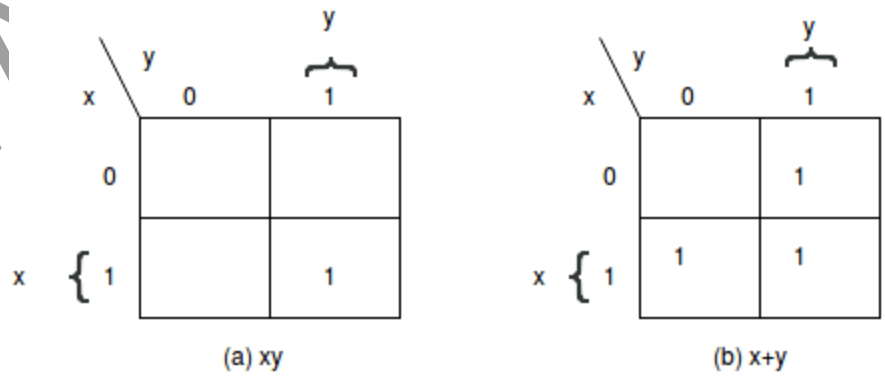
Note: When expressed in a truth table a function of n variables will have 2^n min-terms, equivalent to the 2^n binary numbers obtained from n bits.

There are four min-terms in a two variable map. Therefore, the map consists of four squares, one for each min-term. The 0's and 1's marked for each row, and each column designates the values of variable x and y, respectively.

Two-variable map:



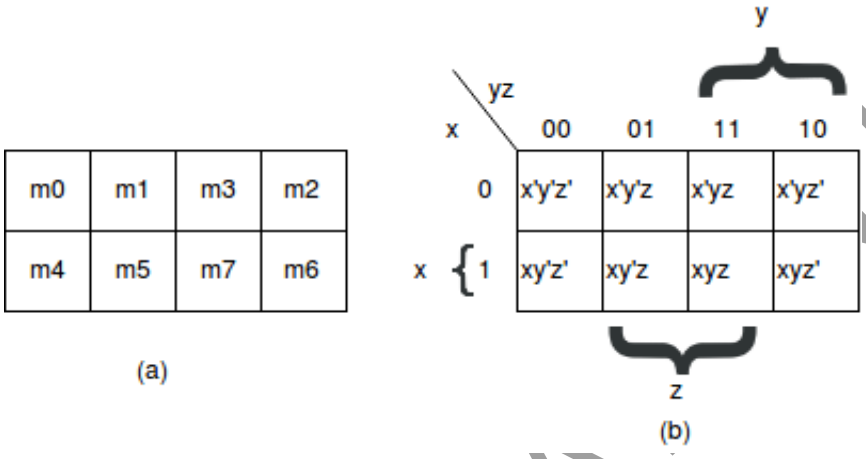
Representation of functions in the two-variable map:



Three variable map

There are eight min-terms in a three-variable map. Therefore, the map consists of eight squares.

Three variable map:



- The map was drawn in part (b) in the above image is marked with numbers in each row and each column to show the relationship between the squares and the three variables.
- Any two adjacent squares in the map differ by only one variable, which is primed in one square and unprimed in the other. For example, m5 and m7 lie in the two adjacent squares. Variable y is primed in m5 and unprimed in m7, whereas the other two variables are the same in both the squares.
- From the postulates of Boolean algebra, it follows that the sum of two min-terms in adjacent squares can be simplified to a single AND term consisting of only two literals. For example, consider the sum of two adjacent squares say m5 and m7:
  m5+m7 = xy'z+xyz= xz(y'+y)= xz.

Logic circuits are classified into two types:

- Sequential logic circuits
- Combinational logic circuits.

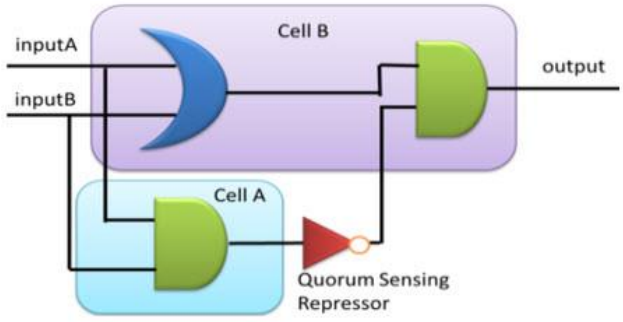
Combinational Logic Circuits

DIWAKAR EDUCA HUB

The combinational logic circuits or time-independent logic circuits in digital circuit theory can be defined as a type of digital logic circuit implemented using Boolean circuits, where the output of logic circuit is a pure function of the present inputs only. The combinational logic circuit operation is instantaneous and these circuits do not have the memory or feedback loops.

This combinational logic is in contrast compared to the sequential logic circuit in which the output depends on both present inputs and also on the previous inputs. Thus, we can say that combinational logic does not have memory, whereas sequential logic stores previous input in its memory. Hence, if the input of combinational logic circuit changes, then the output also changes.

Combinational Logic Circuit Design

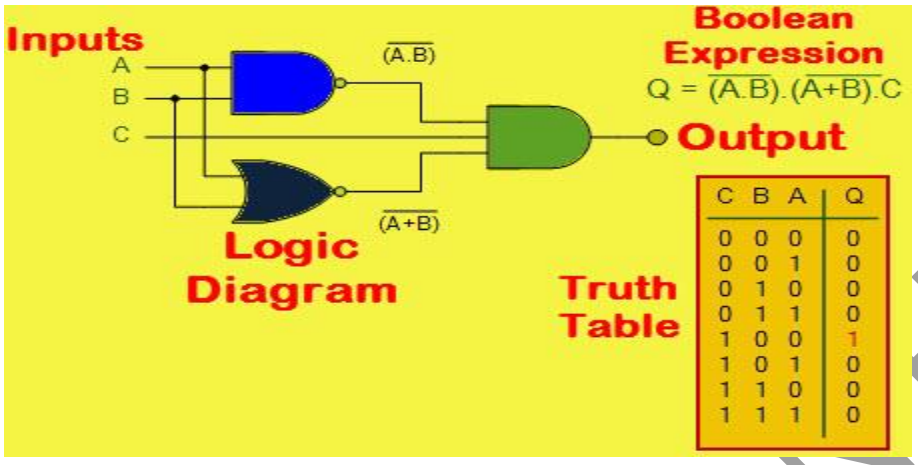


Combinational Logic Circuit

These combinational logic circuits are designed to produce specific outputs from certain inputs. The combinational logic design can be done using two methods such as a sum of products and a product of sums. Combinational logic circuits are generally designed by connecting together or combining the basic logic gates such as NAND, NOR, and NOT. Hence, these logic gates are termed as building blocks. These logic circuits can be a very simple circuit or a very complex circuit or huge combinational circuit can be designed using only universal logic gates such as NAND and NOR gates.

The simple time independent logic circuits that are implemented using Boolean circuits whose output logic value depends only on the input logic values can be called as combinational logic circuits.

DIWAKAR EDUCATION HUB



Combinational Logic Circuit

The figure shows all the three major components of the combinational logic circuit such as logic diagram, truth table, and Boolean expression.

Functions of Combinational Logic Circuit

The function of combinational logic circuits can be specified in three main ways such as:

- Truth Table
- Boolean Algebra
- Logic Diagram

Truth Table

C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Combinational Logic Function Truth Table

DIWAKAR EDUCATION HUB

The logic gate function can be defined using its truth table which consists outputs for of all possible combinations of inputs of the logic gate. An example combinational logic function truth table is shown in the above figure.

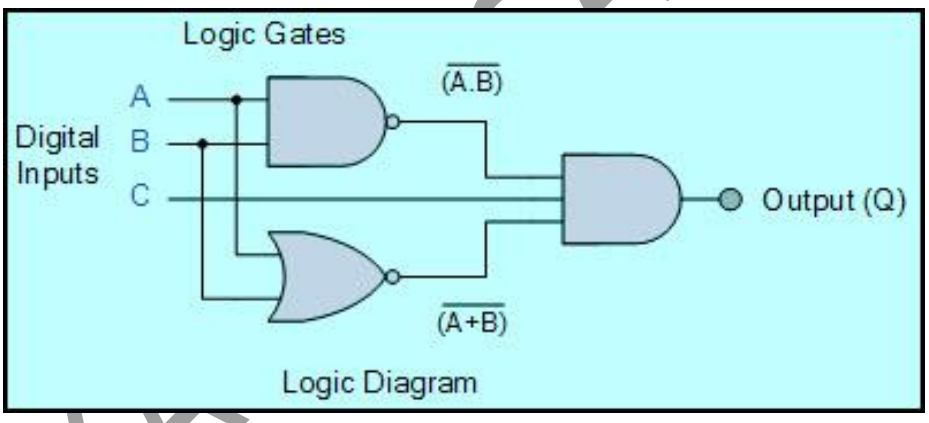
Boolean Algebra

$$Q = (\overline{A.B}).(\overline{A+B}).C$$

Combinational Logic Function Boolean Expression

The output of combinational logic function can be expressed in the form expression using Boolean algebra and an example, Boolean expression for the above truth table is shown in the above figure.

Logic Diagram



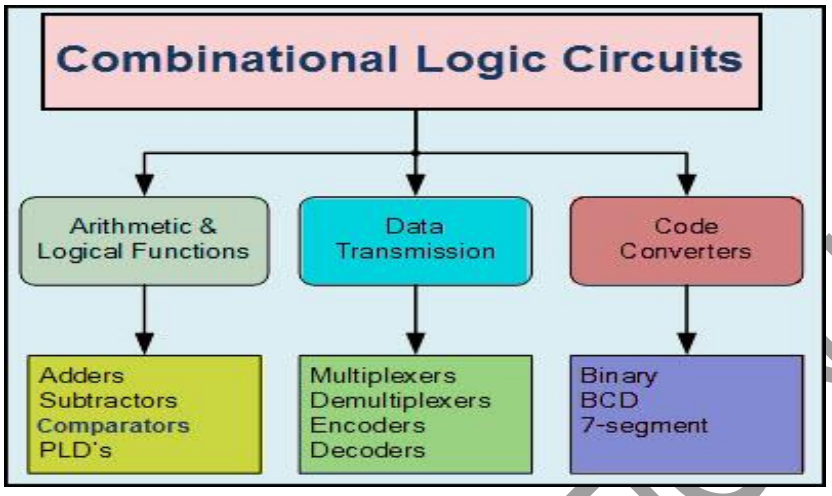
Combinational Logic Circuit using Logic Gates

The graphical representation of combinational logic functions using logic gates is called as logic diagram. The logic diagram for above discussed logic function truth table and Boolean expression can be realized as shown in the above figure.

The combinational logic circuits can be also be called as decision making circuits, as these are designed using individual logic gates. The combinational logic is the process of combining logic gates to process the given two or more inputs such that to generate at least one output signal based on the logic function of each logic gate.

DIWAKAR EDUCATION HUB

Classification of Combinational Logic



Classificaiton of Combinational Logic

The combinational logic circuits can be classified into various types based on the purpose of usage, such as arithmetic & logical functions, data transmission, and code converters. To solve the arithmetic and logical functions we generally use adders, subtractors, and comparators which are generally realized by combining various logic gates called as combinational logic circuits. Similarly, for data transmission, we use multiplexers, demultiplexers, encoders, and decoders which are also realized using combinational logic. The code converters such as binary, BCD, and 7-segment are designed using various logic circuits.

In fact, combinational logic is most frequently used in multiplexer and demultiplexer type circuits. If multiple inputs or outputs are connected to the common signal line, then the logic gates are used for decoding an address in order to select single data input or output switch.

Sequential Logic Circuits

The combinational circuit does not use any memory. Hence the previous state of input does not have any effect on the present state of the circuit. But sequential circuit has memory so output can vary based on input. This type of circuits uses previous input, output, clock and a memory element.

DIWAKAR EDUCATION HUB

The simple logic circuits whose output logic value depends on the input logic values and also on the stored information is called as sequential logic circuits.

Combinational circuit produces an output based on input variable only, but **Sequential circuit** produces an output based on **current input and previous input variables**. That means sequential circuits include memory elements which are capable of storing binary information. That binary information defines the state of the sequential circuit at that time. A latch capable of storing one bit of information.

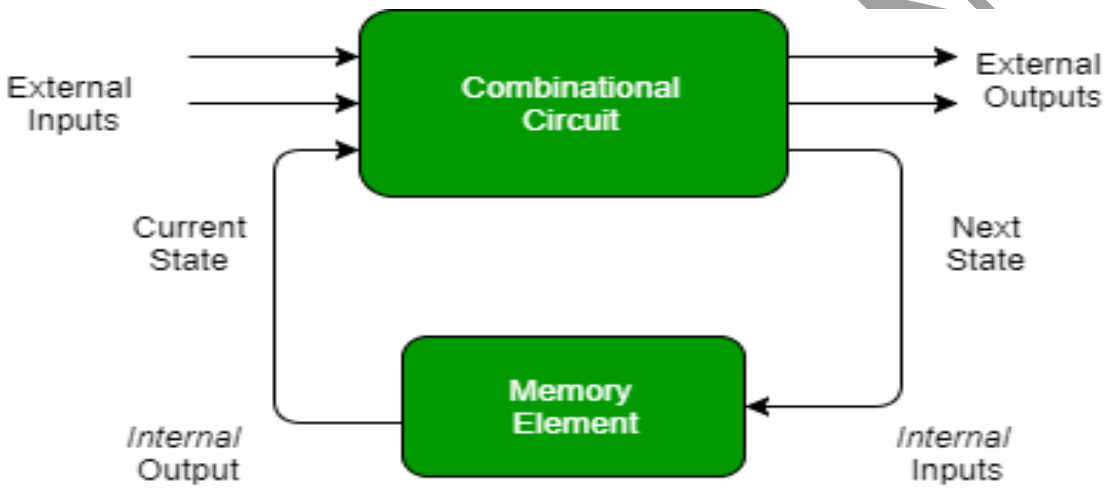


Figure: Sequential Circuit

As shown in figure there are two types of input to the combinational logic :

1. External inputs which not controlled by the circuit.
2. Internal inputs which are a function of a previous output states.

Secondary inputs are state variables produced by the storage elements, where as secondary outputs are excitations for the storage elements.

Types of Sequential Circuits – There are two types of sequential circuit :

1. **Asynchronous sequential circuit –** These circuit **do not use a clock signal** but uses the pulses of the inputs. These circuits are **faster** than synchronous sequential circuits because there is clock pulse and change their state immediately when there is a change in the input signal. We use

DIWAKAR

asynchronous sequential circuits when speed of operation is important and **independent** of internal clock pulse.

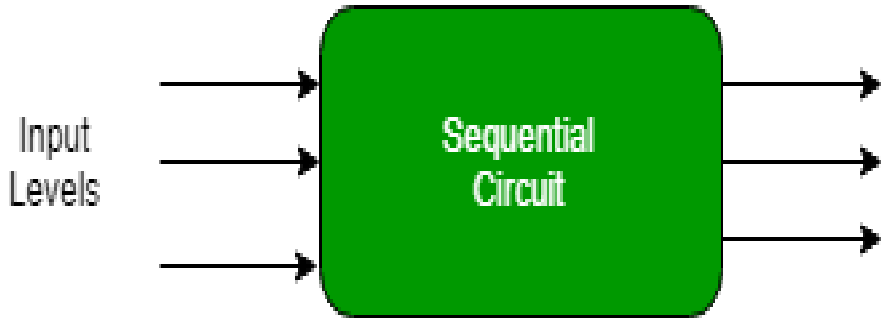


Figure: Asynchronous Sequential Circuit

But these circuits are more **difficult** to design and their output is **uncertain**.

2. **Synchronous sequential circuit –** These circuit **uses clock signal** and level inputs (or pulsed) (with restrictions on pulse width and circuit propagation). The output pulse is the same duration as the clock pulse for the clocked sequential circuits. Since they wait for the next clock pulse to arrive to perform the next operation, so these circuits are bit **slower** compared to asynchronous. Level output changes state at the start of an input pulse and remains in that until the next input or clock pulse.

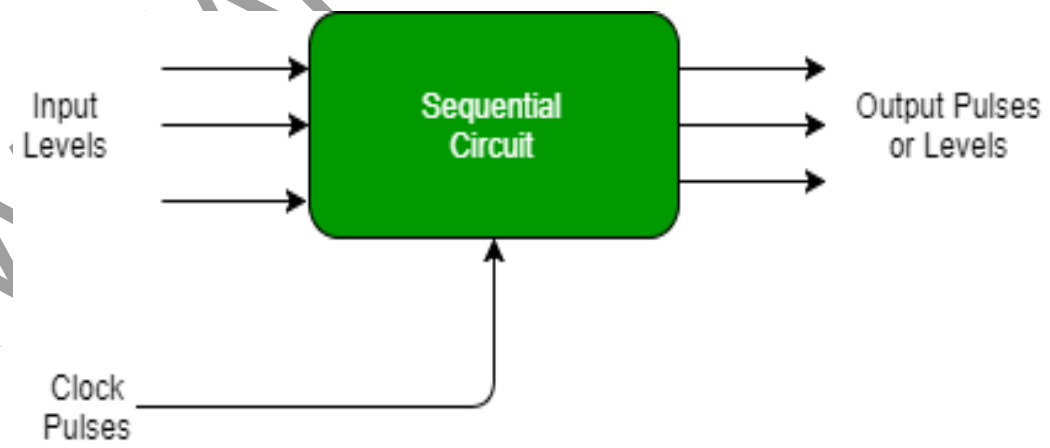
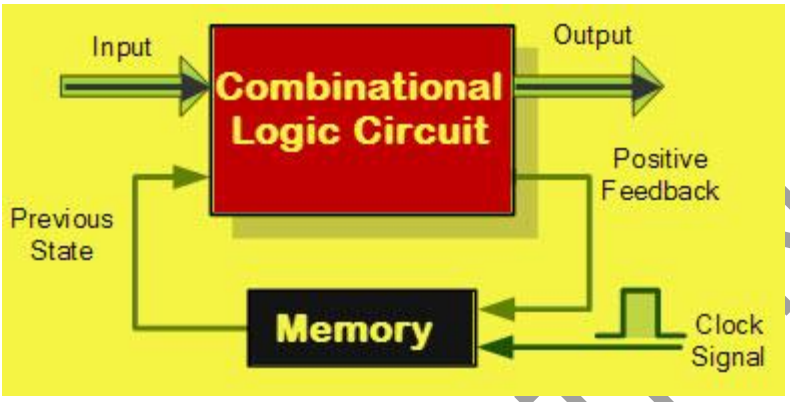


Figure: Synchronous Sequential Circuit

We use synchronous sequential circuit in synchronous counters, flip flops, and in the design of MOORE-MEALY state management machines.



Sequential Logic Circuit

The figure represents the block diagram of the sequential logic circuit.

Difference between Synchronous and Asynchronous Sequential Circuits

As the name suggests both Synchronous and Asynchronous Sequential Circuits are the type of sequential circuits which uses feedback for the next output generation however on the basis of the type of this feedback both circuits can be get differentiated.

Following are the important differences between Synchronous and Asynchronous Sequential Circuits −

Sr. No.	Key	Synchronous Sequential Circuits	Asynchronous Sequential Circuits
1	Definition	Synchronous sequential circuits are digital sequential circuits in which the feedback to the input for next output generation is governed by clock	On other hand Asynchronous sequential circuits are digital sequential circuits in which the feedback to the input for next output generation is not governed by clock signals.

DIWAKAR EDUCATION HUB

Sr. No.	Key	Synchronous Sequential Circuits	Asynchronous Sequential Circuits
		signals.	
2	Memory Unit	In Synchronous sequential circuits, the memory unit which is being get used for governance is clocked flip flop.	On other hand unclocked flip flop or time delay is used as memory element in case of Asynchronous sequential circuits.
3	State	The states of Synchronous sequential circuits are always predictable and thus reliable.	On other hand there are chances for the Asynchronous circuits to enter into a wrong state because of the time difference between the arrivals of inputs. This is called as race condition.
4	Complexity	It is easy to design Synchronous sequential circuits	However on other hand the presence of feedback among logic gates causes instability issues making the design of Asynchronous sequential circuits difficult.
5	Performance	Due to the propagation delay of clock signal in reaching all elements of the circuit the Synchronous sequential circuits are slower in its operation speed	Since there is no clock signal delay, these are fast compared to the Synchronous Sequential Circuits
6	Example	Synchronous circuits are used in counters, shift registers, memory units.	On other hand Asynchronous circuits are used in low power and high speed operations such as simple microprocessors, digital signal processing units and in communication systems for email

DIWAKAR EDUCATION HUB

Sr. No.	Key	Synchronous Sequential Circuits	Asynchronous Sequential Circuits
			applications, internet access and networking.

Difference Between Combinational Logic Circuits and Sequential Logic Circuits

The difference between combinational logic circuits and sequential logic circuits can be easily understood by knowing about each circuit in detail.

Combinational Logic Ciruits

- The digital logic circuits whose outputs can be determined using the logic function of current state input are combinational logic circuits, hence, these are also called as time independent logic circuits.
- Thus, these combinational digital logic circuits don't have the capability to store a state inside them.
- Hence, the combinational logic circuits do not contain any memory elements.
- The arithmetic operations performed on the date stored data in the computers are done using combinational logic circuits.
- The combinational digital logic circuits are fundamentally implemented using different types of devices such as multiplexers, demultiplexers, encoders, decoders, half adder, and full adders.
- The components of arithmetic and logic unit of the computers are generally comprised of combinational digital logic circuits.
- The independent working states of the combinational logic circuits are represented with Boolean algebra and after simplification by using NOR, NOT, and NAND gates the circuit can be implemented.
- The combinational digital circuits don't require any feedbacks.
- The combinational logic circuits are independent of the clock.
- As there are no clocks used in these digital logic circuits, they do not need any triggering.
- The combinational logic circuit's behavior can be defined by using the set of output functions.

DIWAKAR EDUCATION HUB

- In general, sum of products or products of sums method is used for the construction of combinational logic.

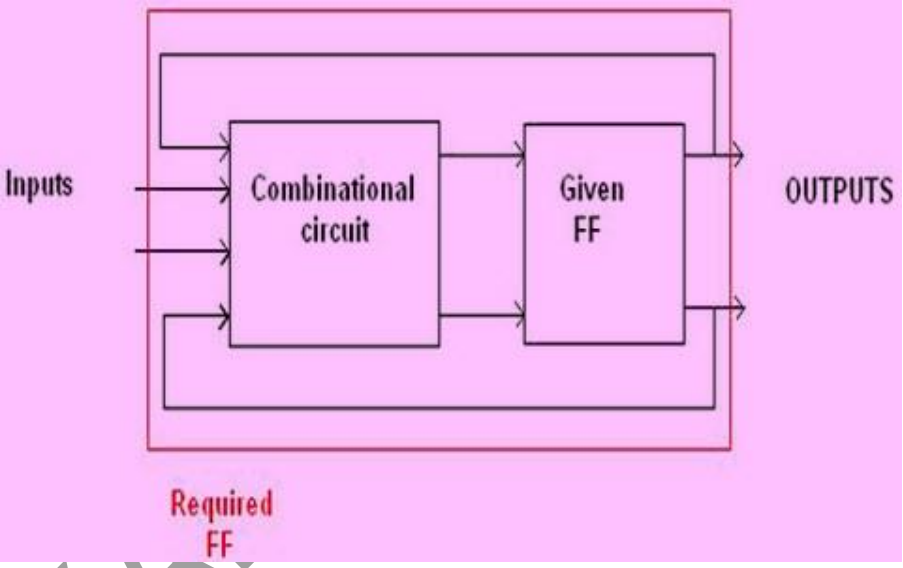
Sequential Logic Circuits

- The digital logic circuits whose outputs can be determined using the logic function of current state inputs and past state inputs are called as sequential logic circuits.
- These sequential digital logic circuits are capable to retain the earlier state of the system based on the current inputs and earlier state.
- Hence, unlike the combinational logic circuits, these sequential digital logic circuits are capable of storing the data in a digital circuit.
- The sequential logic circuits contain memory elements.
- The latch is considered as the simplest element used to retain the earlier memory or state in the sequential digital logic.
- Latches can also be called as flip-flops, but, if we consider the true structural form, then it can be considered as a combinational circuit with one or more than one outputs fed back as inputs.
- These sequential digital logic circuits are used in maximum types of memory elements and also in finite state machines, which are replica watches for sale digital circuit models with finite possible states.
- The maximum number of sequential logic circuits uses a clock for triggering the flip flops operation.
- If the flip flop in the digital logic circuit is triggered, then the circuit is called as synchronous sequential circuit and the other circuits (which are simultaneously not triggered) are called as asynchronous sequential circuits.
- The sequential digital logic circuits utilize the feedbacks from outputs to inputs.
- The sequential logic circuit's behavior can be defined by using the set of output functions and set of next state or memory functions.
- In practical digital logic circuits, combinational digital logic circuits and sequential digital logic circuits are used.

Different Types of Flip Flop Conversion

DIWAKAR EDUCATION HUB

In electronics, flip flop is an electronic circuit and is also called as a latch. Flip flops consist of two stable states which are used to store the data. These are basic building blocks of a digital electronic system which are used in various systems like communications, computers, etc. A basic flip flop can be used to construct a cross coupled inverting elements like invert gates, FETs, BJTs, inverters, vacuum tubes. Conversion of one type of flip flop to another can be done by using a combinational logic circuit. For instance, If a JK Flip Flop is necessary, the i/ps are given to the combinational circuit & the o/p of the combinational circuit is given to the i/ps of the actual flip-flop. Therefore, the o/p of the actual flip-flop is the o/p of the required flip-flop.



Flip Flop Conversion

Flip Flop Conversion

The main purpose of the flip flop conversion is to convert a flip flop into a desired type-B flip flop using some conversion logic. The flip flop conversions are classified into different types that are

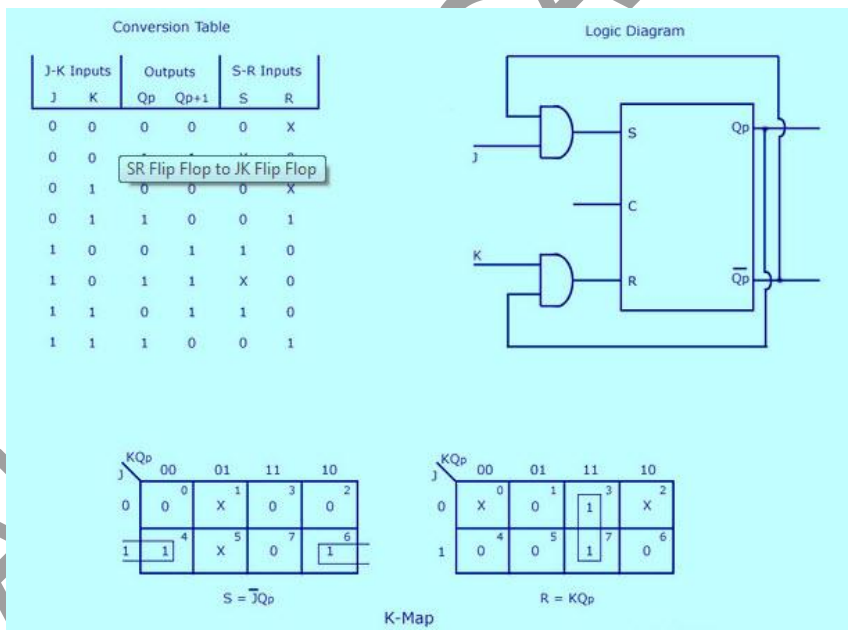
- SR-FF to JK-FF Conversion
- JK-FF to SR-FF Conversion

DIWAKAR EDUCATION HUB

- SR-FF to D-FF Conversion
- D-FF to SR-FF Conversion
- JK-FF to T-FF Conversion
- JK-FF to D-FF Conversion
- D-FF to JK-FF Conversion

SR-Flip Flop to JK-Flip Flop Conversion

In JK-flipflop, j and k are given as external i/ps to S and R in SR-flip flop. Here, both S & R are o/ps of the combinational circuit. The truth tables of flip flop conversions are shown below. The current state is denoted with Qp & Qp+1 is the next state to be found when the J &K i/ps are applied.



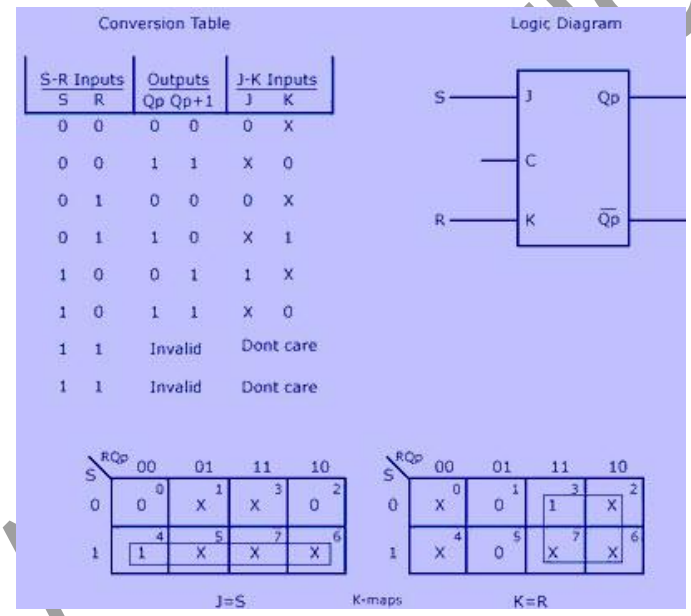
SR-FF to JK-FF Conversion

There are eight possible combinations for two i/ps J and K. For every combination of J, K & Qp, the equivalent Qp+1 states are found. Qp+1 simply recommends the future values to be found by the JK-flip flop after the importance of Qp. Then the table is finished by writing the values of S & R compulsory to get each Qp+1 from

the equivalent Qp. That is, the S and R values are compulsory to change the state of the flip flop from Qp to Qp+1 are written

JK-Flip Flop to SR-Flip Flop Conversion

The conversion of the JK-FF to SR-FF is opposite to the SR-FF to JK-FF. Here S & R will be the external i/ps to J & K, that is shown in the below logic diagram, J & K will be the o/ps of the combinational circuit. So, the J and K values have to be acquired in terms of S, R & Qp. The logic diagram is shown below. The conversion table for flip flop to be written in terms of S, R, Qp, Qp+1, J & K. There are eight possible combinations for two i/ps S and R.



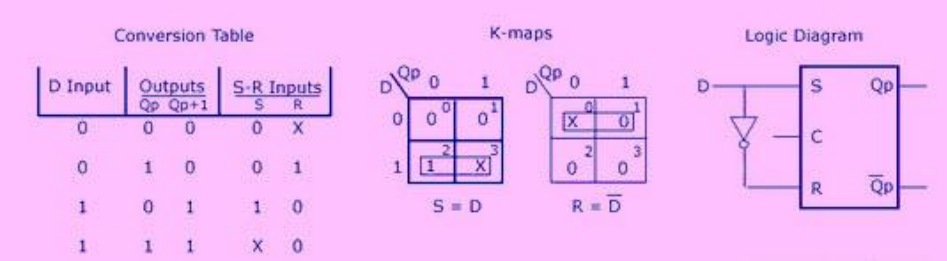
JK-FF to SR-FF Conversion

For every combination, the equivalent Qp+1 o/p's are found. The o/p's for the combinations of S=R=1 are not acceptable for an SR-FF. Therefore the o/p's are considered as invalid and the J & K values are taken as "don't care".

SR-Flip Flop to D-Flip Flop Conversion

DIWAKAR EDUCATION HUB

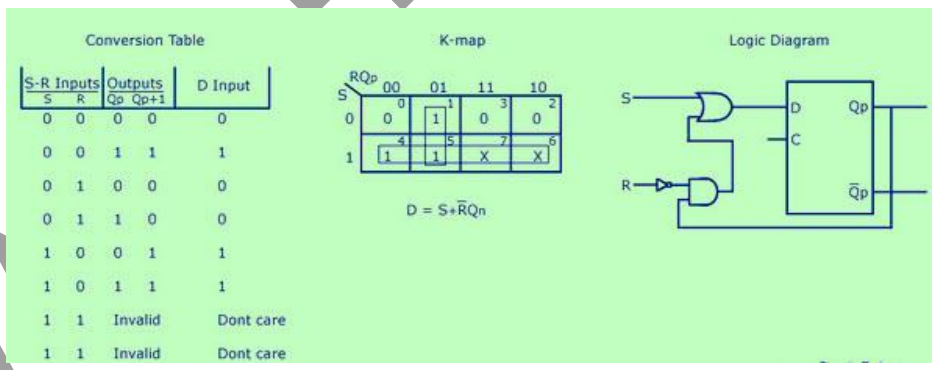
As shown in the below figure, actual inputs of the flip flop are S & R where D is the external i/p. The four combinations of the S & R in terms of D and Qp, conversion table, logic diagram and the Karnaugh map are given below.



SR-FF to D-FF Conversion

D-Flip Flop to SR-Flip Flop Conversion

In this type of conversion, D is the actual i/p of the flip flop where S & R are the external i/ps.There are Eight possible combinations are obtained from the external i/ps S, R & Qp. Nevertheless, since the combination of S=R=1 is unacceptable, the values of D and Qp+1 are taken as “don’t care”. The logic diagram of D-FF to SR-FF is showing the conversion from D-FF to SR-FF, and the Karnaugh map for D in terms of S, R & Qp are given below.



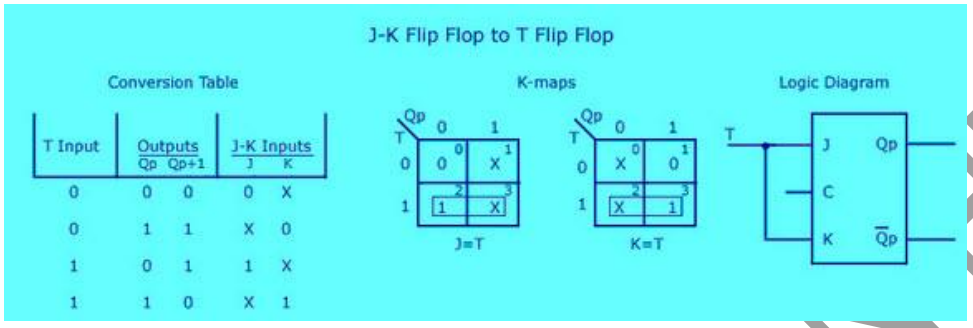
D-FF to SR-FF Conversion

JK-Flip Flop to T-Flip Flop Conversion

In this type of conversion, J & k are the actual i/ps of the flip flop where K is considered as the external i/p. Four combinations are created by T, Qp, J & K that

DIWAKAR EDUCATION HUB

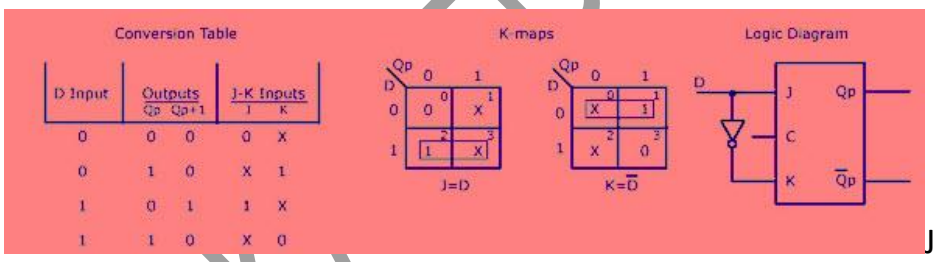
are expressed in terms of T & Qp. The Karnaugh map, the logic diagram and conversion table, are given below.



JK-FF to T-FF Conversion

JK-Flip Flop to D-Flip Flop Conversion

In this type of flip flop conversion, J&K are the actual inputs where D is the external input of the flip flop. The four combinations of the flip flop will be done by using D & Qp, and in terms of these two J&K are expressed. The conversion table with four combinations, JK-FF to D-FF conversion logic diagram and Karnaugh map for J & K in terms of D & are shown below.

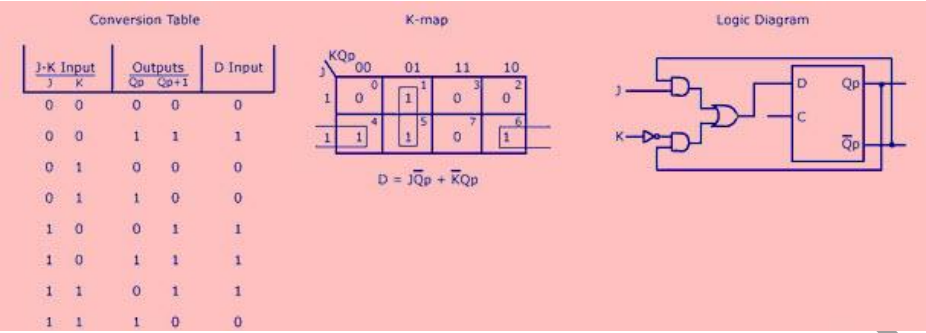


JK-FF to D-FF Conversion

D-Flip Flop to JK-Flip Flop Conversion

In this type of flip flop conversion, J & K are the external i/ps of the flip flop where D is the actual input. The eight combinations can make by using J, K and Qp that is shown in the conversion table below. D is stated in terms of J, K & Qp. The Karnaugh map D in terms of J, K & Qp, conversion table and the logic diagram of the D-FF to JK-FF is shown below.

DIWAKAR EDUCATION HUB



D-FF to JK-FF Conversion

Thus, this is all about different types of flip flop conversions, that includes SR-FF to JK-FF , JK-FF to SR-FF , SR-FF to D-FF , D-FF to SR-FF , JK-FF to T-FF , JK-FF to D-FF and D-FF to JK-FF . We hope that you have got a better understanding of this concept. Furthermore, any doubts regarding the applications of flip-flops or electronics projects, please give your feedback by commenting in the comment section below.Here is a question for you, what are the applications of flip flops?

Integrated Circuit

A microprocessor is digital is a digital circuit which is built using a combination logic functions. The microprocessor package contains an integrated circuit.

An integrated circuit is electronic circuit or device that has electronic components on a small semiconductor chip. It has functionality of logic AND or amplifying of a signal. These are mainly two types of circuits: Digital or Analog. Analog ICs handle continuous signals such as audio signals and Digital ICs handle discrete signals such as binary values.

An integrated circuit, or IC, is small chip that can function as an amplifier, oscillator, timer, microprocessor, or even computer memory. An IC is a small wafer, usually made of silicon, that can hold anywhere from hundreds to millions of transistors, resistors, and capacitors. These extremely small electronics can perform calculations and store data using either digital or analog technology.

Digital ICs use logic gates, which work only with values of ones and zeros. A low signal sent to to a component on a digital IC will result in a value of 0, while a high

DIVAKAR EDUCATION HUB

signal creates a value of 1. Digital ICs are the kind you will usually find in computers, networking equipment, and most consumer electronics.

Analog, or linear ICs work with continuous values. This means a component on a linear IC can take a value of any kind and output another value. The term "linear" is used since the output value is a linear function of the input. For example, a component on a linear IC may multiple an incoming value by a factor of 2.5 and output the result. Linear ICs are typically used in audio and radio frequency amplification.

Types of Integrated Circuits

There are different types of integrated circuits based various criteria. Based on intended application, the Integrated Circuit (IC) can be classified as following:

- Digital Integrated Circuits handle discrete signals such as binary values (0 and 1). These circuits use digital logic gates, multiplexers, flip flops etc.These circuits are easier to design and economical.
- Analog Integrated Circuits handle contiguous signals. These are two types: linear integrated circuits (Linear ICs) and Radio frequency integrated circuits (RF ICs).
- Mixed Integrated Circuits are obtained by the combination of analog and digital integrated circuits. Therefore it have digital to analog (A/D) converter, digital to analog (D/A) converter, and clock/timing integrated circuits.

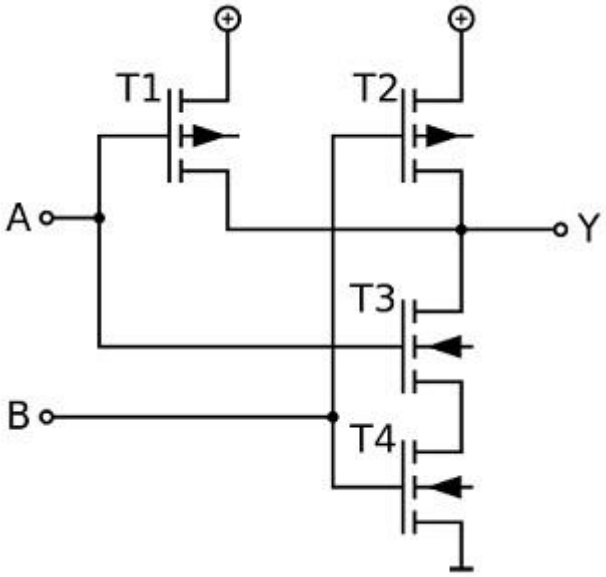
General types of integrated circuits are as following: Comparators, Switching IC, Audio amplifiers, Operational amplifiers, Timers ICs.

Digital Integrated Circuit

Functions of digital integrated circuits to handle discrete signals such as binary values in which “true/false” logical operations are used. Basic Boolean functions such as AND, OR, and NOT are essential in building functionality for modern

DIWAKAR EDUCATION HUB

digital systems. These Boolean functions are implemented using transistors. For example, internal transistor structure of a NAND gate is given in following figure.



There might ten billion or more transistors in modern digital circuit. So, we need Integrated Circuits (ICs) that combine a small or large number of these transistors to achieve particular functionality. These circuits provide benefiting students, very low cost and higher level of reliability. Examples of integrated circuits are MOS, CMOS, TTL etc. CMOS ICs are fault tolerant, reduce risk of chip failure, use of anti-static foam for storage and transport of ICs. TTL technology requires regulated power supply of 5 volts.

Families of Integrated Circuits

A logic family is a group of electronic logic gates. A family has its own discrete logic level, power supply, individual components, characteristic, advantages and disadvantages. Within each family, there is a range of voltage which may be low level or high level.

These families are listed as following below:

Diode Logic (DL)

DIWAKAR EDUCATION HUB

Registers and diodes are used to implement logic. The purpose of diode is to perform OR and AND operation, and logic switch. It is essential that diode is forward biased so that it can conduct. Disadvantages: diodes can not perform NOT operation, diode cannot work for multiple states, only one stage at a time, tend to degrade signals quickly.

Resistor-Transistor Logic (RTL)

Registers and transistors are used to implement logic. Transistors are used to combine signals which are inverted and amplified inputs. These are economical and easy to design but slower in speed. These require great amount of current from supply. RTL gates can be used as interface between linear circuits and digital.

Diode Transistor Logic (DTL)

Registers and diodes are used to implement logic. It has advantages over DL and RTL. Diode can perform AND and OR operation along with a transistor and output signal can be amplified. It has logic inversion which is signal can be restored to full logic level, if we add a transistor at output of logic gates. OR operation can be performed using diodes instead of resistors which removes interaction between input signals. Since input resistor to transistor, switching speed of transistor is limited. DTL was used in tube computers.

Transistor - Transistor Logic (TTL)

Transistors are used to implement logic and Bipolar transistors to contract integrated circuits. There are standard TTL, high speed TTL, low power TTL, and schottky TTL. It is popular logic family.

Emitter Coupled Logic (ECL)

ECL is a non-structured logic. Advantages of this logic are speed, and very low propagation delay.

Complementary Metal Oxide Semiconductor Logic (CMOS)

DIWAKAR EDUCATION HUB

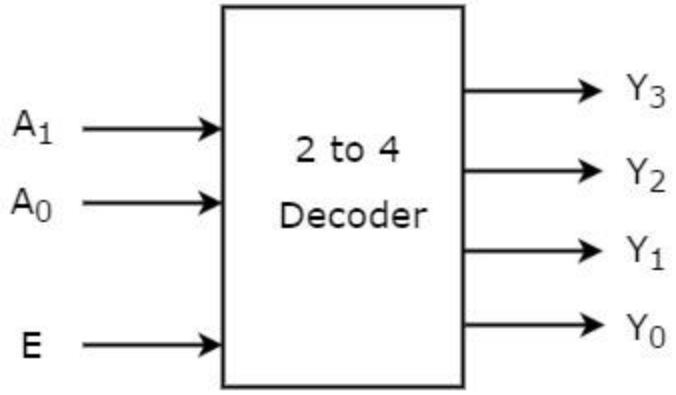
It is known for high fan-out, low power consumption and most reliable logic family.

Decoders

Decoder is a combinational circuit that has 'n' input lines and maximum of 2^n output lines. One of these outputs will be active High based on the combination of inputs present, when the decoder is enabled. That means decoder detects a particular code. The outputs of the decoder are nothing but the **min terms** of 'n' input variables lines, when it is enabled.

2 to 4 Decoder

Let 2 to 4 Decoder has two inputs A_1 & A_0 and four outputs Y_3, Y_2, Y_1 & Y_0.
The **block diagram** of 2 to 4 decoder is shown in the following figure.



One of these four outputs will be '1' for each combination of inputs when enable, E is '1'. The **Truth table** of 2 to 4 decoder is shown below.

Enable	Inputs		Outputs			
E	**A_1**	**A_0**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
0	x	x	0	0	0	0

1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

From Truth table, we can write the **Boolean functions** for each output as

Y3=E.A1.A0

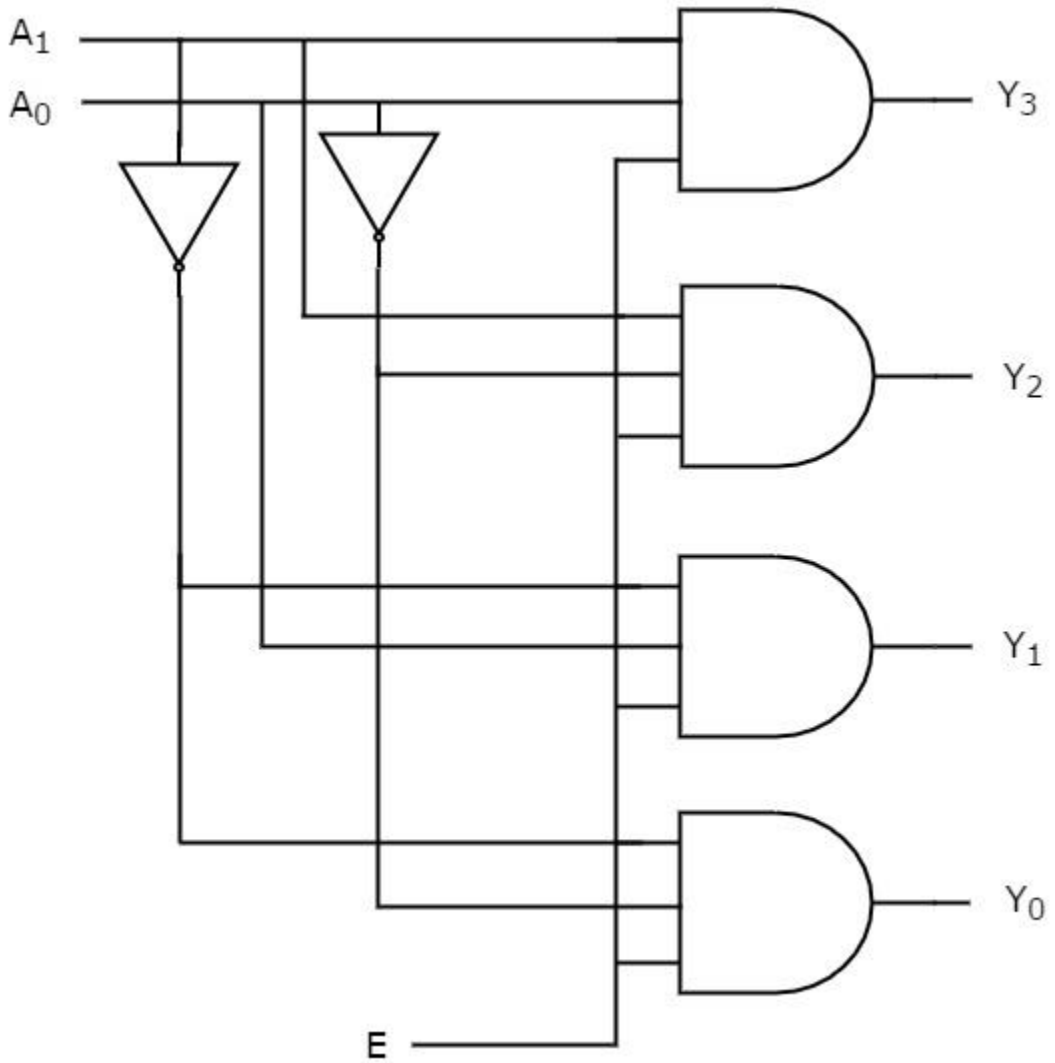
Y2=E.A1.A0'

Y1=E.A1'.A0

Y0=E.A1'.A0'

Each output is having one product term. So, there are four product terms in total. We can implement these four product terms by using four AND gates having three inputs each & two inverters. The **circuit diagram** of 2 to 4 decoder is shown in the following figure.

DIWAKAR EDUCATION HUB



Therefore, the outputs of 2 to 4 decoder are nothing but the **min terms** of two input variables A_1 & A_0, when enable, E is equal to one. If enable, E is zero, then all the outputs of decoder will be equal to zero.

Similarly, 3 to 8 decoder produces eight min terms of three input variables A_2, A_1 & A_0 and 4 to 16 decoder produces sixteen min terms of four input variables A_3, A_2, A_1 & A_0.

Implementation of Higher-order Decoders

Now, let us implement the following two higher-order decoders using lower-order decoders.

- 3 to 8 decoder
- 4 to 16 decoder

3 to 8 Decoder

In this section, let us implement **3 to 8 decoder using 2 to 4 decoders**. We know that 2 to 4 Decoder has two inputs, A_1 & A_0 and four outputs, Y_3 to Y_0. Whereas, 3 to 8 Decoder has three inputs A_2, A_1 & A_0 and eight outputs, Y_7 to Y_0.

We can find the number of lower order decoders required for implementing higher order decoder using the following formula.

Required number of lower order decoders =m2/m1 Required number of lower order decoders = m2/m1

Where,

m1/m1 is the number of outputs of lower order decoder.

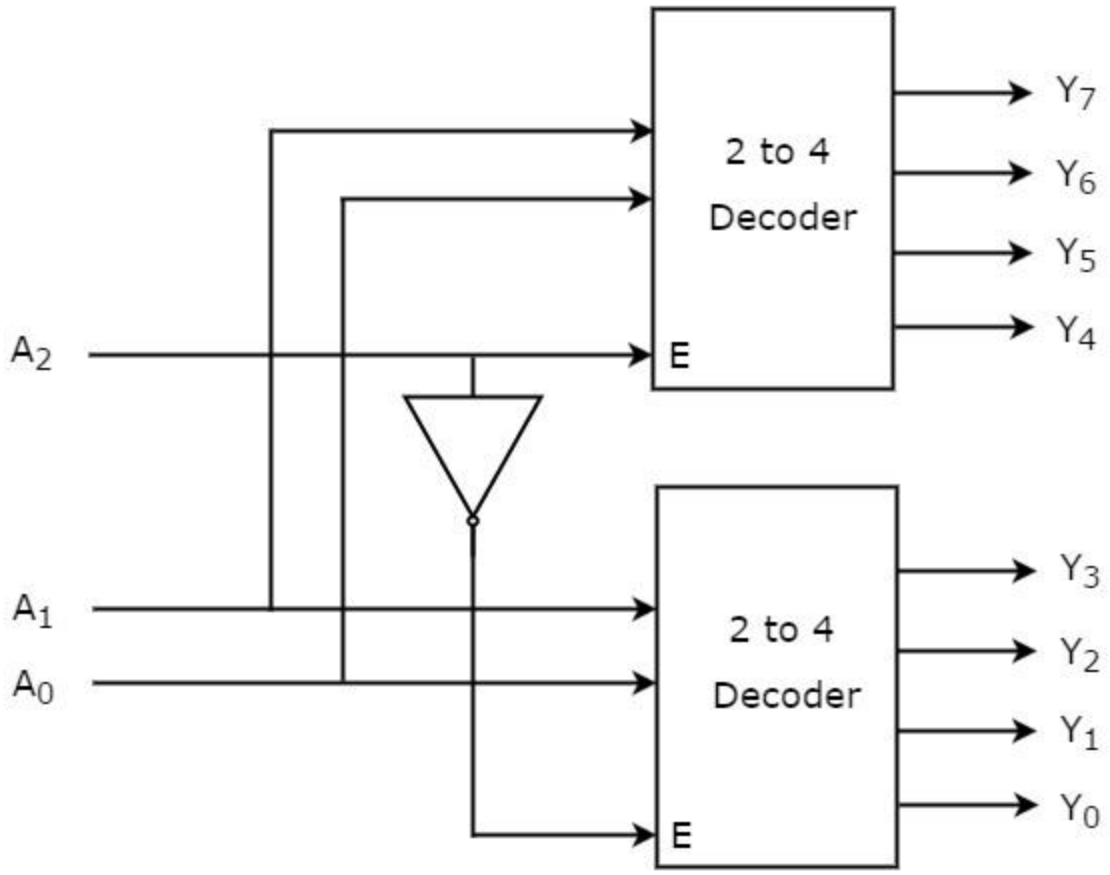
m2/m2 is the number of outputs of higher order decoder.

Here, m1 = 4 and m2 = 8. Substitute, these two values in the above formula.

Required number of 2 to 4 decoders = 8/4 =2 Required number of 2 to 4 decoders=8/4=2

Therefore, we require two 2 to 4 decoders for implementing one 3 to 8 decoder. The **block diagram** of 3 to 8 decoder using 2 to 4 decoders is shown in the following figure.

DIWAKAR EDUCATION HUB



The parallel inputs A_1 & A_0 are applied to each 2 to 4 decoder. The complement of input A_2 is connected to Enable, E of lower 2 to 4 decoder in order to get the outputs, Y_3 to Y_0. These are the **lower four min terms**. The input, A_2 is directly connected to Enable, E of upper 2 to 4 decoder in order to get the outputs, Y_7 to Y_4. These are the **higher four min terms**.

4 to 16 Decoder

In this section, let us implement **4 to 16 decoder using 3 to 8 decoders**. We know that 3 to 8 Decoder has three inputs A_2, A_1 & A_0 and eight outputs, Y_7 to Y_0. Whereas, 4 to 16 Decoder has four inputs A_3, A_2, A_1 & A_0 and sixteen outputs, Y_{15} to Y_0

We know the following formula for finding the number of lower order decoders required.

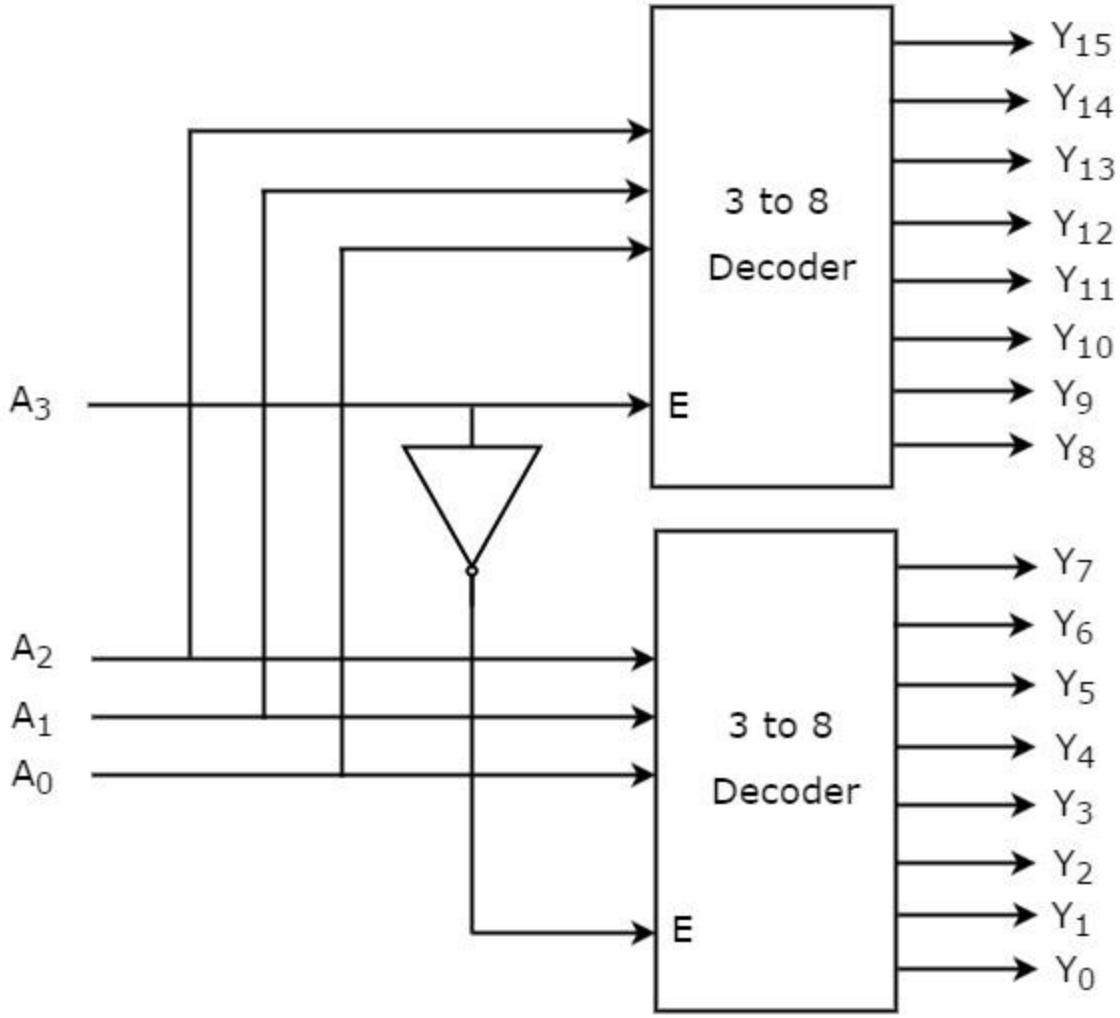
DIWAKAR EL

Required number of lower order decoders = m2/m1 Required number of lower order decoders = m2/m1

Substitute, m1 = 8 and m2 = 16 in the above formula.

Required number of 3 to 8 decoders = 16/8 = 2 Required number of 3 to 8 decoders = 16/8 =2

Therefore, we require two 3 to 8 decoders for implementing one 4 to 16 decoder. The **block diagram** of 4 to 16 decoder using 3 to 8 decoders is shown in the following figure.



The parallel inputs A_2, A_1 & A_0 are applied to each 3 to 8 decoder. The complement of input, A3 is connected to Enable, E of lower 3 to 8 decoder in order to get the outputs, Y_7 to Y_0. These are the **lower eight min terms**. The input, A_3 is directly connected to Enable, E of upper 3 to 8 decoder in order to get the outputs, Y_{15} to Y_8. These are the **higher eight min terms**.

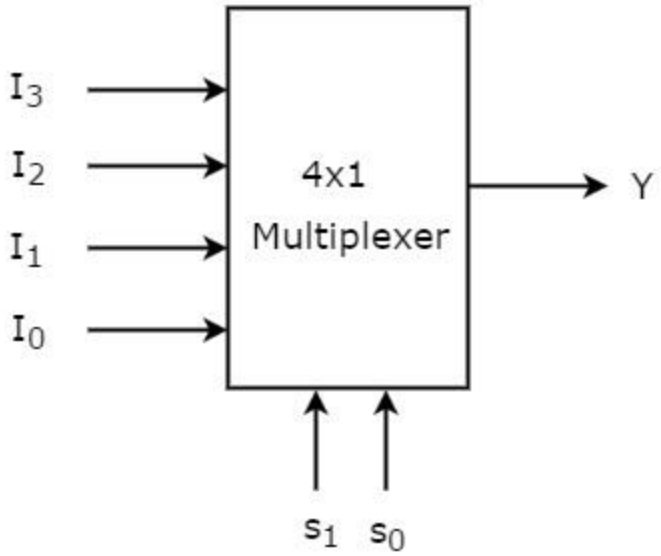
Multiplexers

Multiplexer is a combinational circuit that has maximum of 2^n data inputs, 'n' selection lines and single output line. One of these data inputs will be connected to the output based on the values of selection lines.

Since there are 'n' selection lines, there will be 2^n possible combinations of zeros and ones. So, each combination will select only one data input. Multiplexer is also called as **Mux**.

4x1 Multiplexer

4x1 Multiplexer has four data inputs I_3, I_2, I_1 & I_0, two selection lines s_1 & s_0 and one output Y. The **block diagram** of 4x1 Multiplexer is shown in the following figure.



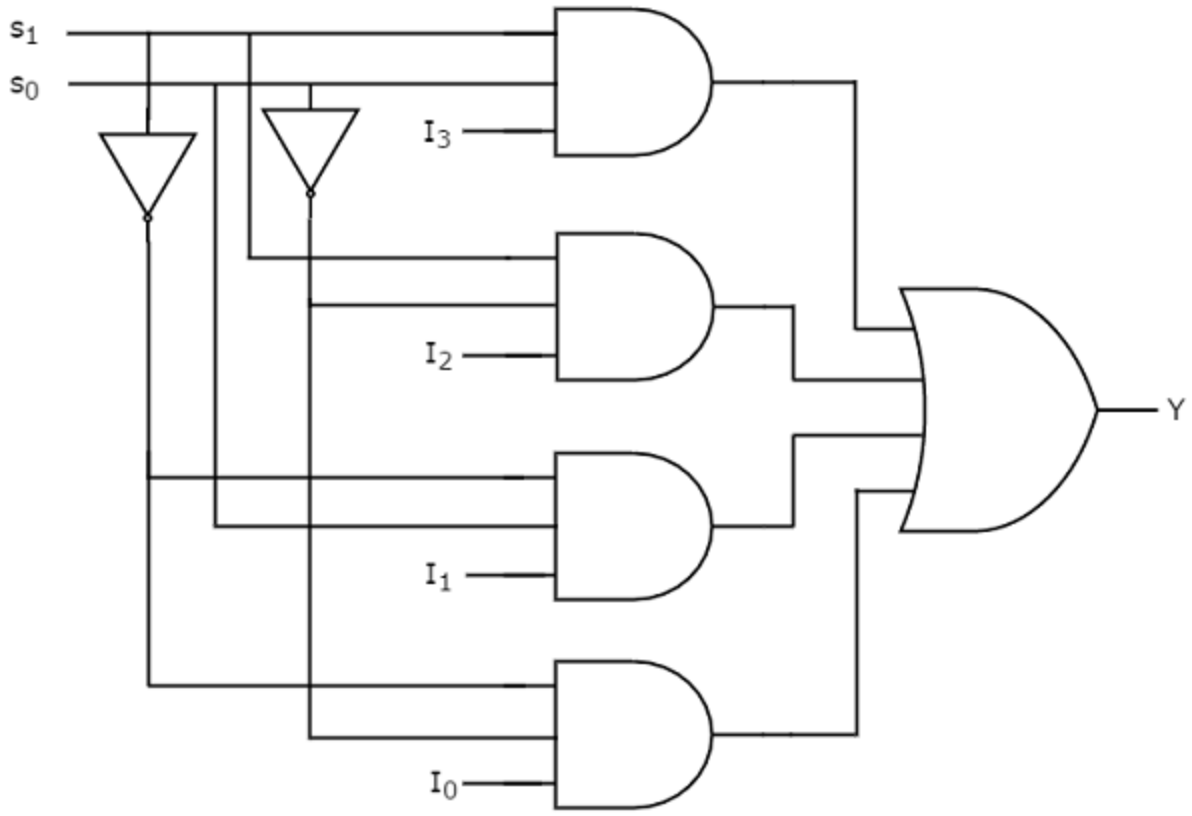
One of these 4 inputs will be connected to the output based on the combination of inputs present at these two selection lines. **Truth table** of 4x1 Multiplexer is shown below.

Selection Lines		Output
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

From Truth table, we can directly write the **Boolean function** for output, Y as

$$Y=S1'S0'I0+S1'S0I1+S1S0'I2+S1S0I3$$

We can implement this Boolean function using Inverters, AND gates & OR gate. The **circuit diagram** of 4x1 multiplexer is shown in the following figure.



We can easily understand the operation of the above circuit. Similarly, you can implement 8x1 Multiplexer and 16x1 multiplexer by following the same procedure.

Implementation of Higher-order Multiplexers.

Now, implement the following two higher-order Multiplexers using lower-order Multiplexers.

- 8x1 Multiplexer
- 16x1 Multiplexer

8x1 Multiplexer

In this section, implement 8x1 Multiplexer using 4x1 Multiplexers and 2x1 Multiplexer. We know that 4x1 Multiplexer has 4 data inputs, 2 selection lines and one output. Whereas, 8x1 Multiplexer has 8 data inputs, 3 selection lines and one output.

So, we require two **4x1 Multiplexers** in first stage in order to get the 8 data inputs. Since, each 4x1 Multiplexer produces one output, we require a **2x1 Multiplexer** in second stage by considering the outputs of first stage as inputs and to produce the final output.

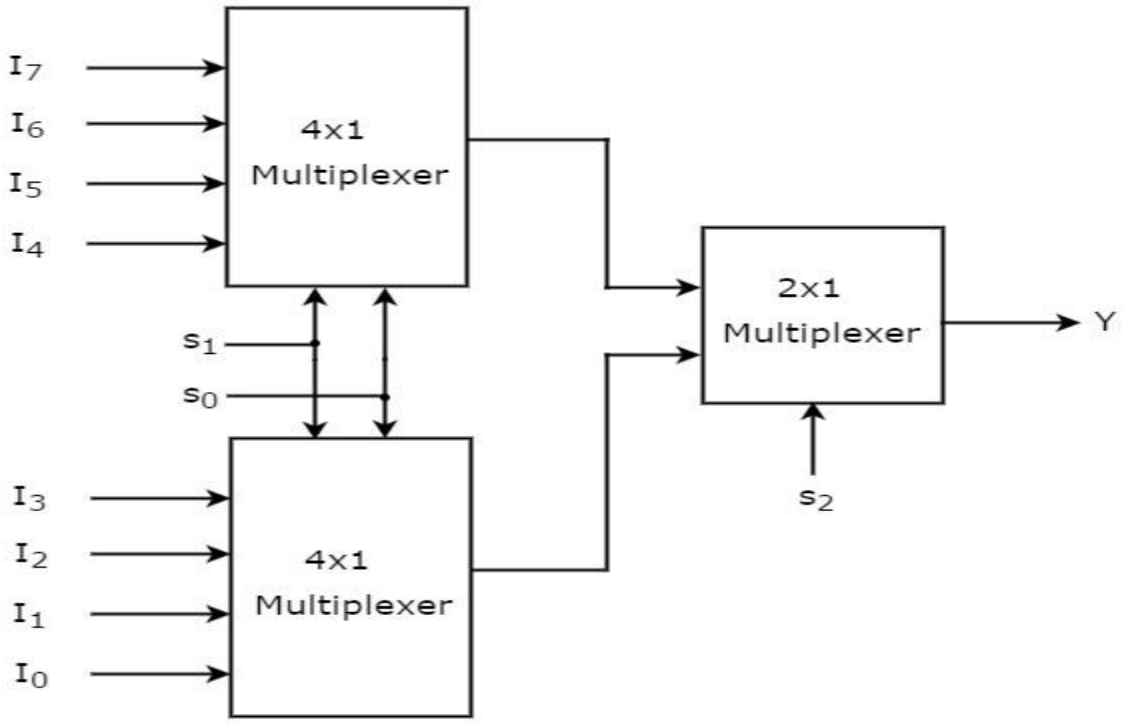
Let the 8x1 Multiplexer has eight data inputs I_7 to I_0, three selection lines s_2, s_1 & s0 and one output Y. The **Truth table** of 8x1 Multiplexer is shown below.

Selection Inputs			**Output**
S_2	**S_1**	**S_0**	**Y**
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3

DIWAKAR EDUCATION HUB

1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

We can implement 8x1 Multiplexer using lower order Multiplexers easily by considering the above Truth table. The **block diagram** of 8x1 Multiplexer is shown in the following figure.



The same **selection lines, s_1 & s_0** are applied to both 4x1 Multiplexers. The data inputs of upper 4x1 Multiplexer are I_7 to I_4 and the data inputs of lower 4x1 Multiplexer are I_3 to I_0. Therefore, each 4x1 Multiplexer produces an output based on the values of selection lines, s_1 & s_0.

The outputs of first stage 4x1 Multiplexers are applied as inputs of 2x1 Multiplexer that is present in second stage. The other **selection line, s_2** is applied to 2x1 Multiplexer.

- If s_2 is zero, then the output of 2x1 Multiplexer will be one of the 4 inputs I_3 to I_0 based on the values of selection lines s_1 & s_0.
- If s_2 is one, then the output of 2x1 Multiplexer will be one of the 4 inputs I_7 to I_4 based on the values of selection lines s_1 & s_0.

Therefore, the overall combination of two 4x1 Multiplexers and one 2x1 Multiplexer performs as one 8x1 Multiplexer.

16x1 Multiplexer

In this section, implement 16x1 Multiplexer using 8x1 Multiplexers and 2x1 Multiplexer. We know that 8x1 Multiplexer has 8 data inputs, 3 selection lines and one output. Whereas, 16x1 Multiplexer has 16 data inputs, 4 selection lines and one output.

So, we require two **8x1 Multiplexers** in first stage in order to get the 16 data inputs. Since, each 8x1 Multiplexer produces one output, we require a 2x1 Multiplexer in second stage by considering the outputs of first stage as inputs and to produce the final output.

Let the 16x1 Multiplexer has sixteen data inputs I_{15} to I_0, four selection lines s_3 to s_0 and one output Y. The **Truth table** of 16x1 Multiplexer is shown below.

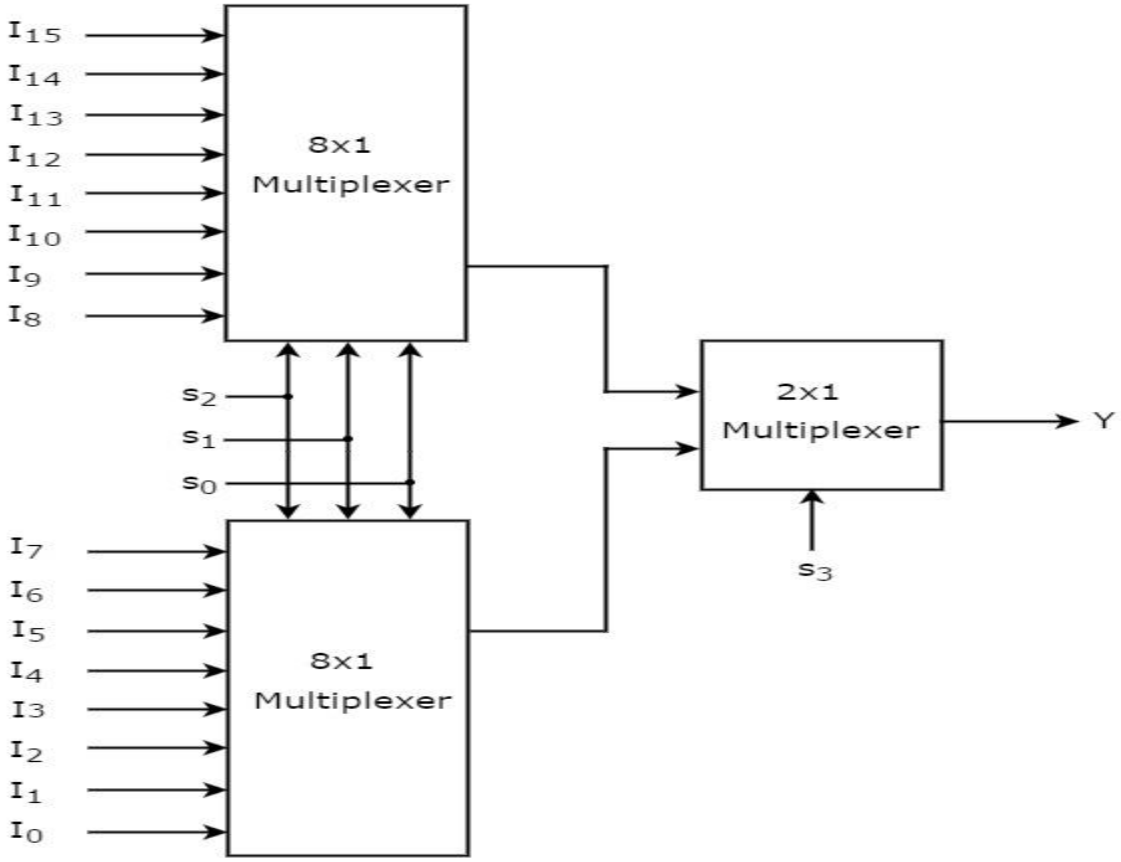
Selection Inputs				**Output**
S_3	**S_2**	**S_1**	**S_0**	**Y**
0	0	0	0	I_0
0	0	0	1	I_1
0	0	1	0	I_2
0	0	1	1	I_3
0	1	0	0	I_4
0	1	0	1	I_5

DIWAKAR EDUCATION HUB

0	1	1	0	I_6
0	1	1	1	I_7
1	0	0	0	I_8
1	0	0	1	I_9
1	0	1	0	I_{10}
1	0	1	1	I_{11}
1	1	0	0	I_{12}
1	1	0	1	I_{13}
1	1	1	0	I_{14}
1	1	1	1	I_{15}

We can implement 16x1 Multiplexer using lower order Multiplexers easily by considering the above Truth table. The **block diagram** of 16x1 Multiplexer is shown in the following figure.

DIWAKAR EDUCATION HUB



The **same selection lines, s_2, s_1 & s_0** are applied to both 8x1 Multiplexers. The data inputs of upper 8x1 Multiplexer are I_{15} to I_8 and the data inputs of lower 8x1 Multiplexer are I_7 to I_0. Therefore, each 8x1 Multiplexer produces an output based on the values of selection lines, s_2, s_1 & s_0.

The outputs of first stage 8x1 Multiplexers are applied as inputs of 2x1 Multiplexer that is present in second stage. The other **selection line, s_3** is applied to 2x1 Multiplexer.

- If s_3 is zero, then the output of 2x1 Multiplexer will be one of the 8 inputs Is_7 to I_0 based on the values of selection lines s_2, s_1 & s_0.
- If s_3 is one, then the output of 2x1 Multiplexer will be one of the 8 inputs I_{15} to I_8 based on the values of selection lines s_2, s_1 & s_0.

Therefore, the overall combination of two 8x1 Multiplexers and one 2x1 Multiplexer performs as one 16x1 Multiplexer.

Digital Registers

DIWAKAR EDUCATION HUB

Flip-flop is a 1 bit memory cell which can be used for storing the digital data. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flop. Such a group of flip-flop is known as a **Register**. The **n-bit register** will consist of **n** number of flip-flop and it is capable of storing an **n-bit** word.

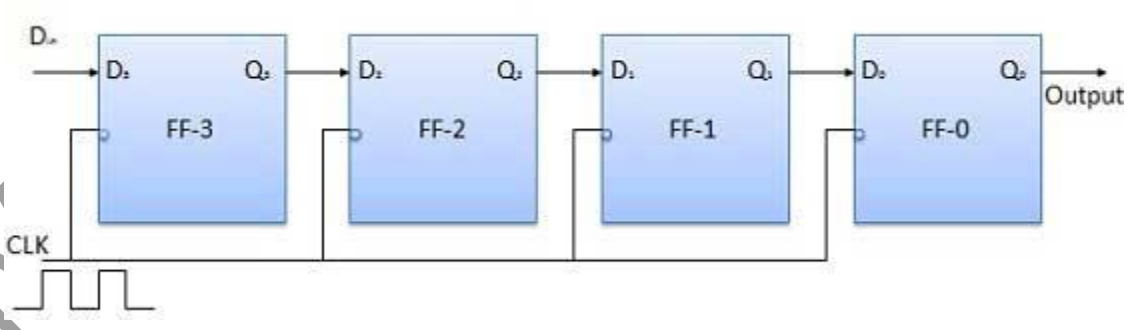
The binary data in a register can be moved within the register from one flip-flop to another. The registers that allow such data transfers are called as **shift registers**. There are four mode of operations of a shift register.

- Serial Input Serial Output
- Serial Input Parallel Output
- Parallel Input Serial Output
- Parallel Input Parallel Output

Serial Input Serial Output

Let all the flip-flop be initially in the reset condition i.e. $Q_3 = Q_2 = Q_1 = Q_0 = 0$. If an entry of a four bit binary number 1 1 1 1 is made into the register, this number should be applied to $\mathbf{D_{in}}$ bit with the LSB bit applied first. The D input of FF-3 i.e. D_3 is connected to serial data input $\mathbf{D_{in}}$. Output of FF-3 i.e. Q_3 is connected to the input of the next flip-flop i.e. D_2 and so on.

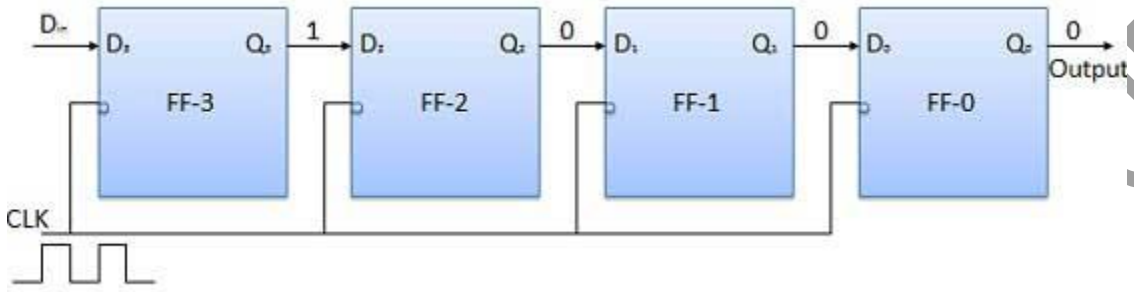
Block Diagram



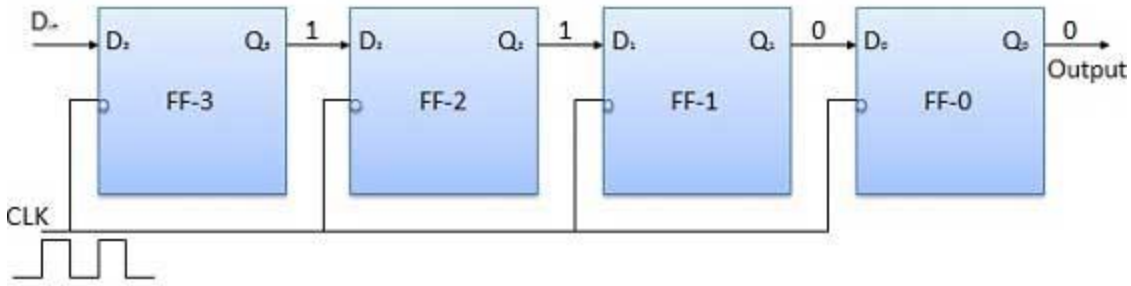
Operation

Before application of clock signal, let $Q_3\ Q_2\ Q_1\ Q_0 = 0000$ and apply LSB bit of the number to be entered to D_{in}. So $D_{in} = D_3 = 1$. Apply the clock. On the first falling

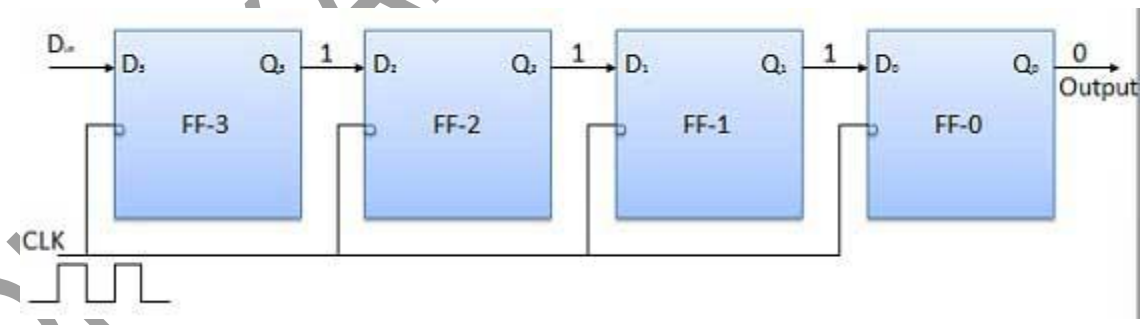
edge of clock, the FF-3 is set, and stored word in the register is $Q_3\ Q_2\ Q_1\ Q_0$ = 1000.



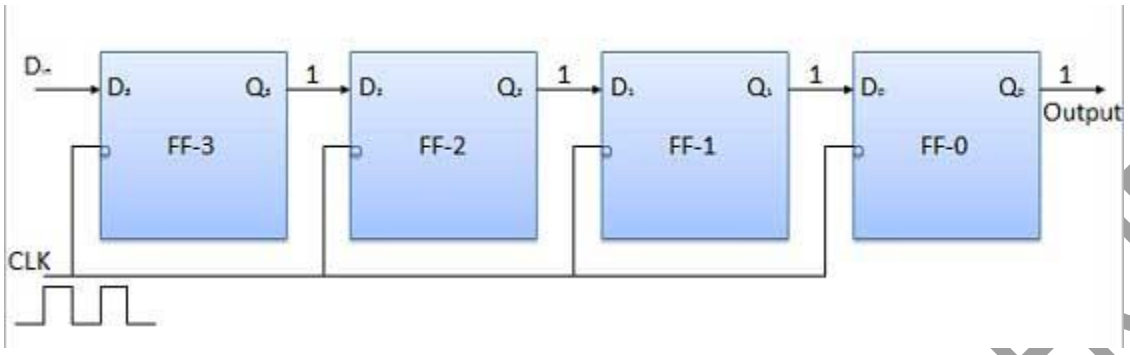
Apply the next bit to D_{in}. So D_{in} = 1. As soon as the next negative edge of the clock hits, FF-2 will set and the stored word change to $Q_3\ Q_2\ Q_1\ Q_0$ = 1100.



Apply the next bit to be stored i.e. 1 to D_{in}. Apply the clock pulse. As soon as the third negative clock edge hits, FF-1 will be set and output will be modified to $Q_3\ Q_2\ Q_1\ Q_0$ = 1110.



Similarly with D_{in} = 1 and with the fourth negative clock edge arriving, the stored word in the register is $Q_3\ Q_2\ Q_1\ Q_0$ = 1111.



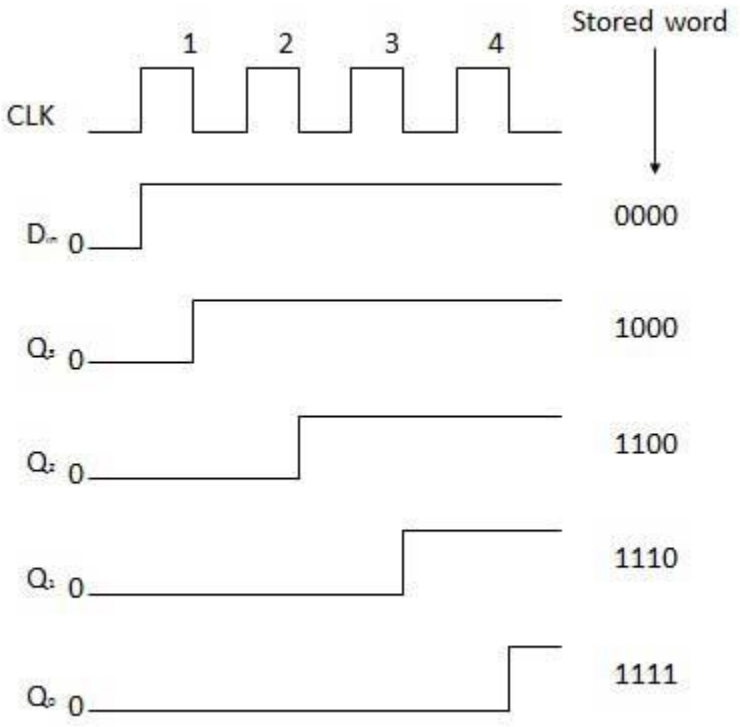
Truth Table

	CLK	$D_n = Q_3$	$Q_3 = D_2$	$Q_2 = D_1$	$Q_1 = D_0$	Q_0
Initially			0	0	0	0
(i)	↓	1	1	0	0	0
(ii)	↓	1	1	1	0	0
(iii)	↓	1	1	1	1	0
(iv)	↓	1	1	1	1	1

→ Direction of data travel

Waveforms

DIWAKAR EDUCATION HUB

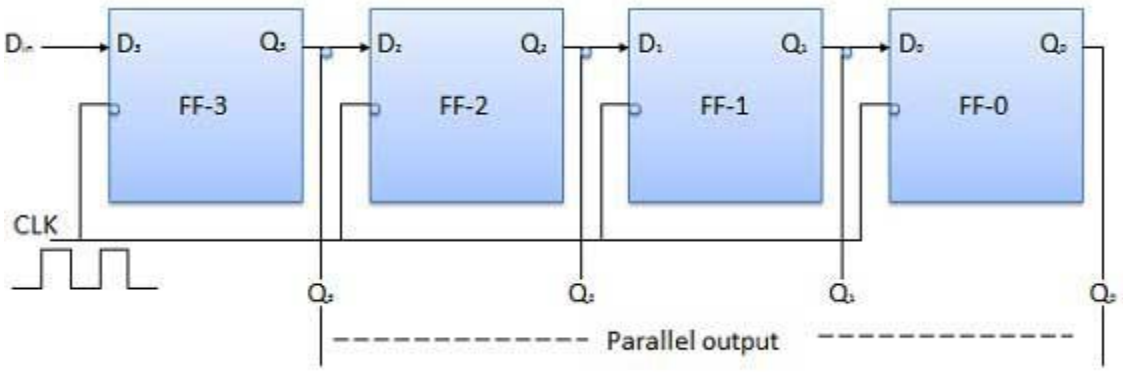


Serial Input Parallel Output

- In such types of operations, the data is entered serially and taken out in parallel fashion.
- Data is loaded bit by bit. The outputs are disabled as long as the data is loading.
- As soon as the data loading gets completed, all the flip-flops contain their required data, the outputs are enabled so that all the loaded data is made available over all the output lines at the same time.
- 4 clock cycles are required to load a four bit word. Hence the speed of operation of SIPO mode is same as that of SISO mode.

Block Diagram

DIWAKAR EDU HUB



Parallel Input Serial Output (PISO)

- Data bits are entered in parallel fashion.
- The circuit shown below is a four bit parallel input serial output register.
- Output of previous Flip Flop is connected to the input of the next one via a combinational circuit.
- The binary input word B_0, B_1, B_2, B_3 is applied though the same combinational circuit.
- There are two modes in which this circuit can work namely - shift mode or load mode.

Load mode

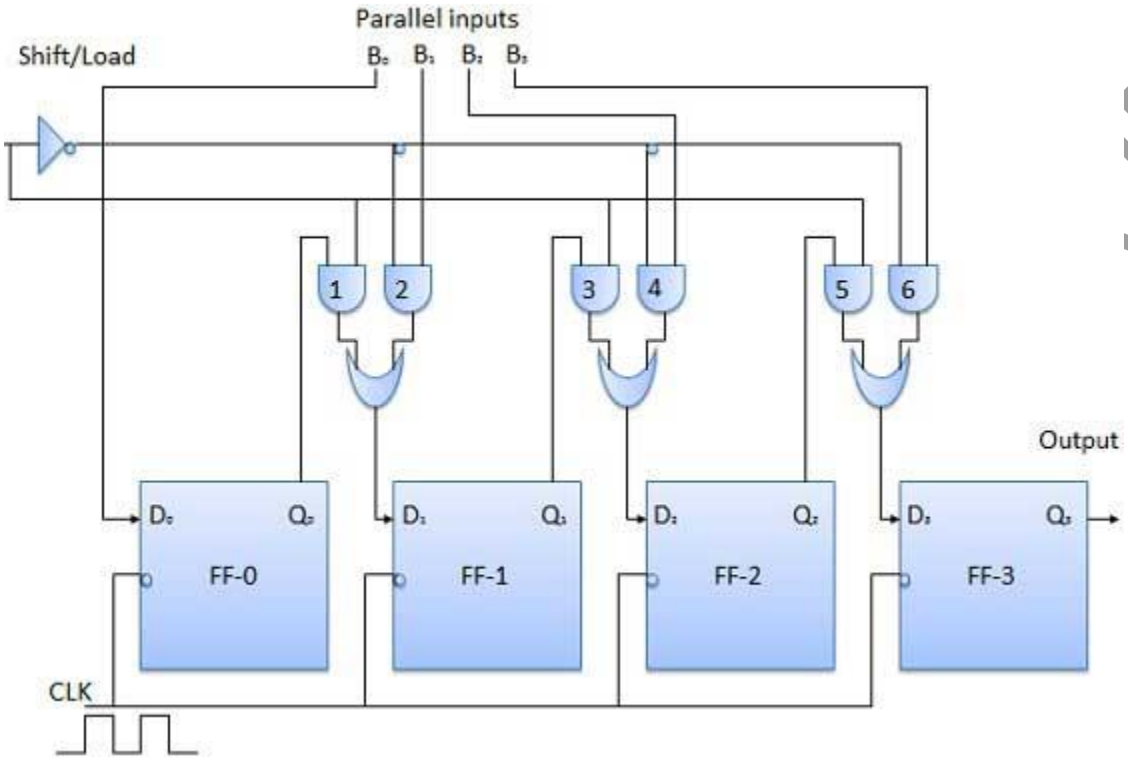
When the shift/load bar line is low (0), the AND gate 2, 4 and 6 become active they will pass B_1, B_2, B_3 bits to the corresponding flip-flops. On the low going edge of clock, the binary input B_0, B_1, B_2, B_3 will get loaded into the corresponding flip-flops. Thus parallel loading takes place.

Shift mode

When the shift/load bar line is low (1), the AND gate 2, 4 and 6 become inactive. Hence the parallel loading of the data becomes impossible. But the AND gate 1,3 and 5 become active. Therefore the shifting of data from left to right bit by bit on application of clock pulses. Thus the parallel in serial out operation takes place.

DIWAKAR EDUCATION HUB

Block Diagram

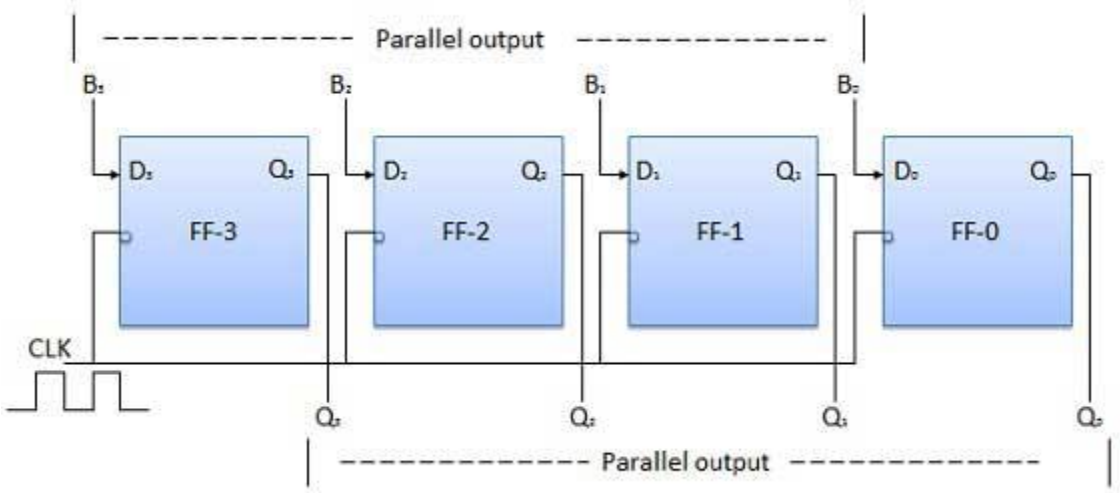


Parallel Input Parallel Output (PIPO)

In this mode, the 4 bit binary input B_0, B_1, B_2, B_3 is applied to the data inputs D_0, D_1, D_2, D_3 respectively of the four flip-flops. As soon as a negative clock edge is applied, the input binary bits will be loaded into the flip-flops simultaneously. The loaded bits will appear simultaneously to the output side. Only clock pulse is essential to load all the bits.

Block Diagram

DIWAKAR EDU

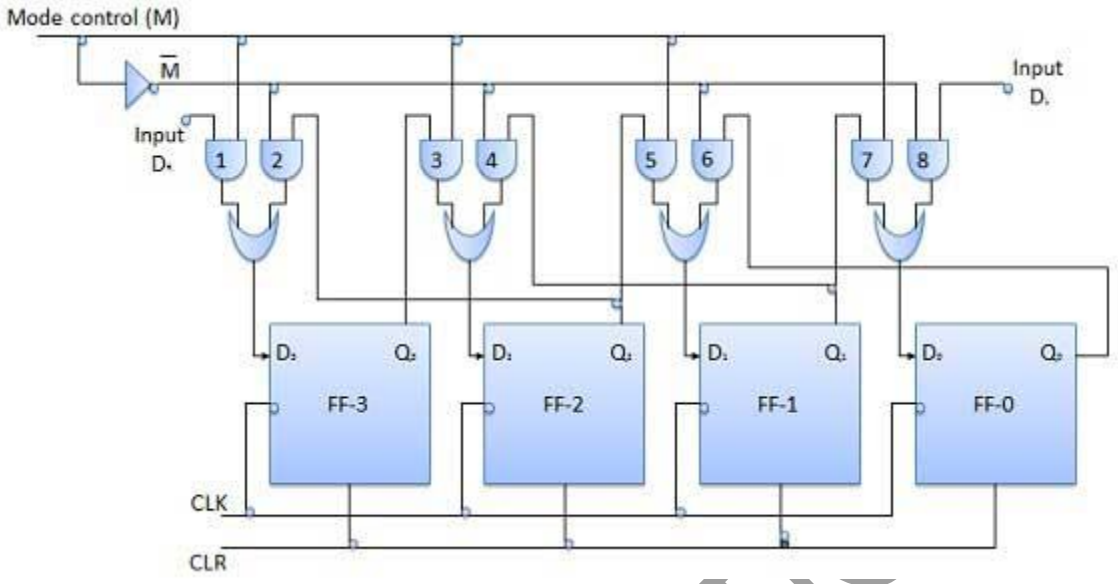


Bidirectional Shift Register

- If a binary number is shifted left by one position then it is equivalent to multiplying the original number by 2. Similarly if a binary number is shifted right by one position then it is equivalent to dividing the original number by 2.
- Hence if we want to use the shift register to multiply and divide the given binary number, then we should be able to move the data in either left or right direction.
- Such a register is called bi-directional register. A four bit bi-directional shift register is shown in fig.
- There are two serial inputs namely the serial right shift data input DR, and the serial left shift data input DL along with a mode select input (M).

Block Diagram

DIWAKAR EDUCATION HUB



Operation

S.N.	Condition	Operation
1	**With M = 1 – Shift right operation**	If M = 1, then the AND gates 1, 3, 5 and 7 are enabled whereas the remaining AND gates 2, 4, 6 and 8 will be disabled. The data at D_R is shifted to right bit by bit from FF-3 to FF-0 on the application of clock pulses. Thus with M = 1 we get the serial right shift operation.

DIWAKAR EDUCATION HUB

2	**With M = 0 – Shift left operation**	When the mode control M is connected to 0 then the AND gates 2, 4, 6 and 8 are enabled while 1, 3, 5 and 7 are disabled. The data at D_L is shifted left bit by bit from FF-0 to FF-3 on the application of clock pulses. Thus with M = 0 we get the serial right shift operation.

Universal Shift Register

A shift register which can shift the data in only one direction is called a uni-directional shift register. A shift register which can shift the data in both directions is called a bi-directional shift register. Applying the same logic, a shift register which can shift the data in both directions as well as load it parallely, is known as a universal shift register. The shift register is capable of performing the following operation −

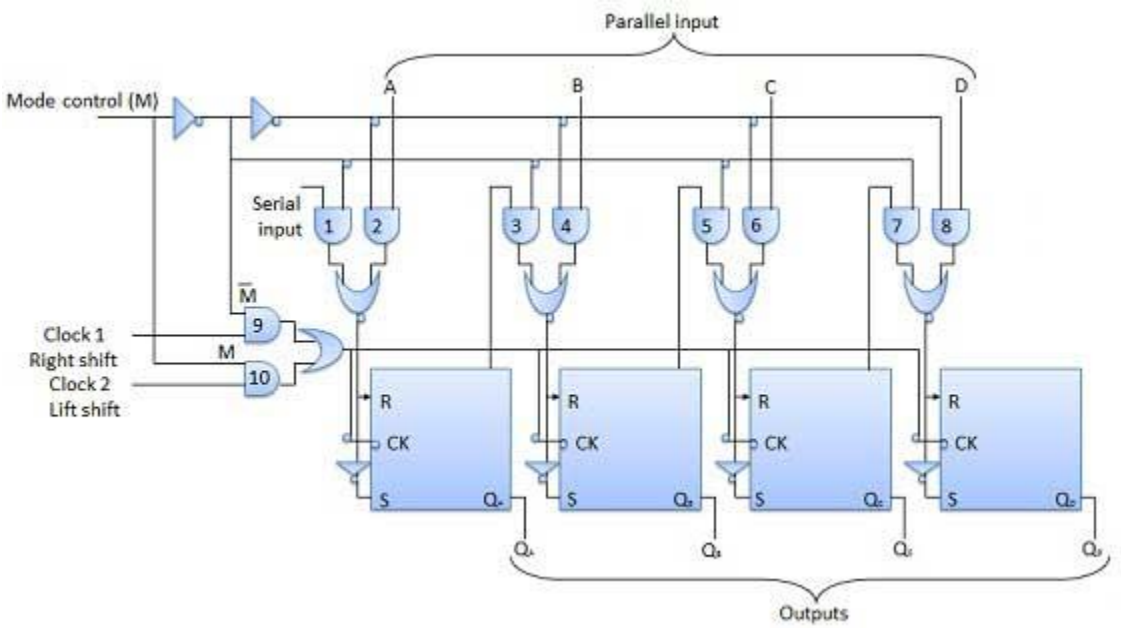
- Parallel loading
- Left Shifting
- Right shifting

The mode control input is connected to logic 1 for parallel loading operation whereas it is connected to 0 for serial shifting. With mode control pin connected

DIWAKAR EDUCATION HUB

to ground, the universal shift register acts as a bi-directional register. For serial left operation, the input is applied to the serial input which goes to AND gate-1 shown in figure. Whereas for the shift right operation, the serial input is applied to D input.

Block Diagram



Counter

in digital logic and computing, a **Counter** is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal. Counters are used in digital electronics for counting purpose, they can count specific event happening in the circuit. For example, in UP counter a counter increases count for every rising edge of clock. Not only counting, a counter can follow the certain sequence based on our design like any random sequence 0,1,3,2… .They can also be designed with the help of flip flops.

Counter Classification

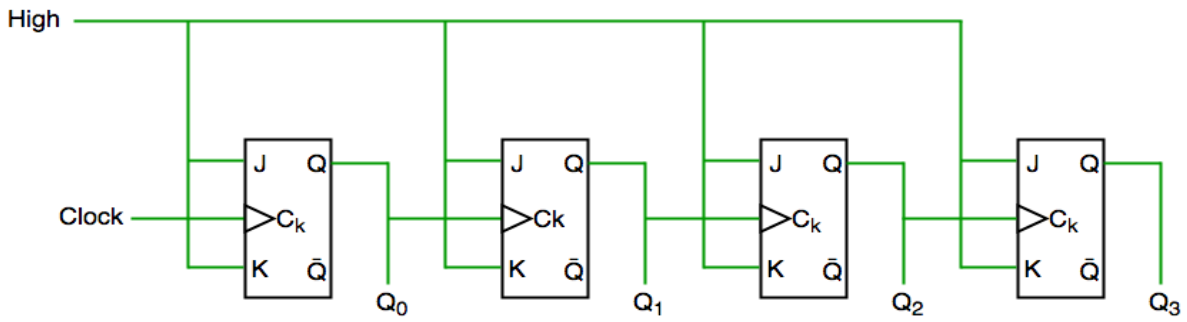
Counters are broadly divided into two categories

1. Asynchronous counter

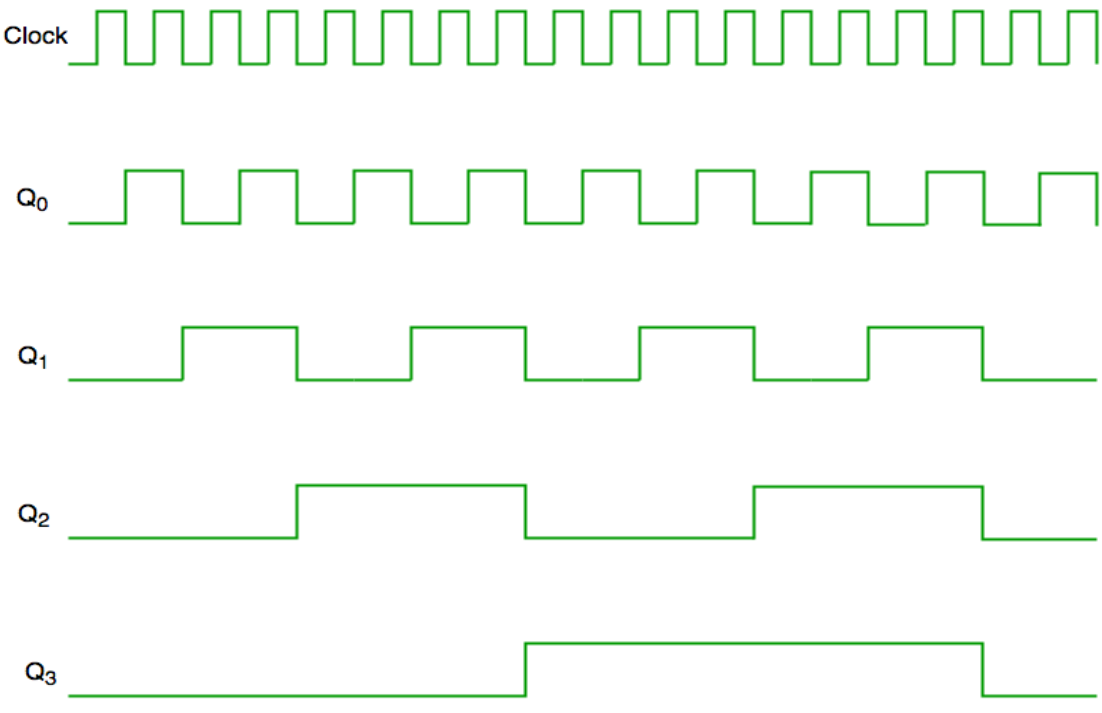
2. Synchronous counter

1. **Asynchronous Counter**

In asynchronous counter we don't use universal clock, only first flip flop is driven by main clock and the clock input of rest of the following flip flop is driven by output of previous flip flops. We can understand it by following diagram-



(a) Asynchronous counter

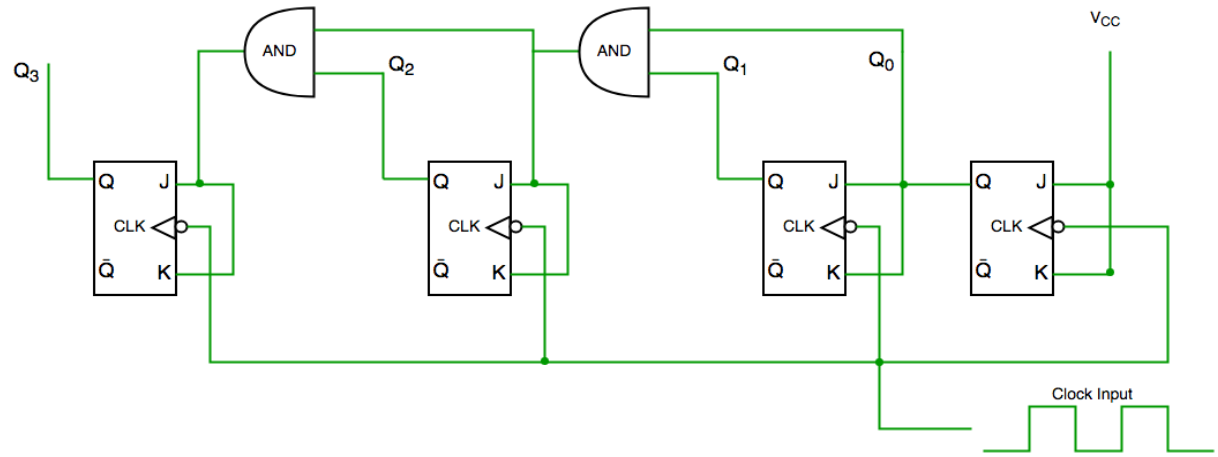


(b) Timing Diagram

It is evident from timing diagram that Q0 is changing as soon as the rising edge of clock pulse is encountered, Q1 is changing when rising edge of Q0 is encountered(because Q0 is like clock pulse for second flip flop) and so on. In this way ripples are generated through Q0,Q1,Q2,Q3 hence it is also called **RIPPLE counter.**

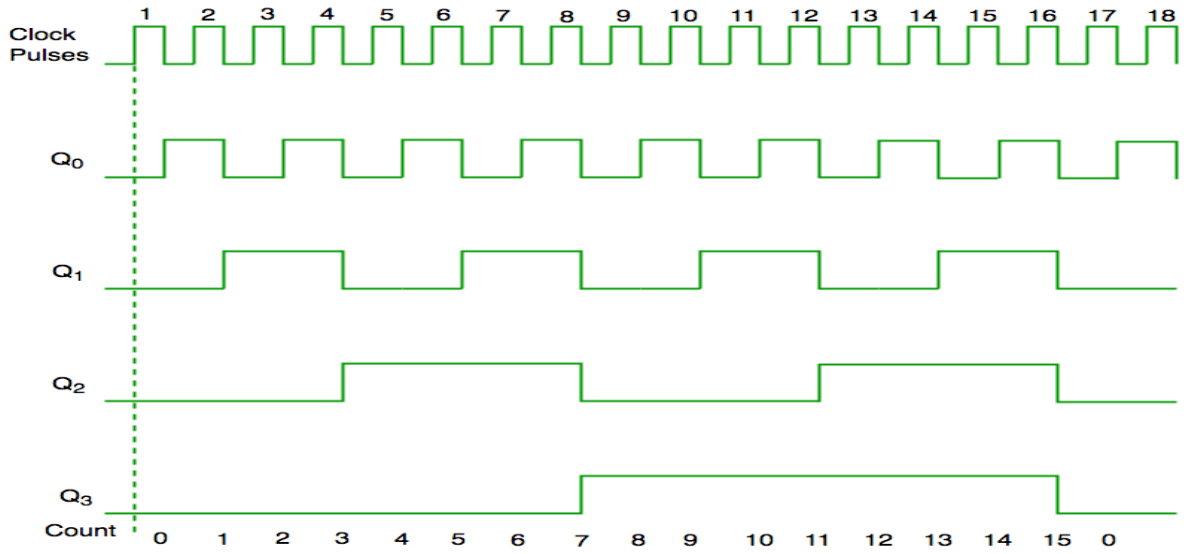
2. **Synchronous Counter**

Unlike the asynchronous counter, synchronous counter has one global clock which drives each flip flop so output changes in parallel. The one advantage of synchronous counter over asynchronous counter is, it can operate on higher frequency than asynchronous counter as it does not have cumulative delay because of same clock is given to each flip flop.



Synchronous counter circuit

DIWAKAR EDUCATION HUB



Timing diagram synchronous counter

From circuit diagram we see that Q0 bit gives response to each falling edge of clock while Q1 is dependent on Q0, Q2 is dependent on Q1 and Q0 , Q3 is dependent on Q2,Q1 and Q0.

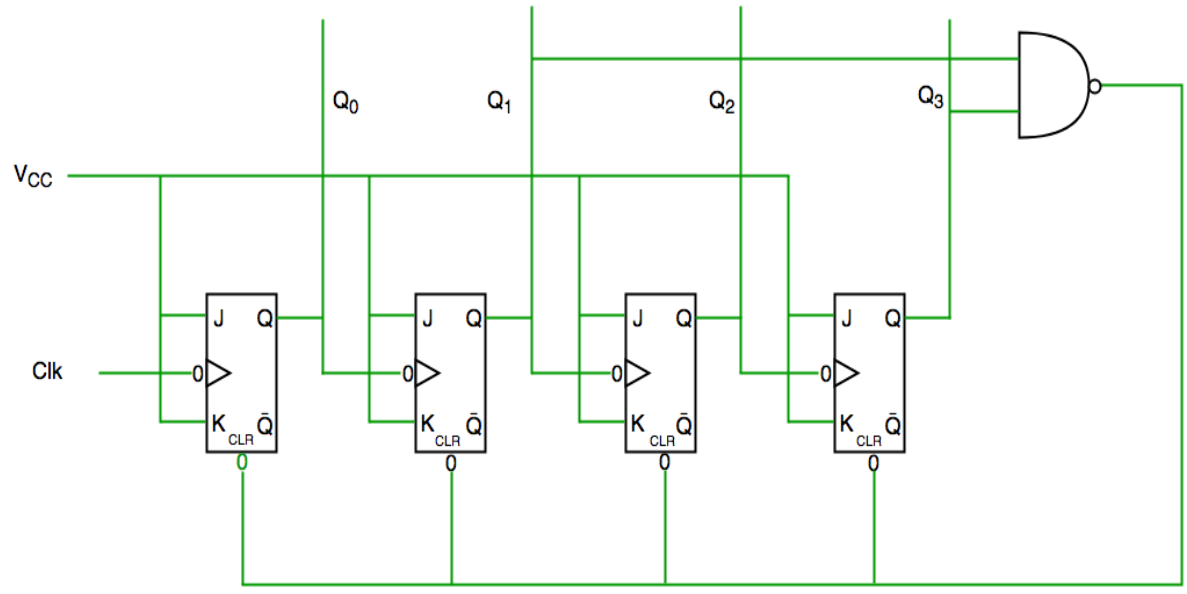
Decade Counter

A decade counter counts ten different states and then reset to its initial states. A simple decade counter will count from 0 to 9 but we can also make the decade counters which can go through any ten states between 0 to 15(for 4 bit counter).

Clock pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0

3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

Truth table for simple decade counter



Decade counter circuit diagram

We see from circuit diagram that we have used nand gate for Q3 and Q1 and feeding this to clear input line because binary representation of 10 is—1010

And we see Q3 and Q1 are 1 here, if we give NAND of these two bits to clear input then counter will be clear at 10 and again start from beginning.

Important point: Number of flip flops used in counter are always greater than equal to ($\log_2 n$) where n=number of states in counter.

Representation of Data

- Data and instructions cannot be entered and processed directly into computers using human language. Any type of data be it numbers, letters, special symbols, sound or pictures must first be converted into machine-readable form i.e. binary form. Due to this reason, it is important to understand how a computer together with its peripheral devices handles data in its electronic circuits, on magnetic media and in optical devices.

Data representation in digital circuits

- Electronic components, such as microprocessor, are made up of millions of electronic circuits. The availability of high voltage(on) in these circuits is interpreted as '1' while a low voltage (off) is interpreted as '0'.This concept can be compared to switching on and off an electric circuit. When the switch is closed the high voltage in the circuit causes the bulb to light ('1' state).on the other hand when the switch is open, the bulb goes off ('0' state). This forms a basis for describing data representation in digital computers using the binary number system.

Data representation on magnetic media

- The laser beam reflected from the land is interpreted, as 1.The laser entering the pot is not reflected. This is interpreted as 0.The reflected pattern of light from the rotating disk falls on a receiving photoelectric

DIWAKAR EDUCATION HUB

detector that transforms the patterns into digital form.The presence of a magnetic field in one direction on magnetic media is interpreted as 1; while the field in the opposite direction is interpreted as “0”.Magnetic technology is mostly used on storage devices that are coated with special magnetic materials such as iron oxide. Data is written on the media by arranging the magnetic dipoles of some iron oxide particles to face in the same direction and some others in the opposite direction

Data representation on optical media

In optical devices, the presence of light is interpreted as ‘1’ while its absence is interpreted as ‘0’.Optical devices use this technology to read or store data. Take example of a CD-ROM, if the shiny surface is placed under a powerful microscope, the surface is observed to have very tiny holes called **pits**. The areas that do not have pits are called **land**.

Computer does not understand human language. Any data, viz., letters, symbols, pictures, audio, videos, etc., fed to computer should be converted to machine language first. Computers represent data in the following three forms –

Number System

We are introduced to concept of numbers from a very early age. To a computer, everything is a number, i.e., alphabets, pictures, sounds, etc., are numbers. Number system is categorized into four types −

- Binary number system consists of only two values, either 0 or 1
- Octal number system represents values in 8 digits.
- Decimal number system represents values in 10 digits.
- Hexadecimal number system represents values in 16 digits.

Number System		
System	**Base**	**Digits**

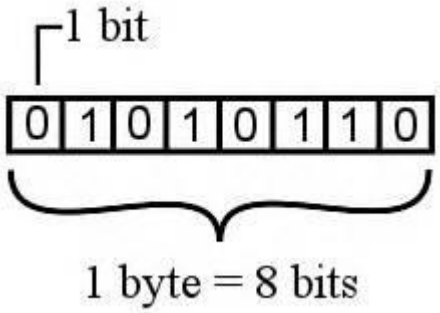
DIWAKAR EDUCATION HUB

Binary	2	0 1
Octal	8	0 1 2 3 4 5 6 7
Decimal	10	0 1 2 3 4 5 6 7 8 9
Hexadecimal	16	0 1 2 3 4 5 6 7 8 9 A B C D E F

Bits and Bytes

Bits – A bit is a smallest possible unit of data that a computer can recognize or use. Computer usually uses bits in groups.

Bytes – group of eight bits is called a byte. Half a byte is called a nibble.



The following table shows conversion of Bits and Bytes –

Byte Value	Bit Value
1 Byte	8 Bits
1024 Bytes	1 Kilobyte
1024 Kilobytes	1 Megabyte
1024 Megabytes	1 Gigabyte

DIWAKAR EDUCATION HUB

1024 Gigabytes	1 Terabyte
1024 Terabytes	1 Petabyte
1024 Petabytes	1 Exabyte
1024 Exabytes	1 Zettabyte
1024 Zettabytes	1 Yottabyte
1024 Yottabytes	1 Brontobyte
1024 Brontobytes	1 Geopbytes

Text Code

Text code is format used commonly to represent alphabets, punctuation marks and other symbols. Four most popular text code systems are −

- EBCDIC
- ASCII
- Extended ASCII
- Unicode

EBCDIC

Extended Binary Coded Decimal Interchange Code is an 8-bit code that defines 256 symbols. Given below is the EBCDIC **Tabular column**

DIWAKAR EDUCATION HUB

Special characters	EBCDIC
<	01001011
(	01001100
+	01001101
/	01001110
&	01010000
:	01111011
#	01111011
@	01111100
‘	01111101
=	01111110
“	01111111
,	01101011
%	01101100
-	01101101
>	01101110

Alphabetic	EBCDIC
A	11000001
B	11000010
C	11000011
D	11000100
E	11000101
F	11000110
G	11000111
H	11001000
I	11001001
J	11010001
K	11010010
L	11010011
M	11010100
N	11010101
O	11010110
P	11010111

ASCII

American Standard Code for Information Interchange is an 8-bit code that specifies character values from 0 to 127.

ASCII Tabular column

ASCII Code	Decimal Value	Character
0000 0000	0	Null prompt
0000 0001	1	Start of heading
0000 0010	2	Start of text

0000 0011	3	End of text
0000 0100	4	End of transmit
0000 0101	5	Enquiry
0000 0110	6	Acknowledge
0000 0111	7	Audible bell
0000 1000	8	Backspace
0000 1001	9	Horizontal tab
0000 1010	10	Line Feed

Extended ASCII

Extended American Standard Code for Information Interchange is an 8-bit code that specifies character values from 128 to 255.

Extended ASCII Tabular column

Char	Code	Char	Code	Char	Code	Char
Ą	161	˘	162	Ł	163	¤
Š	169	Ş	170	Ť	171	Ź
ą	177	˛	178	ł	179	´
š	185	ş	186	t'	187	ź
-	193	Â	194	Ă	195	Ä

Unicode

DIWAKAR EDUCATION HUB

Unicode Worldwide Character Standard uses 4 to 32 bits to represent letters, numbers and symbol.

Unicode Tabular Column

Char	UTF-16	UTF-8
A	0041	41
c	0063	63
ö	00F6	C3 86
亜	4E9C	E4 BA 9C
𝄞	D834 DD1E	F0 9D 84 9E

Data Types

A very simple but very important concept available in almost all the programming languages which is called **data types**. As its name indicates, a data type represents a type of the data which you can process using your computer program. It can be numeric, alphanumeric, decimal, etc.

Keep Computer Programming aside for a while and take an easy example of adding two whole numbers 10 & 20, which can be done simply as follows −

10 + 20

Another problem where we want to add two decimal numbers 10.50 & 20.50, which will be written as follows −

10.50 + 20.50

The two examples are straightforward. Now another example where we want to record student information in a notebook. Here we would like to record the following information −

Name:

DIWAKAR EDUCATION HUB

Class:

Section:

Age:

Sex:

Now, put one student record as per the given requirement –

Name: Zara Ali

Class: 6th

Section: J

Age: 13

Sex: F

The first example dealt with whole numbers, the second example added two decimal numbers, whereas the third example is dealing with a mix of different data. put it as follows –

- Student name "Zara Ali" is a sequence of characters which is also called a string.
- Student class "6th" has been represented by a mix of whole number and a string of two characters. Such a mix is called alphanumeric.
- Student section has been represented by a single character which is 'J'.
- Student age has been represented by a whole number which is 13.
- Student sex has been represented by a single character which is 'F'.

This way, we realized that in our day-to-day life, we deal with different types of data such as strings, characters, whole numbers (integers), and decimal numbers (floating point numbers).

Similarly, when we write a computer program to process different types of data, we need to specify its type clearly; otherwise the computer does not understand

DIWAKAR EDUCATION HUB

how different operations can be performed on that given data. Different programming languages use different keywords to specify different data types. For example, C and Java programming languages use **int** to specify integer data, whereas **char** specifies a character data type.

Subsequent chapters will show you how to use different data types in different situations. For now, check the important data types available in C, Java, and Python and the keywords we will use to specify those data types.

C and Java Data Types

C and Java support almost the same set of data types, though Java supports additional data types. For now, we are taking a few common data types supported by both the programming languages −

Type	Keyword	Value range which can be represented by this data type
Character	char	-128 to 127 or 0 to 255
Number	int	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
Small Number	short	-32,768 to 32,767
Long Number	long	-2,147,483,648 to 2,147,483,647
Decimal Number	float	1.2E-38 to 3.4E+38 till 6 decimal places

DIWAKAR EDUCATION HUB

These data types are called primitive data types and you can use these data types to build more complex data types, which are called user-defined data type, for example a string will be a sequence of characters.

Python Data Types

Python has five standard data types but this programming language does not make use of any keyword to specify a particular data type, rather Python is intelligent enough to understand a given data type automatically.

- Numbers
- String
- List
- Tuple
- Dictionary

Here, Number specifies all types of numbers including decimal numbers and string represents a sequence of characters with a length of 1 or more characters. For now, proceed with these two data types and skip List, Tuple, and Dictionary, which are advanced data types in Python.

Number System

The technique to represent and work with numbers is called **number system**. **Decimal number system** is the most common number system. Other popular number systems include **binary number system, octal number system, hexadecimal number system,** etc.

Decimal Number System

Decimal number system is a **base 10** number system having 10 digits from 0 to 9. This means that any numerical quantity can be represented using these 10 digits. Decimal number system is also a **positional value system**. This means that the value of digits will depend on its position. Let us take an example to understand this.

DIWAKAR EDUCATION HUB

Say we have three numbers – 734, 971 and 207. The value of 7 in all three numbers is different−

- In 734, value of 7 is 7 hundreds or 700 or 7 × 100 or 7 × 10^2
- In 971, value of 7 is 7 tens or 70 or 7 × 10 or 7 × 10^1
- In 207, value 0f 7 is 7 units or 7 or 7 × 1 or 7 × 10^0

The weightage of each position can be represented as follows −

10^5	10^4	10^3	10^2	10^1	10^0

In digital systems, instructions are given through electric signals; variation is done by varying the voltage of the signal. Having 10 different voltages to implement decimal number system in digital equipment is difficult. So, many number systems that are easier to implement digitally have been developed. Let's look at them in detail.

Binary Number System

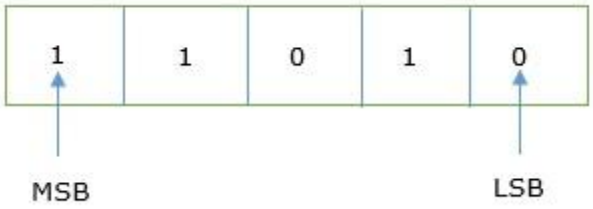
The easiest way to vary instructions through electric signals is two-state system – on and off. On is represented as 1 and off as 0, though 0 is not actually no signal but signal at a lower voltage. The number system having just these two digits – 0 and 1 – is called **binary number system**.

Each binary digit is also called a **bit**. Binary number system is also positional value system, where each digit has a value expressed in powers of 2, as displayed here.

2^5	2^4	2^3	2^2	2^1	2^0

In any binary number, the rightmost digit is called **least significant bit (LSB)** and leftmost digit is called **most significant bit (MSB)**.

DIWAKAR EDUCATION HUB



And decimal equivalent of this number is sum of product of each digit with its positional value.

$11010_2 = 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$

$= 16 + 8 + 0 + 2 + 0$

$= 26_{10}$

Computer memory is measured in terms of how many bits it can store. Here is a chart for memory capacity conversion.

- 1 byte (B) = 8 bits
- 1 Kilobytes (KB) = 1024 bytes
- 1 Megabyte (MB) = 1024 KB
- 1 Gigabyte (GB) = 1024 MB
- 1 Terabyte (TB) = 1024 GB
- 1 Exabyte (EB) = 1024 PB
- 1 Zettabyte = 1024 EB
- 1 Yottabyte (YB) = 1024 ZB

Octal Number System

Octal number system has eight digits – 0, 1, 2, 3, 4, 5, 6 and 7. Octal number system is also a positional value system with where each digit has its value expressed in powers of 8, as shown here –



DIWAKAR EDUCATION HUB

Decimal equivalent of any octal number is sum of product of each digit with its positional value.

$726_8 = 7\times8^2 + 2\times8^1 + 6\times8^0$

$= 448 + 16 + 6$

$= 470_{10}$

Hexadecimal Number System

Octal number system has 16 symbols – 0 to 9 and A to F where A is equal to 10, B is equal to 11 and so on till F. Hexadecimal number system is also a positional value system with where each digit has its value expressed in powers of 16, as shown here −

16^5	16^4	16^3	16^2	16^1	16^0

Decimal equivalent of any hexadecimal number is sum of product of each digit with its positional value.

$27FB_{16} = 2\times16^3 + 7\times16^2 + 15\times16^1 + 10\times16^0$

$= 8192 + 1792 + 240 + 10$

$= 10234_{10}$

Number System Relationship

The following table depicts the relationship between decimal, binary, octal and hexadecimal number systems.

HEXADECIMAL	DECIMAL	OCTAL	BINARY
0	0	0	0000

DIWAKAR EDUCATION HUB

1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	8	10	1000
9	9	11	1001
A	10	12	1010
B	11	13	1011
C	12	14	1100
D	13	15	1101
E	14	16	1110

DIWAKAR EDUCATION HUB

F	15	17	1111

ASCII

Besides numerical data, computer must be able to handle alphabets, punctuation marks, mathematical operators, special symbols, etc. that form the complete character set of English language. The complete set of characters or symbols are called alphanumeric codes. The complete alphanumeric code typically includes –

- 26 upper case letters
- 26 lower case letters
- 10 digits
- 7 punctuation marks
- 20 to 40 special characters

Now a computer understands only numeric values, whatever the number system used. So all characters must have a numeric equivalent called the alphanumeric code. The most widely used alphanumeric code is American Standard Code for Information Interchange (ASCII). ASCII is a 7-bit code that has 128 (27) possible codes.

DIWAKAR EDUCATION HUB

ASCII Code - Character to Binary

0	0011 0000	I	0100 1001	b	0110 0010	v	0111 0110
1	0011 0001	J	0100 1010	c	0110 0011	w	0111 0111
2	0011 0010	K	0100 1011	d	0110 0100	x	0111 1000
3	0011 0011	L	0100 1100	e	0110 0101	y	0111 1001
4	0011 0100	M	0100 1101	f	0110 0110	z	0111 1010
5	0011 0101	N	0100 1110	g	0110 0110		
6	0011 0110	O	0100 1111	h	0110 1000	:	0011 1010
7	0011 0110	P	0101 0000	i	0110 1001	;	0011 1011
8	0011 1000	Q	0101 0001	j	0110 1010	?	0011 1111
9	0011 1001	R	0101 0010	k	0110 1011	·	0010 1110
		S	0101 0011	l	0110 1100	'	0010 1111
		T	0101 0100	m	0110 1101	!	0010 0001
A	0100 0001	U	0101 0101	n	0110 1110	'	0010 1100
B	0100 0010	V	0101 0110	o	0110 1111	"	0010 0010
C	0100 0011	W	0101 0111	p	0111 0000	(	0010 1000
D	0100 0100	X	0101 1000	q	0111 0001	)	0010 1001
E	0100 0101	Y	0101 1001	r	0111 0010	space	0010 0000
F	0100 0110	Z	0101 1010	s	0111 0011		
G	0100 0111			t	0111 0100		
H	0100 1000	a	0110 0001	u	0111 0101		

ISCII

ISCII stands for **Indian Script Code for Information Interchange**. IISCII was developed to support Indian languages on computer. Language supported by IISCI include Devanagari, Tamil, Bangla, Gujarati, Gurmukhi, Tamil, Telugu, etc. IISCI is mostly used by government departments and before it could catch on, a new universal encoding standard called **Unicode** was introduced.

Unicode

Unicode is an international coding system designed to be used with different language scripts. Each character or symbol is assigned a unique numeric value, largely within the framework of ASCII. Earlier, each script had its own encoding system, which could conflict with each other.

In contrast, this is what Unicode officially aims to do − Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language.

Number System Conversion

There are many methods or techniques which can be used to convert numbers from one base to another. Here the following −

- Decimal to Other Base System
- Other Base System to Decimal
- Other Base System to Non-Decimal
- Shortcut method − Binary to Octal
- Shortcut method − Octal to Binary
- Shortcut method − Binary to Hexadecimal
- Shortcut method − Hexadecimal to Binary

Decimal to Other Base System

Steps

- **Step 1** − Divide the decimal number to be converted by the value of the new base.

DIWAKAR EDUCATION HUB

- **Step 2** − Get the remainder from Step 1 as the rightmost digit (least significant digit) of new base number.
- **Step 3** − Divide the quotient of the previous divide by the new base.
- **Step 4** − Record the remainder from Step 3 as the next digit (to the left) of the new base number.

Repeat Steps 3 and 4, getting remainders from right to left, until the quotient becomes zero in Step 3.

The last remainder thus obtained will be the Most Significant Digit (MSD) of the new base number.

Example –

Decimal Number: 29_{10}

Calculating Binary Equivalent –

Step	Operation	Result	Remainder
Step 1	29 / 2	14	1
Step 2	14 / 2	7	0
Step 3	7 / 2	3	1
Step 4	3 / 2	1	1
Step 5	1 / 2	0	1

As mentioned in Steps 2 and 4, the remainders have to be arranged in the reverse order so that the first remainder becomes the Least Significant Digit (LSD) and the last remainder becomes the Most Significant Digit (MSD).

DIWAKAR EDUCATION HUB

Decimal Number − 29_{10} = Binary Number − 11101_2.

Other Base System to Decimal System

Steps

- **Step 1** − Determine the column (positional) value of each digit (this depends on the position of the digit and the base of the number system).
- **Step 2** − Multiply the obtained column values (in Step 1) by the digits in the corresponding columns.
- **Step 3** − Sum the products calculated in Step 2. The total is the equivalent value in decimal.

Example

Binary Number − 11101_2

Calculating Decimal Equivalent −

Step	Binary Number	Decimal Number
Step 1	11101_2	$((1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0))_{10}$
Step 2	11101_2	$(16 + 8 + 4 + 0 + 1)_{10}$
Step 3	11101_2	29_{10}

Binary Number − 11101_2 = Decimal Number − 29_{10}

Other Base System to Non-Decimal System

Steps

- **Step 1** − Convert the original number to a decimal number (base 10).

DIWAKAR EDUCATION HUB

- **Step 2** – Convert the decimal number so obtained to the new base number.

Example

Octal Number – 25_8

Calculating Binary Equivalent –

Step 1 – Convert to Decimal

Step	Octal Number	Decimal Number
Step 1	25_8	$((2 \times 8^1) + (5 \times 8^0))_{10}$
Step 2	25_8	$(16 + 5)_{10}$
Step 3	25_8	21_{10}

Octal Number – 25_8 = Decimal Number – 21_{10}

Step 2 – Convert Decimal to Binary

Step	Operation	Result	Remainder
Step 1	21 / 2	10	1
Step 2	10 / 2	5	0
Step 3	5 / 2	2	1
Step 4	2 / 2	1	0
Step 5	1 / 2	0	1

Decimal Number – 21_{10} = Binary Number – 10101_2

DIWAKAR EDUCATION HUB

Octal Number − 25_8 = Binary Number − 10101_2

Shortcut method - Binary to Octal

Steps

- **Step 1** − Divide the binary digits into groups of three (starting from the right).
- **Step 2** − Convert each group of three binary digits to one octal digit.

Example

Binary Number − 10101_2

Calculating Octal Equivalent −

Step	Binary Number	Octal Number
Step 1	10101_2	010 101
Step 2	10101_2	2_8 5_8
Step 3	10101_2	25_8

Binary Number − 10101_2 = Octal Number − 25_8

Shortcut method - Octal to Binary

Steps

- **Step 1** − Convert each octal digit to a 3 digit binary number (the octal digits may be treated as decimal for this conversion).
- **Step 2** − Combine all the resulting binary groups (of 3 digits each) into a single binary number.

Example

DIWAKAR EDUCATION HUB

Octal Number − 25_8

Calculating Binary Equivalent −

Step	Octal Number	Binary Number
Step 1	25_8	2_{10} 5_{10}
Step 2	25_8	010_2 101_2
Step 3	25_8	010101_2

Octal Number − 25_8 = Binary Number − 10101_2

Shortcut method - Binary to Hexadecimal

Steps

- **Step 1** − Divide the binary digits into groups of four (starting from the right).
- **Step 2** − Convert each group of four binary digits to one hexadecimal symbol.

Example

Binary Number − 10101_2

Calculating hexadecimal Equivalent −

Step	Binary Number	Hexadecimal Number
Step 1	10101_2	0001 0101
Step 2	10101_2	1_{10} 5_{10}
Step 3	10101_2	15_{16}

DIWAKAR EDUCATION HUB

Binary Number − 10101_2 = Hexadecimal Number − 15_{16}

Shortcut method - Hexadecimal to Binary

Steps

- **Step 1** − Convert each hexadecimal digit to a 4 digit binary number (the hexadecimal digits may be treated as decimal for this conversion).
- **Step 2** − Combine all the resulting binary groups (of 4 digits each) into a single binary number.

Example

Hexadecimal Number − 15_{16}

Calculating Binary Equivalent −

Step	Hexadecimal Number	Binary Number
Step 1	15_{16}	1_{10} 5_{10}
Step 2	15_{16}	0001_2 0101_2
Step 3	15_{16}	00010101_2

Hexadecimal Number − 15_{16} = Binary Number − 10101_2

Complement Arithmetic

Complements are used in the digital computers in order to simplify the subtraction operation and for the logical manipulations. For each radix-r system (radix r represents base of number system) there are two types of complements.

S.N.	Complement	Description

DIWAKAR EDUCATION HUB

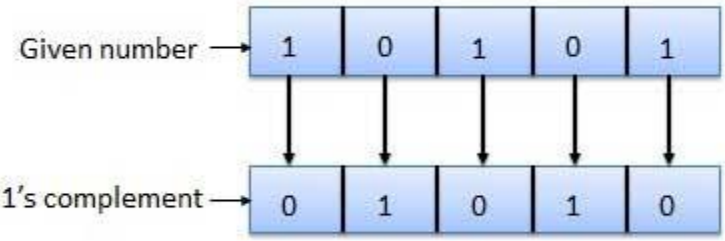
1	Radix Complement	The radix complement is referred to as the r's complement
2	Diminished Radix Complement	The diminished radix complement is referred to as the (r-1)'s complement

Binary system complements

As the binary system has base r = 2. So the two types of complements for the binary system are 2's complement and 1's complement.

1's complement

The 1's complement of a number is found by changing all 1's to 0's and all 0's to 1's. This is called as taking complement or 1's complement. Example of 1's Complement is as follows.



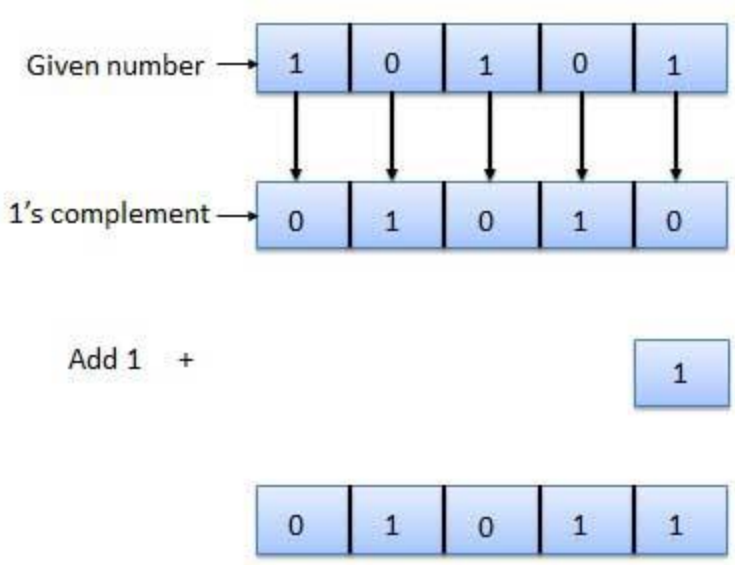
2's complement

The 2's complement of binary number is obtained by adding 1 to the Least Significant Bit (LSB) of 1's complement of the number.

2's complement = 1's complement + 1

Example of 2's Complement is as follows.

DIWAKAR EDUCATION HUB



Fixed Point and Floating Point Number Representations

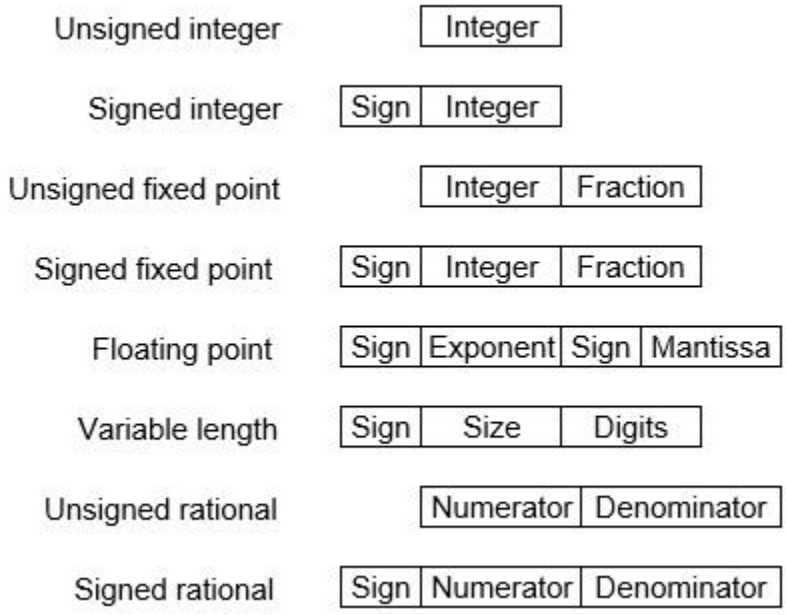
Digital Computers use Binary number system to represent all types of information inside the computers. Alphanumeric characters are represented using binary bits (i.e., 0 and 1). Digital representations are easier to design, storage is easy, accuracy and precision are greater.

There are various types of number representation techniques for digital number representation, for example: Binary number system, octal number system, decimal number system, and hexadecimal number system etc. But Binary number system is most relevant and popular for representing numbers in digital computer system.

Storing Real Number

These are structures as following below –

DIWAKAR EDUCATION HUB



There are two major approaches to store real numbers (i.e., numbers with fractional component) in modern computing. These are (i) Fixed Point Notation and (ii) Floating Point Notation. In fixed point notation, there are a fixed number of digits after the decimal point, whereas floating point number allows for a varying number of digits after the decimal point.

Fixed-Point Representation –

This representation has fixed number of bits for integer part and for fractional part. For example, if given fixed-point representation is IIII.FFFF, then you can store minimum value is 0000.0001 and maximum value is 9999.9999. There are three parts of a fixed-point number representation: the sign field, integer field, and fractional field.



We can represent these numbers using:

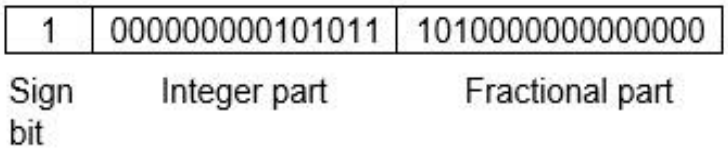
DIWAKAR EDUCA HUB

- Signed representation: range from $-(2^{(k-1)}-1)$ to $(2^{(k-1)}-1)$, for k bits.
- 1's complement representation: range from $-(2^{(k-1)}-1)$ to $(2^{(k-1)}-1)$, for k bits.
- 2's complementation representation: range from $-(2^{(k-1)})$ to $(2^{(k-1)}-1)$, for k bits.

2's complementation representation is preferred in computer system because of unambiguous property and easier for arithmetic operations.

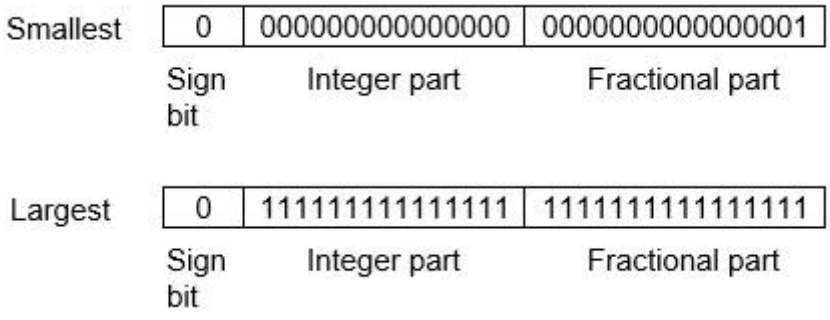
Example –Assume number is using 32-bit format which reserve 1 bit for the sign, 15 bits for the integer part and 16 bits for the fractional part.

Then, -43.625 is represented as following:



Where, 0 is used to represent + and 1 is used to represent. 000000000101011 is 15 bit binary value for decimal 43 and 1010000000000000 is 16 bit binary value for fractional 0.625.

The advantage of using a fixed-point representation is performance and disadvantage is relatively limited range of values that they can represent. So, it is usually inadequate for numerical analysis as it does not allow enough numbers and accuracy. A number whose representation exceeds 32 bits would have to be stored inexactly.



DIWAKAR EDUCATION HUB

These are above smallest positive number and largest positive number which can be store in 32-bit representation as given above format. Therefore, the smallest positive number is $2^{-16} \approx 0.000015$ approximate and the largest positive number is $(2^{15}-1)+(1-2^{-16})=2^{15}(1-2^{-16}) =32768$, and gap between these numbers is 2^{-16}.

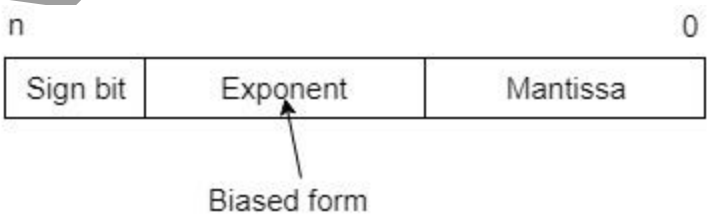
We can move the radix point either left or right with the help of only integer field is 1.

Floating Point Representation

This representation does not reserve a specific number of bits for the integer part or the fractional part. Instead it reserves a certain number of bits for the number (called the mantissa or significand) and a certain number of bits to say where within that number the decimal place sits (called the exponent).

The floating number representation of a number has two part: the first part represents a signed fixed point number called mantissa. The second part of designates the position of the decimal (or binary) point and is called the exponent. The fixed point mantissa may be fraction or an integer. Floating -point is always interpreted to represent a number in the following form: Mxr^e.

Only the mantissa m and the exponent e are physically represented in the register (including their sign). A floating-point binary number is represented in a similar manner except that is uses base 2 for the exponent. A floating-point number is said to be normalized if the most significant digit of the mantissa is 1.



So, actual number is $(-1)^s(1+m)x2^{(e-Bias)}$, where s is the sign bit, m is the mantissa, e is the exponent value, and Bias is the bias number.

DIWAKAR EDUCATION HUB

Note that signed integers and exponent are represented by either sign representation, or one's complement representation, or two's complement representation.

The floating point representation is more flexible. Any non-zero number can be represented in the normalized form of $\pm(1.b_1b_2b_3 \ldots)_2 x 2^n$ This is normalized form of a number x.

The following description explains terminology and primary details of IEEE 754 binary floating point representation. The discussion confines to single and double precision formats.

Usually, a real number in binary will be represented in the following format,

$$I_mI_{m-1}\ldots I_2I_1I_0.F_1F_2\ldots F_nF_{n-1}$$

Where I_m and F_n will be either 0 or 1 of integer and fraction parts respectively.

A finite number can also represented by four integers components, a sign (s), a base (b), a significand (m), and an exponent (e). Then the numerical value of the number is evaluated as

$(-1)^s$ x m x b^e ——— Where m < |b|

Depending on base and the number of bits used to encode various components, the IEEE 754 standard defines five basic formats. Among the five formats, the binary32 and the binary64 formats are single precision and double precision formats respectively in which the base is 2.

Table – 1 Precision Representation

Precision	**Base**	**Sign**	**Exponent**	**Significand**
Single precision	2	1	8	23+1

DIWAKAR EDUCATION HUB

Double precision	2	1	11	52+1

Single Precision Format:

As mentioned in Table 1 the single precision format has 23 bits for significand (1 represents implied bit, details below), 8 bits for exponent and 1 bit for sign.

For example, the rational number 9÷2 can be converted to single precision float format as following,

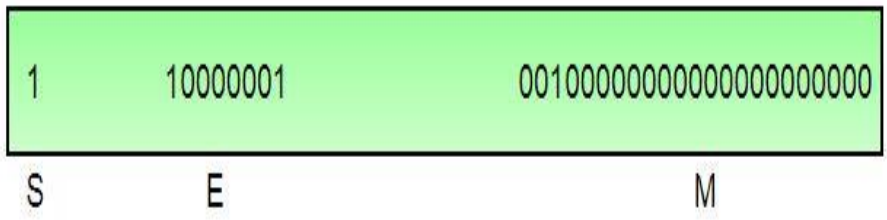
$$9_{(10)} \div 2_{(10)} = 4.5_{(10)} = 100.1_{(2)}$$

The result said to be **normalized**, if it is represented with leading 1 bit, i.e. $1.001_{(2)} \times 2^2$. (Similarly when the number $0.000000001101_{(2)} \times 2^3$ is normalized, it appears as $1.101_{(2)} \times 2^{-6}$). Omitting this implied 1 on left extreme gives us the **mantissa** of float number. A normalized number provides more accuracy than corresponding **de-normalized** number. The implied most significant bit can be used to represent even more accurate significand (23 + 1 = 24 bits) which is called **subnormal** representation. The floating point numbers are to be represented in normalized form.

The subnormal numbers fall into the category of de-normalized numbers. The subnormal representation slightly reduces the exponent range and can't be normalized since that would result in an exponent which doesn't fit in the field. Subnormal numbers are less accurate, i.e. they have less room for nonzero bits in the fraction field, than normalized numbers. Indeed, the accuracy drops as the size of the subnormal number decreases. However, the subnormal representation is useful in filing gaps of floating point scale near zero.

In other words, the above result can be written as $(-1)^0 \times 1.001_{(2)} \times 2^2$ which yields the integer components as s = 0, b = 2, significand (m) = 1.001, mantissa = 001 and e = 2. The corresponding single precision floating number can be represented in binary as shown below,

DIWAKAR EDUCATION HUB



Where the exponent field is supposed to be 2, yet encoded as 129 (127+2) called **biased exponent**. The exponent field is in plain binary format which also represents negative exponents with an encoding (like sign magnitude, 1's complement, 2's complement, etc.). The biased exponent is used for the representation of negative exponents. The biased exponent has advantages over other negative representations in performing bitwise comparing of two floating point numbers for equality.

A **bias** of $(2^{n-1} - 1)$, where **n** is # of bits used in exponent, is added to the exponent (e) to get biased exponent (E). So, the biased exponent (E) of single precision number can be obtained as

$$E = e + 127$$

The range of exponent in single precision format is -128 to +127. Other values are used for special symbols.

Note: When we unpack a floating point number the exponent obtained is the biased exponent. Subtracting 127 from the biased exponent we can extract unbiased exponent.

Double Precision Format:

As mentioned in Table – 1 the double precision format has 52 bits for significand (1 represents implied bit), 11 bits for exponent and 1 bit for sign. All other definitions are same for double precision format, except for the size of various components.

Precision:

DIWAKAR EDUCATION HUB

The smallest change that can be represented in floating point representation is called as precision. The fractional part of a single precision normalized number has exactly 23 bits of resolution, (24 bits with the implied bit). This corresponds to $\log_{(10)} (2^{23}) = 6.924 = 7$ (the characteristic of logarithm) decimal digits of accuracy. Similarly, in case of double precision numbers the precision is $\log_{(10)} (2^{52}) = 15.654 = 16$ decimal digits.

Accuracy:

Accuracy in floating point representation is governed by number of significand bits, whereas range is limited by exponent. Not all real numbers can exactly be represented in floating point format. For any numberwhich is not floating point number, there are two options for floating point approximation, say, the closest floating point number less than x as x_ and the closest floating point number greater than x as x+. A **rounding** operation is performed on number of significant bits in the mantissa field based on the selected mode. The **round down** mode causes x set to x_, the **round up** mode causes x set to x+, the **round towards zero** mode causes x is either x_ or x+ whichever is between zero and. The **round to nearest** mode sets x to x_ or x+ whichever is nearest to x. Usually **round to nearest** is most used mode. The closeness of floating point representation to the actual value is called as **accuracy**.

Special Bit Patterns:

The standard defines few special floating point bit patterns. Zero can't have most significant 1 bit, hence can't be normalized. The hidden bit representation requires a special technique for storing zero. We will have two different bit patterns +0 and -0 for the same numerical value zero. For single precision floating point representation, these patterns are given below,

0 00000000 00000000000000000000000 = +0

1 00000000 00000000000000000000000 = -0

DIWAKAR EDUCATION HUB

Similarly, the standard represents two different bit patters for +INF and -INF. The same are given below,

0 11111111 00000000000000000000000 = +INF

1 11111111 00000000000000000000000 = -INF

All of these special numbers, as well as other special numbers (below) are subnormal numbers, represented through the use of a special bit pattern in the exponent field. This slightly reduces the exponent range, but this is quite acceptable since the range is so large.

An attempt to compute expressions like 0 x INF, 0 ÷ INF, etc. make no mathematical sense. The standard calls the result of such expressions as Not a Number (NaN). Any subsequent expression with NaN yields NaN. The representation of NaN has non-zero significand and all 1s in the exponent field. These are shown below for single precision format (x is don't care bits),

x 11111111 1**m**00000000000000000000

Where **m** can be 0 or 1. This gives us two different representations of NaN.

0 11111111 11000000000000000000000 _____ Signaling NaN (SNaN)

0 11111111 10000000000000000000000 _____Quiet NaN (QNaN)

Usually QNaN and SNaN are used for error handling. QNaN do not raise any exceptions as they propagate through most operations. Whereas SNaN are which when consumed by most operations will raise an invalid exception.

Overflow and Underflow:

Overflow is said to occur when the true result of an arithmetic operation is finite but larger in magnitude than the largest floating point number which can be stored using the given precision. **Underflow** is said to occur when the true result of an arithmetic operation is smaller in magnitude (infinitesimal) than the

DIWAKAR EDUCATION HUB

smallest normalized floating point number which can be stored. Overflow can't be ignored in calculations whereas underflow can effectively be replaced by zero.

Endianness:

The IEEE 754 standard defines a binary floating point format. The architecture details are left to the hardware manufacturers. The storage order of individual bytes in binary floating point numbers varies from architecture to architecture.

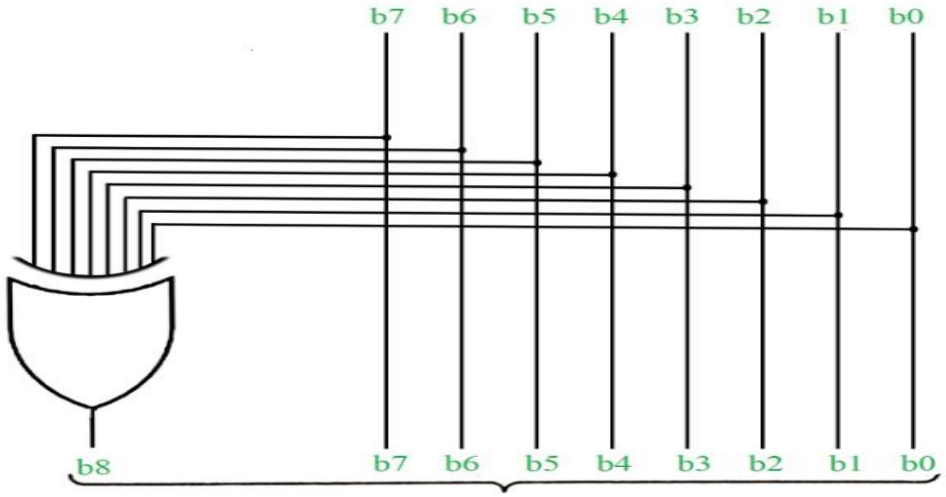
Error Detection Codes

The binary information is transferred from one location to another location through some communication medium. The external noise can change bits from 1 to 0 or 0 to 1.This changes in values are called errors. For efficient data transfer, there should be an error detection and correction codes. An error detection code is a binary code that detects digital errors during transmission. A famous error detection code is a Parity Bit method.

Parity Bit Method :

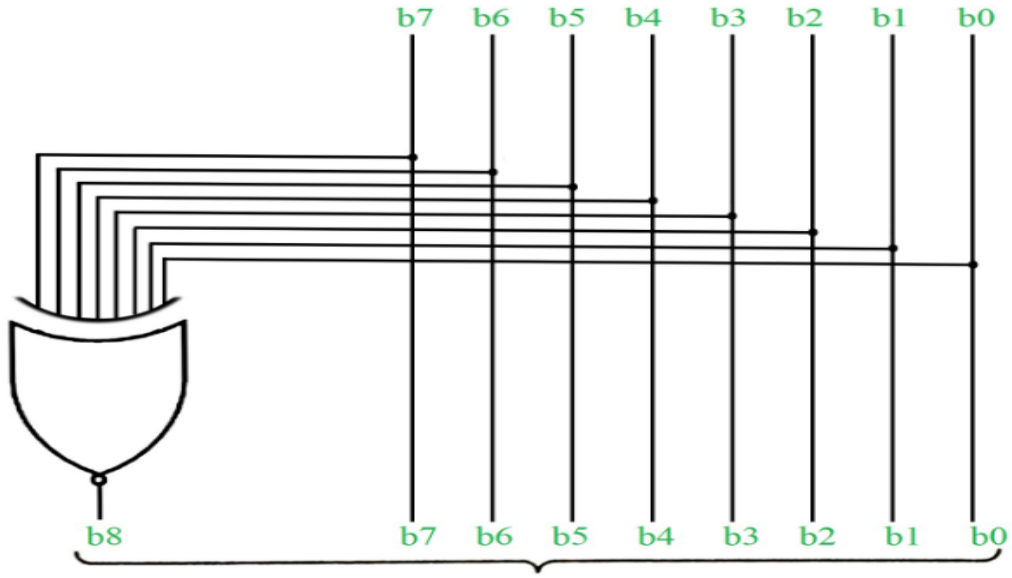
A parity bit is an extra bit included in binary message to make total number of 1's either odd or even. Parity word denotes number of 1's in a binary string. There are two parity system-even and odd. In even parity system 1 is appended to binary string it there is an odd number of 1's in string otherwise 0 is appended to make total even number of 1's.

DIWAKAR EDUCATION HUB



In odd parity system, 1 is appended to binary string if there is even a number of 1's to make an odd number of 1's. The receiver knows that whether sender is an odd parity generator or even parity generator. Suppose if sender is an odd parity generator then there must be an odd number of 1's in received binary string. If an error occurs to a single bit that is either bit is changed to 1 to 0 or O to 1, received binary bit will have an even number of 1's which will indicate an error.

DIWAKAR EDUCA



msg transmitted
Odd Parity Generator

The limitation of this method is that only error in a single bit would be identified.

MESSAGE (XYZ)	P(ODD)	P(EVEN)
000	1	0
001	0	1
010	0	1
011	1	0
100	0	1
101	1	0

DIWAKAR EDUCATION HUB

110	1	0
111	0	1

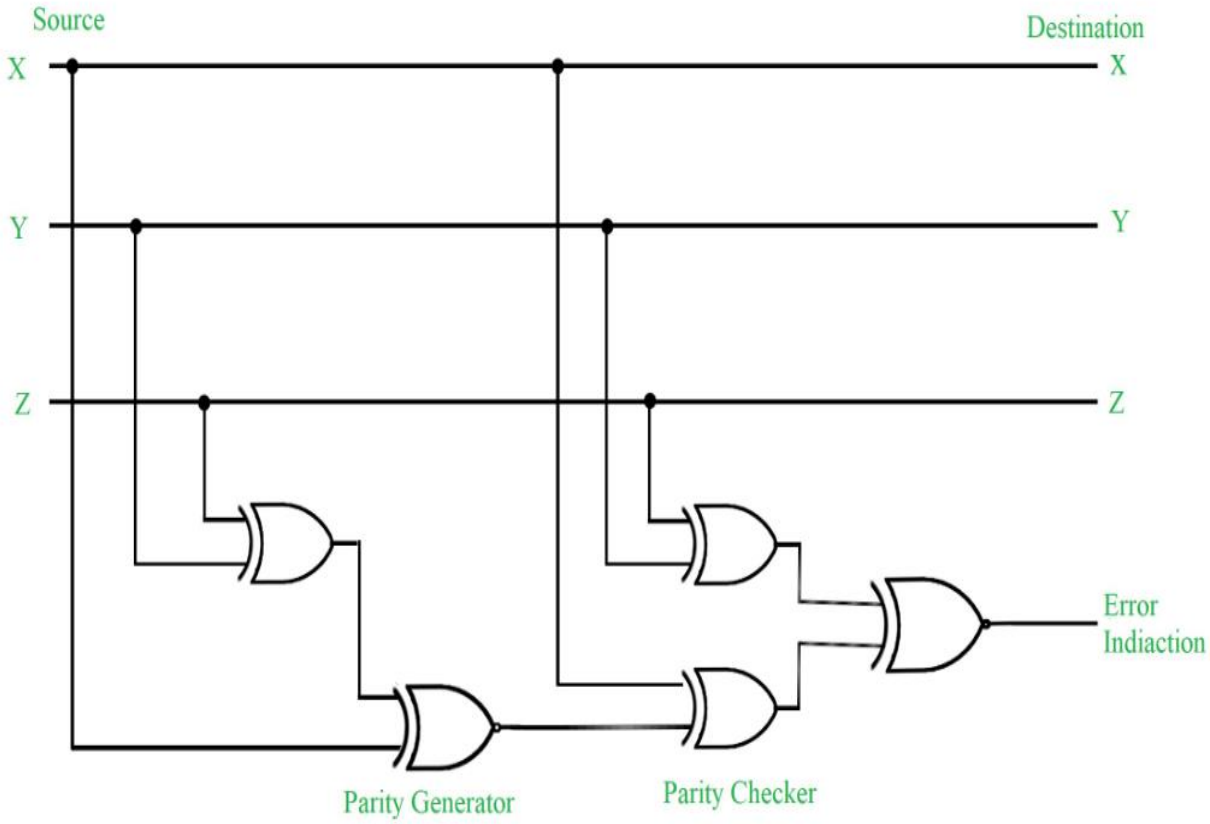


Figure – Error Detection with Odd Parity Bit

Points to Remember :

- In 1's complement of signed number +0 and -0 has two different representation.
- The range of signed magnitude representation of an 8-bit number in which 1-bit is used as a signed bit as follows -2^7 to $+2^7$.

- Floating point number is said to be normalized if most significant digit of mantissa is one. For example, 6-bit binary number 001101 is normalized because of two leading 0's.
- Booth algorithm that uses two n bit numbers for multiplication gives results in 2n bits.
- The booth algorithm uses 2's complement representation of numbers and work for both positive and negative numbers.
- If k-bits are used to represent exponent then bits number = (2^{k-1}) and range of exponent = $-(2^{k-1}-1)$ to $(2^{k}-1)$.

Features of Error detection codes

These are the following features of error detection codes:

- These codes are used when we use message backward error correction techniques for reliable data transmission. A feedback message is sent by the receiver to inform the sender whether the message is received without any error or not at the receiver side. If the message contains errors, the sender retransmits the message.
- In error detection codes, in fixed-size blocks of bits, the message is contained. In this, the redundant bits are added for correcting and detecting errors.
- These codes involve checking of the error. No matter how many error bits are there and the type of error.
- Parity check, Checksum, and CRC are the error detection technique.

Computer Arithmetic

Computer arithmetic is a field of computer science that investigates how computers should represent numbers and perform operations on them. It includes integer arithmetic, fixed-point arithmetic, and the arithmetic this book focuses on: floating-point (FP) arithmetic, which will be more thoroughly described in Chapter 1. For now, let us say that it is the common way computers

DIWAKAR EDUCATION HUB

approximate real numbers and that it is described in the IEEE-754 standard [IEE 08]. As in scientific notations, numbers are represented using an exponent and a significand, except that this significand has to fit on a certain amount of bits. As this number of bits (called precision) is limited, each operation may be inexact due to the rounding. This makes computer arithmetic sometimes inaccurate: the result of a long computation may be far from the mathematical result that would have been obtained if all the computations were correct. This also makes computer arithmetic unintuitive: for instance, the FP addition is not always associative.

Register Transfer Language

A digital computer system exhibits an interconnection of digital modules such as registers, decoders, arithmetic elements, and Control logic.

These digital modules are interconnected with some common data and control paths to form a complete digital system.

Moreover, digital modules are best defined by the registers and the operations that are performed on the data stored in them.

The operations performed on the data stored in registers are called **Micro-operations**.

The internal hardware organization of a digital system is best defined by specifying:

- The set of registers and the flow of data between them.
- The sequence of micro-operations performed on the data which are stored in the registers.
- The control paths that initiates the sequence of micro-operation

The **Register Transfer Language** is the symbolic representation of notations used to specify the sequence of micro-operations.

DIWAKAR EDUCATION HUB

In a computer system, data transfer takes place between processor registers and memory and between processor registers and input-output systems. These data transfer can be represented by standard notations given below:

- Notations R0, R1, R2..., and so on represent processor registers.
- The addresses of memory locations are represented by names such as LOC, PLACE, MEM, etc.
- Input-output registers are represented by names such as DATA IN, DATA OUT and so on.
- The content of register or memory location is denoted by placing square brackets around the name of the register or memory location.

Register Transfer

The term Register Transfer refers to the availability of hardware logic circuits that can perform a given micro-operation and transfer the result of the operation to the same or another register.

Most of the standard notations used for specifying operations on various registers are stated below.

- The memory address register is designated by **MAR**.
- Program Counter **PC** holds the next instruction's address.
- Instruction Register **IR** holds the instruction being executed.
- **R1** (Processor Register).
- We can also indicate individual bits by placing them in parenthesis. For instance, PC (8-15), R2 (5), etc.
- Data Transfer from one register to another register is represented in symbolic form by means of replacement operator. For instance, the

DIWAKAR EDUCATION HUB

following statement denotes a transfer of the data of register R1 into register R2.

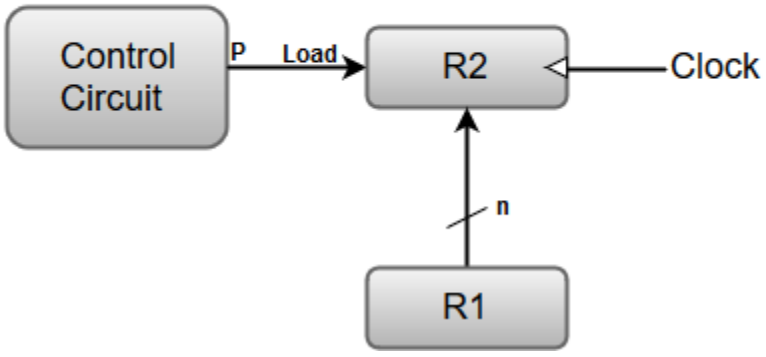
1. R2 ← R1

- Typically, most of the users want the transfer to occur only in a predetermined control condition. This can be shown by following if-then statement:
  If (P=1) then (R2 ← R1); Here P is a control signal generated in the control section.
- It is more convenient to specify a control function (P) by separating the control variables from the register transfer operation. For instance, the following statement defines the data transfer operation under a specific control function (P).

1. P: R2 ← R1

The following image shows the block diagram that depicts the transfer of data from R1 to R2.

Transfer from R1 to R2 when P = 1:



Here, the letter 'n' indicates the number of bits for the register. The 'n' outputs of the register R1 are connected to the 'n' inputs of register R2.

DIW EDUCATION HUB

A load input is activated by the control variable 'P' which is transferred to the register R2.

Bus and Memory Transfers

A digital system composed of many registers, and paths must be provided to transfer information from one register to another. The number of wires connecting all of the registers will be excessive if separate lines are used between each register and all other registers in the system.

A bus structure, on the other hand, is more efficient for transferring information between registers in a multi-register configuration system.

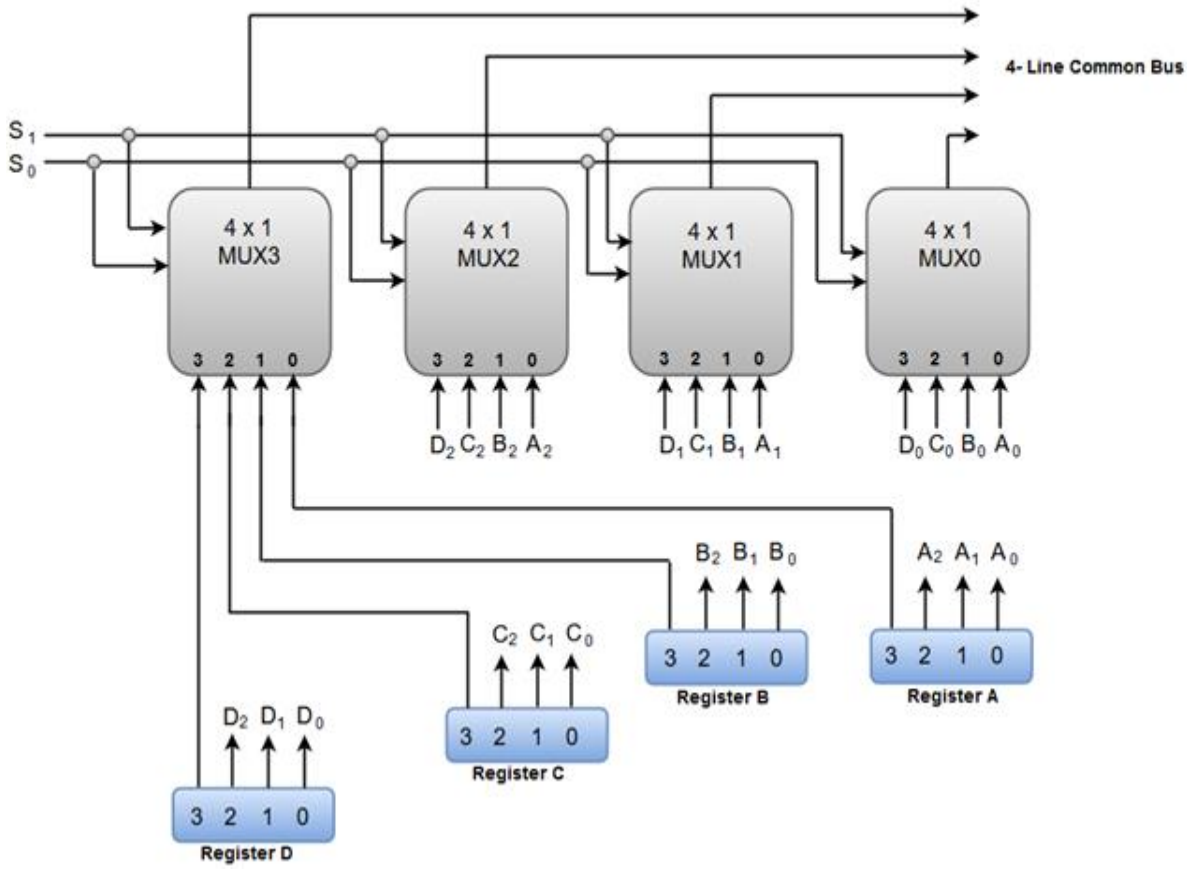
A bus consists of a set of common lines, one for each bit of register, through which binary information is transferred one at a time. Control signals determine which register is selected by the bus during a particular register transfer.

The following block diagram shows a Bus system for four registers. It is constructed with the help of four 4 * 1 Multiplexers each having four data inputs (0 through 3) and two selection inputs (S1 and S2).

We have used labels to make it more convenient for you to understand the input-output configuration of a Bus system for four registers. For instance, output 1 of register A is connected to input 0 of MUX1.

DIWAKAR EDUCATION HUB

Bus System for 4 Registers:



The two selection lines S1 and S2 are connected to the selection inputs of all four multiplexers. The selection lines choose the four bits of one register and transfer them into the four-line common bus.

When both of the select lines are at low logic, i.e. S1S0 = 00, the 0 data inputs of all four multiplexers are selected and applied to the outputs that forms the bus. This, in turn, causes the bus lines to receive the content of register A since the outputs of this register are connected to the 0 data inputs of the multiplexers.

Similarly, when S1S0 = 01, register B is selected, and the bus lines will receive the content provided by register B.

DIWAKAR

The following function table shows the register that is selected by the bus for each of the four possible binary values of the Selection lines.

S1	S0	Register Selected
0	0	A
0	1	B
1	0	C
1	1	D

Note: The number of multiplexers needed to construct the bus is equal to the number of bits in each register. The size of each multiplexer must be 'k * 1' since it multiplexes 'k' data lines. For instance, a common bus for eight registers of 16 bits each requires 16 multiplexers, one for each line in the bus. Each multiplexer must have eight data input lines and three selection lines to multiplex one significant bit in the eight registers.

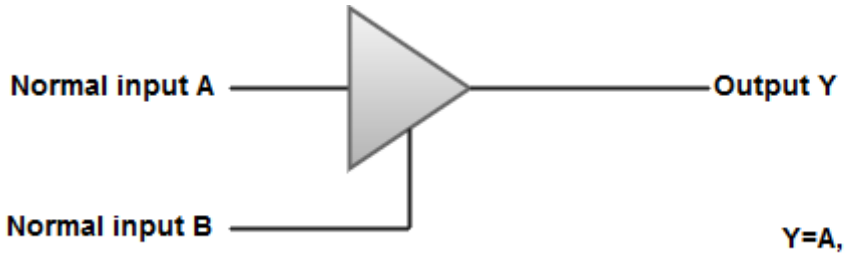
A bus system can also be constructed using **three-state gates** instead of multiplexers.

The **three state gates** can be considered as a digital circuit that has three gates, two of which are signals equivalent to logic 1 and 0 as in a conventional gate. However, the third gate exhibits a high-impedance state.

The most commonly used three state gates in case of the bus system is a **buffer gate**.

The graphical symbol of a three-state buffer gate can be represented as:

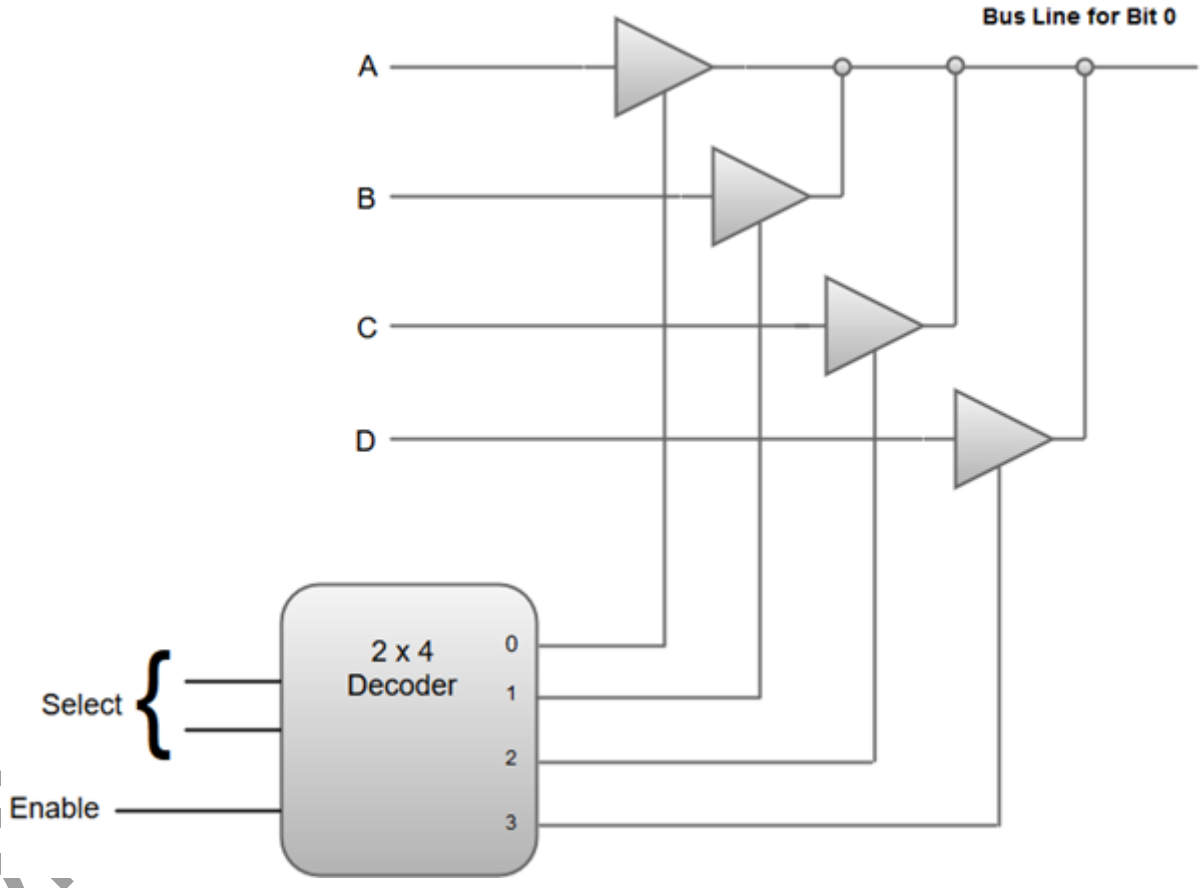
DIWAKAR EDUCATION HUB



Y=A, if C=1
High-impedance if C=0

The following diagram demonstrates the construction of a bus system with three-state buffers.

Bus line with three state buffer:



- The outputs generated by the four buffers are connected to form a single bus line.

- Only one buffer can be in active state at a given point of time.
- The control inputs to the buffers determine which of the four normal inputs will communicate with the bus line.
- A 2 * 4 decoder ensures that no more than one control input is active at any given point of time.

Memory Transfer

Most of the standard notations used for specifying operations on memory transfer are stated below.

- The transfer of information from a memory unit to the user end is called a **Read** operation.
- The transfer of new information to be stored in the memory is called a **Write** operation.
- A memory word is designated by the letter **M**.
- We must specify the address of memory word while writing the memory transfer operations.
- The **address register** is designated by **AR** and the **data register** by **DR**.
- Thus, a read operation can be stated as:

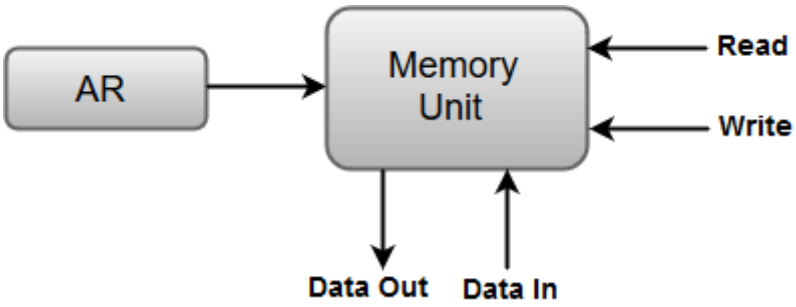
1. Read: DR ← M [AR]

- The **Read** statement causes a transfer of information into the data register (DR) from the memory word (M) selected by the address register (AR).
- And the corresponding write operation can be stated as:

1. Write: M [AR] ← R1

- The Write statement causes a transfer of information from register R1 into the memory word (M) selected by address register (AR).

DIWAKAR EDUCATION HUB



Micro-Operations

The operations executed on data stored in registers are called micro-operations. A micro-operation is an elementary operation performed on the information stored in one or more registers.

Example: Shift, count, clear and load.

Types of Micro-Operations

The micro-operations in digital computers are of 4 types:

1. Register transfer micro-operations transfer binary information from one register to another.
2. Arithmetic micro-operations perform arithmetic operations on numeric data stored in registers.
3. Logic micro-operations perform bit manipulation operation on non-numeric data stored in registers.
4. Shift micro-operations perform shift micro-operations performed on data.

Arithmetic Micro-operations

In general, the Arithmetic Micro-operations deals with the operations performed on numeric data stored in the registers.

The basic Arithmetic Micro-operations are classified in the following categories:

1. Addition

DIWAKAR EDUCATION HUB

2. Subtraction
3. Increment
4. Decrement
5. Shift

Some additional Arithmetic Micro-operations are classified as:

1. Add with carry
2. Subtract with borrow
3. Transfer/Load, etc.

The following table shows the symbolic representation of various Arithmetic Micro-operations.

Symbolic Representation	Description
R3 ← R1 + R2	The contents of R1 plus R2 are transferred to R3.
R3 ← R1 - R2	The contents of R1 minus R2 are transferred to R3.
R2 ← R2'	Complement the contents of R2 (1's complement)
R2 ← R2' + 1	2's complement the contents of R2 (negate)
R3 ← R1 + R2' + 1	R1 plus the 2's complement of R2 (subtraction)
R1 ← R1 + 1	Increment the contents of R1 by one

ATION HUB

R1 ← R1 - 1	Decrement the contents of R1 by one

Note: The increment and decrement micro-operations are symbolized by '+ 1' and '? 1' respectively. Arithmetic operations like multiply and divide are not included in the basic set of micro-operations.

Logic Micro-Operations

These are binary micro-operations performed on the bits stored in the registers. These operations consider each bit separately and treat them as binary variables.

Let us consider the X-OR micro-operation with the contents of two registers R1 and R2.

P: R1 ← R1 X-OR R2

In the above statement we have also included a Control Function.

Assume that each register has 3 bits. Let the content of R1 be **010** and R2 be **100**. The X-OR micro-operation will be:

010 → R1

100 → R2

110 → R1 after P=1

Shift Micro-Operations

These are used for serial transfer of data. That means we can shift the contents of the register to the left or right. In the **shift left** operation the serial input transfers a bit to the right most position and in **shift right** operation the serial input transfers a bit to the left most position.

There are three types of shifts as follows:

DIWAKAR EDUCATION HUB

a) Logical Shift

It transfers 0 through the serial input. The symbol **"shl"** is used for logical shift left and **"shr"** is used for logical shift right.

$$R1 \leftarrow she\ R1$$

$$R1 \leftarrow she\ R1$$

The register symbol must be same on both sides of arrows.

b) Circular Shift

This circulates or rotates the bits of register around the two ends without any loss of data or contents. In this, the serial output of the shift register is connected to its serial input. **"cil"** and **"cir"** is used for circular shift left and right respectively.

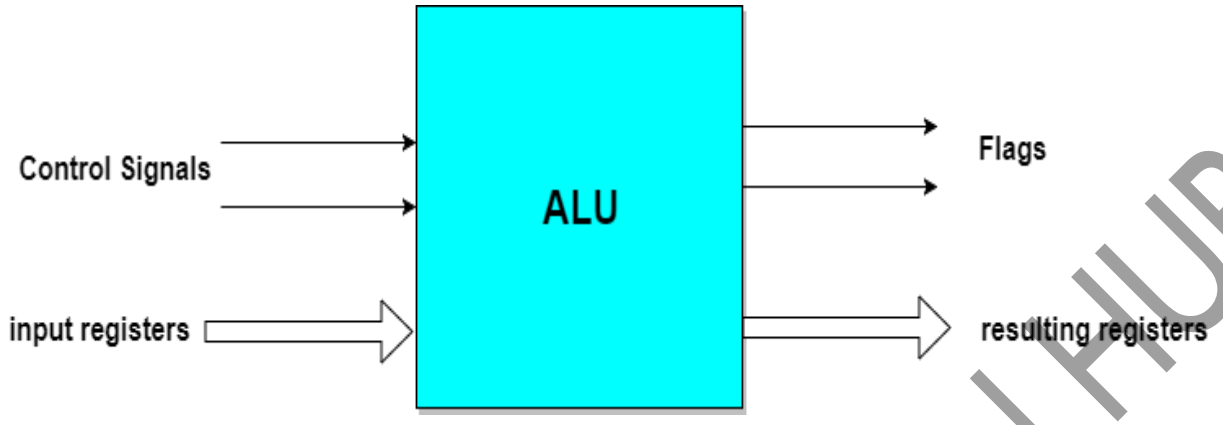
c) Arithmetic Shift

This shifts a signed binary number to left or right. An **arithmetic shift left** multiplies a signed binary number by 2 and **shift left** divides the number by 2. Arithmetic shift micro-operation leaves the sign bit unchanged because the signed number remains same when it is multiplied or divided by 2.

Arithmetic Logical Unit

Instead of having individual registers performing the micro-operations, computer system provides a number of registers connected to a common unit called as Arithmetic Logical Unit (ALU). ALU is the main and one of the most important unit inisde CPU of computer. All the logical and mathematical operations of computer are performed here. The contents of specific register is placed in the in the input of ALU. ALU performs the given operation and then transfer it to the destination register.

DIWAKAR EDUCATION HUB



Registers

Register is a very fast computer memory, used to store data/instruction in-execution.

A **Register** is a group of flip-flops with each flip-flop capable of storing **one bit** of information. An n-bit register has a group of n flip-flops and is capable of storing binary information of n-bits.

A register consists of a group of flip-flops and gates. The flip-flops hold the binary information and gates control when and how new information is transferred into a register. Various types of registers are available commercially. The simplest register is one that consists of only flip-flops with no external gates.

These days registers are also implemented as a register file.

Loading the Registers

The transfer of new information into a register is referred to as loading the register. If all the bits of register are loaded simultaneously with a common clock pulse than the loading is said to be done in parallel.

Instruction Codes

While a **Program**, as we all know, is, A set of instructions that specify the operations, operands, and the sequence by which processing has to occur.

DIWAKAR EDUCATION HUB

An **instruction code** is a group of bits that tells the computer to perform a specific operation part.

Instruction Code: Operation Code

The operation code of an instruction is a group of bits that define operations such as add, subtract, multiply, shift and compliment. The number of bits required for the operation code depends upon the total number of operations available on the computer. The operation code must consist of at least **n bits** for a given **2^n** operations. The operation part of an instruction code specifies the operation to be performed.

Instruction Code: Register Part

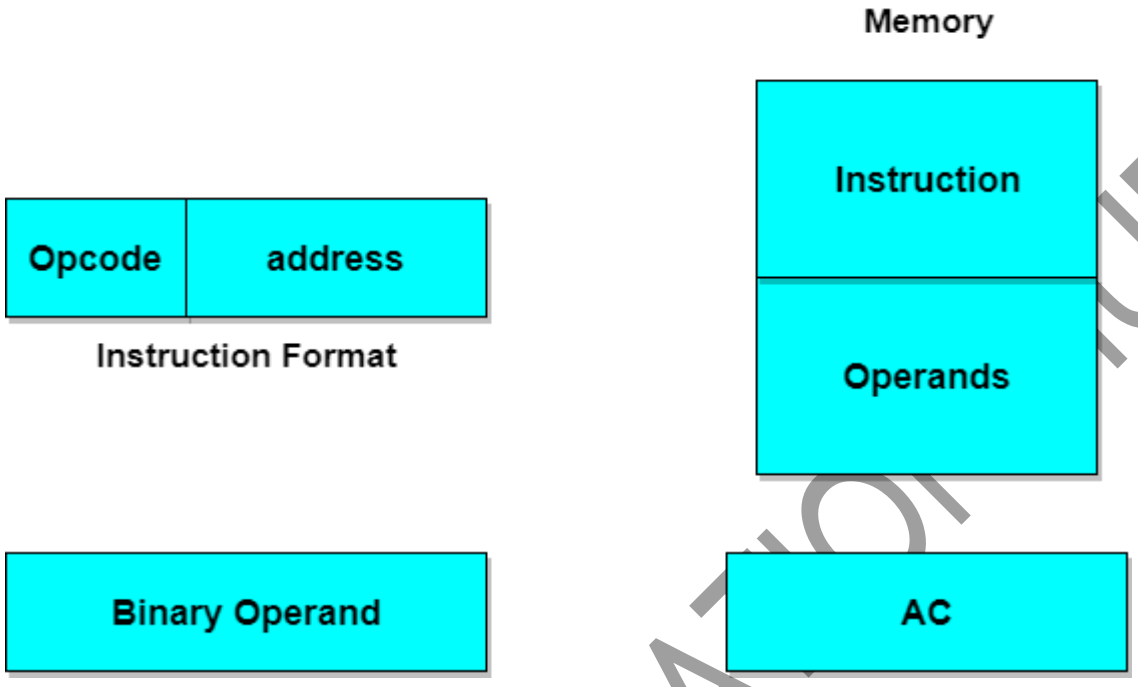
The operation must be performed on the data stored in registers. An instruction code therefore specifies not only operations to be performed but also the registers where the operands(data) will be found as well as the registers where the result has to be stored.

Stored Program Organisation

The simplest way to organize a computer is to have **Processor Register** and instruction code with two parts. The first part specifies the operation to be performed and second specifies an address. The memory address tells where the operand in memory will be found.

Instructions are stored in one section of memory and data in another.

DIWAKAR EDUCATION HUB



Computers with a single processor register is known as **Accumulator (AC)**. The operation is performed with the memory operand and the content of AC.

Common Bus System

The basic computer has 8 registers, a memory unit and a control unit. Paths must be provided to transfer data from one register to another. An efficient method for transferring data in a system is to use a **Common Bus System**. The output of registers and memory are connected to the common bus.

Load(LD)

The lines from the common bus are connected to the inputs of each register and data inputs of memory. The particular register whose **LD** input is enabled receives the data from the bus during the next clock pulse transition.

Before studying about instruction formats lets first study about the operand address parts.

When the 2nd part of an instruction code specifies the operand, the instruction is said to have **immediate operand**. And when the 2nd part of the instruction code

DIWAKAR EDUCATION HUB

specifies the address of an operand, the instruction is said to have a **direct address**. And in **indirect address**, the 2nd part of instruction code, specifies the address of a memory word in which the address of the operand is found.

Computer Instructions

The basic computer has three instruction code formats. The **Operation code** (opcode) part of the instruction contains 3 bits and remaining 13 bits depends upon the operation code encountered.

There are three types of formats:

1. Memory Reference Instruction

It uses 12 bits to specify the address and 1 bit to specify the addressing mode (**I**). **I** is equal to 0 for direct address and 1 for indirect address.

2. Register Reference Instruction

These instructions are recognized by the opcode 111 with a 0 in the left most bit of instruction. The other 12 bits specify the operation to be executed.

3. Input-Output Instruction

These instructions are recognized by the operation code 111 with a 1 in the left most bit of instruction. The remaining 12 bits are used to specify the input-output operation.

Format of Instruction

The format of an instruction is depicted in a rectangular box symbolizing the bits of an instruction. Basic fields of an instruction format are given below:

1. An operation code field that specifies the operation to be performed.
2. An address field that designates the memory address or register.

DIWAKAR EDUCATION HUB

3. A mode field that specifies the way the operand of effective address is determined.

Computers may have instructions of different lengths containing varying number of addresses. The number of address field in the instruction format depends upon the internal organization of its registers.

Adressing Modes and Instruction Cycle

The operation field of an instruction specifies the operation to be performed. This operation will be executed on some data which is stored in computer registers or the main memory. The way any operand is selected during the program execution is dependent on the addressing mode of the instruction. The purpose of using addressing modes is as follows:

1. To give the programming versatility to the user.
2. To reduce the number of bits in addressing field of instruction.

Types of Addressing Modes

Below we have discussed different types of addressing modes one by one:

Immediate Mode

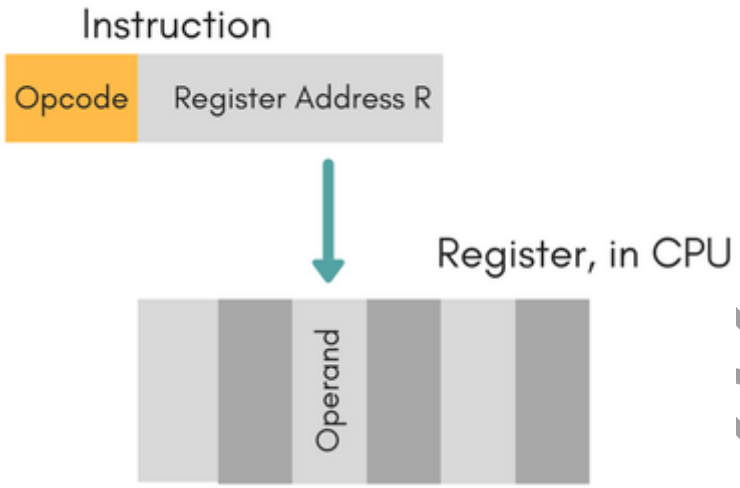
In this mode, the operand is specified in the instruction itself. An immediate mode instruction has an operand field rather than the address field.

For example: ADD 7, which says Add 7 to contents of accumulator. 7 is the operand here.

Register Mode

In this mode the operand is stored in the register and this register is present in CPU. The instruction has the address of the Register where the operand is stored.

DIWAKAR EDUCATION HUB



Advantages

- Shorter instructions and faster instruction fetch.
- Faster memory access to the operand(s)

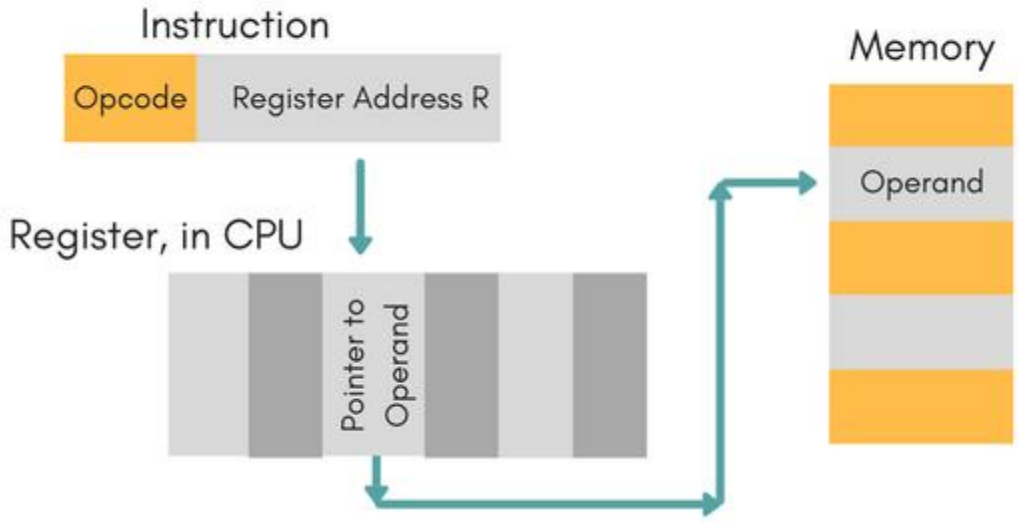
Disadvantages

- Very limited address space
- Using multiple registers helps performance but it complicates the instructions.

Register Indirect Mode

In this mode, the instruction specifies the register whose contents give us the address of operand which is in memory. Thus, the register contains the address of operand rather than the operand itself.

DIWAKAR EDUCATION HUB



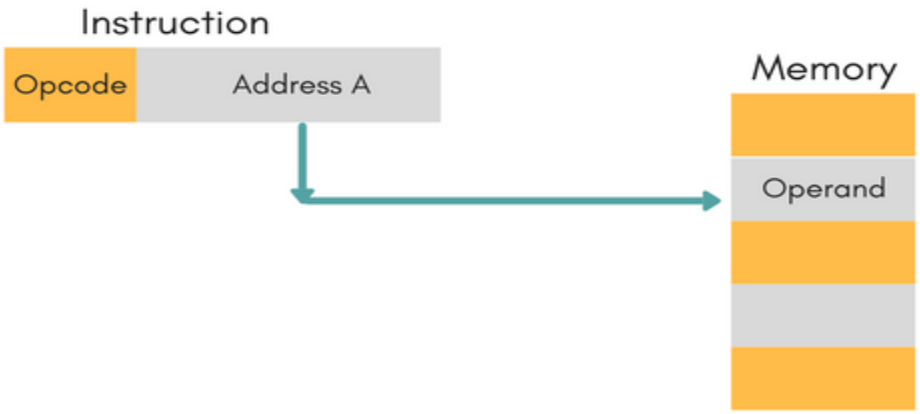
Auto Increment/Decrement Mode

In this the register is incremented or decremented after or before its value is used.

Direct Addressing Mode

In this mode, effective address of operand is present in instruction itself.

- Single memory reference to access data.
- No additional calculations to find the effective address of the operand.

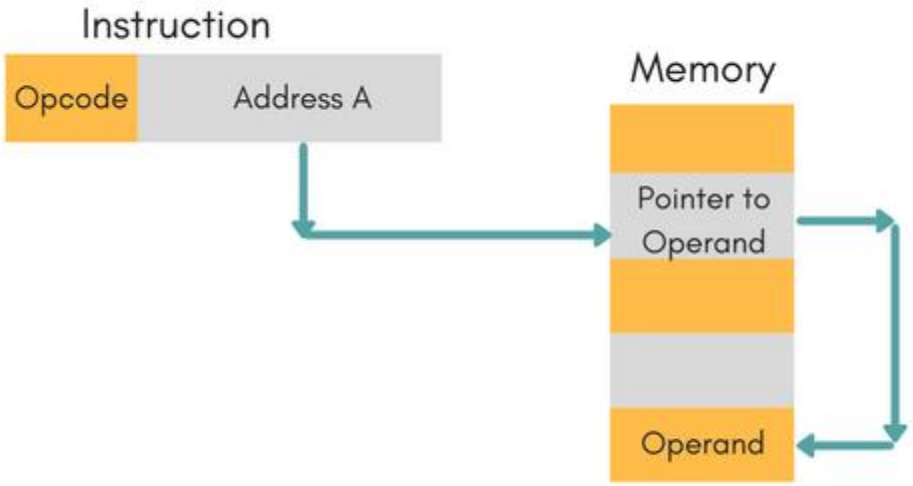


For Example: ADD R1, 4000 - In this the 4000 is effective address of operand.

NOTE: Effective Address is the location where operand is present.

Indirect Addressing Mode

In this, the address field of instruction gives the address where the effective address is stored in memory. This slows down the execution, as this includes multiple memory lookups to find the operand.

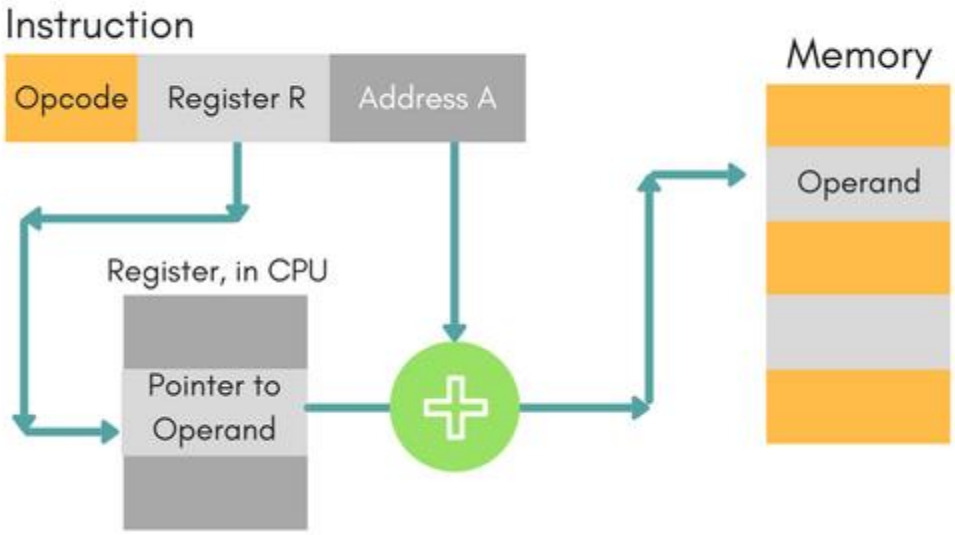


Displacement Addressing Mode

In this the contents of the indexed register is added to the Address part of the instruction, to obtain the effective address of operand.

EA = A + (R), In this the address field holds two values, A(which is the base value) and R(that holds the displacement), or vice versa.

DIWAKAR



Relative Addressing Mode

It is a version of Displacement addressing mode.

In this the contents of PC(Program Counter) is added to address part of instruction to obtain the effective address.

EA = A + (PC), where EA is effective address and PC is program counter.

The operand is A cells away from the current cell(the one pointed to by PC)

Base Register Addressing Mode

It is again a version of Displacement addressing mode. This can be defined as EA = A + (R), where A is displacement and R holds pointer to base address.

Stack Addressing Mode

In this mode, operand is at the top of the stack. For example: ADD, this instruction will POP top two items from the stack, add them, and will then PUSH the result to the top of the stack.

Instruction Cycle

DIWAKAR EDUCATION HUB

An instruction cycle, also known as **fetch-decode-execute cycle** is the basic operational process of a computer. This process is repeated continuously by CPU from boot up to shut down of computer.

Following are the steps that occur during an instruction cycle:

1. Fetch the Instruction

The instruction is fetched from memory address that is stored in PC(Program Counter) and stored in the instruction register IR. At the end of the fetch operation, PC is incremented by 1 and it then points to the next instruction to be executed.

2. Decode the Instruction

The instruction in the IR is executed by the decoder.

3. Read the Effective Address

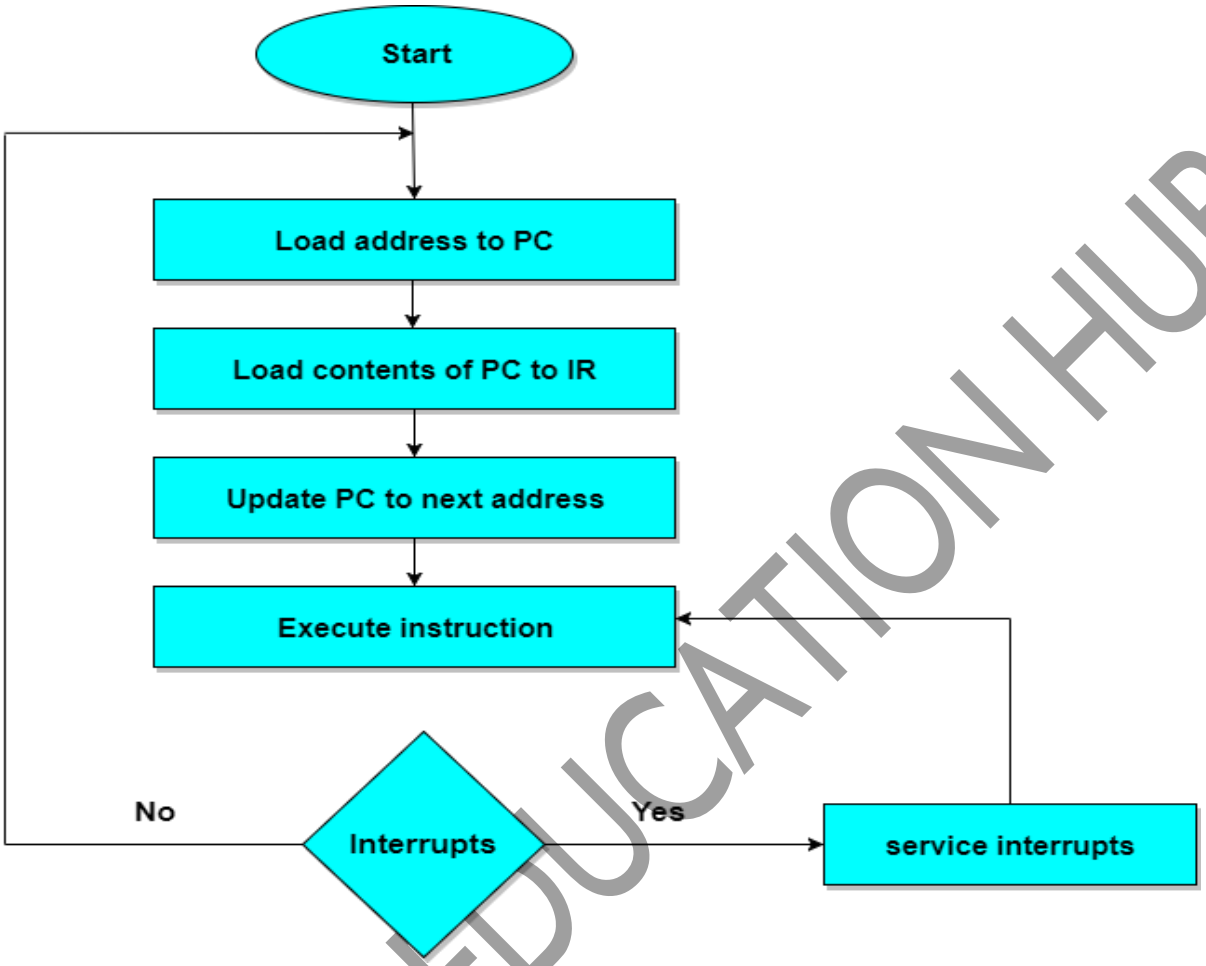
If the instruction has an indirect address, the effective address is read from the memory. Otherwise operands are directly read in case of immediate operand instruction.

4. Execute the Instruction

The Control Unit passes the information in the form of control signals to the functional unit of CPU. The result generated is stored in main memory or sent to an output device.

The cycle is then repeated by fetching the next instruction. Thus in this way the instruction cycle is repeated continuously.

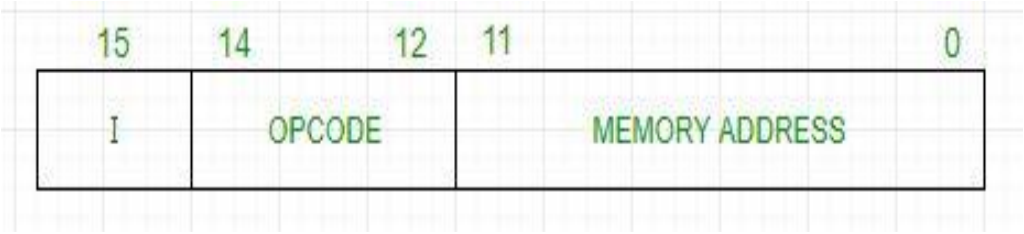
DIWAKAR EDUCATION HUB



Memory-Reference Instructions

The basic computer has 16-bit instruction register (IR) which can denote either memory reference or register reference or input-output instruction.

1. **Memory Reference –** These instructions refer to memory address as an operand. The other operand is always accumulator. Specifies 12-bit address, 3-bit opcode (other than 111) and 1-bit addressing mode for direct and indirect addressing.



DIWAKAR EDUCATION HUB

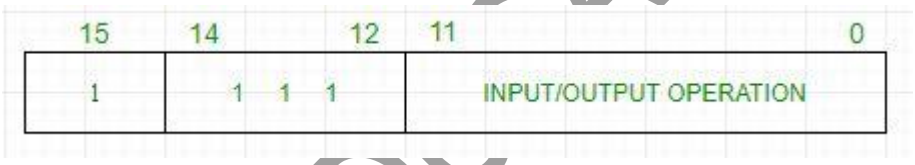
Example –
IR register contains = 0001XXXXXXXXXXX, i.e. ADD after fetching and decoding of instruction we find out that it is a memory reference instruction for ADD operation.

Hence, DR ← M[AR]

AC ← AC + DR, SC ← 0

Input-Output instructions

Input/Output – These instructions are for communication between computer and outside environment. The IR(14 – 12) is 111 (differentiates it from memory reference) and IR(15) is 1 (differentiates it from register reference instructions). The rest 12 bits specify I/O operation.



Example –
IR register contains = 1111100000000000, i.e. INP after fetch and decode cycle we find out that it is an input/output instruction for inputing character. Hence, INPUT character from peripheral device.

The set of instructions incorporated in16 bit IR register are:

1. Arithmetic, logical and shift instructions (and, add, complement, circulate left, right, etc)
2. To move information to and from memory (store the accumulator, load the accumulator)
3. Program control instructions with status conditions (branch, skip)
4. Input output instructions (input character, output character)

DIWAKAR EDUCATION HUB

Machine Language

Machine language, or machine code, is a low-level language comprised of binary digits (ones and zeros). High-level languages, such as Swift and C++ must be compiled into machine language before the code is run on a computer.

Since computers are digital devices, they only recognize binary data. Every program, video, image, and character of text is represented in binary. This binary data, or machine code, is processed as input by the CPU. The resulting output is sent to the operating system or an application, which displays the data visually. For example, the ASCII value for the letter "A" is 01000001 in machine code, but this data is displayed as "A" on the screen. An image may have thousands or even millions of binary values that determine the color of each pixel.

While machine code is comprised of 1s and 0s, different processor architectures use different machine code. For example, a PowerPC processor, which has a RISC architecture, requires different code than an Intel x86 processor, which has a CISC architecture. A compiler must compile high-level source cod for the correct processor architecture in order for a program to run correctly.

The exact machine language for a program or action can differ by operating system. The specific operating system dictates how a compiler writes a program or action into machine language.

Computer programs are written in one or more programming languages, like C++, Java, or Visual Basic. A computer cannot directly understand the programming languages used to create computer programs, so the program code must be compiled. Once a program's code is compiled, the computer can understand it because the program's code is turned into machine language.

Machine language example

Below is an example of machine language (binary) for the text "Hello World."

DIWAKAR EDUCATION HUB

```
01001000 01100101 01101100 01101100 01101111 00100000 01010111
01101111 01110010 01101100 01100100
```

Below is another example of machine language (non-binary), which prints the letter "A" 1000 times to the computer screen.

```
169 1 160 0 153 0 128 153 0 129 153 130 153 0 131 200 208 241 96
```

Assembly language

Sometimes referred to as **assembly** or **ASM**, an **assembly language** is a low-level programming language.

Programs written in assembly languages are compiled by an assembler. Every assembler has its own assembly language, which is designed for one specific computer architecture.

Why is ASM useful?

Machine language is a series of numbers, which is not easy for humans to read. Using ASM, programmers can write human-readable programs that correspond almost exactly to machine language.

The disadvantage is that everything the computer does must be described explicitly, in precise detail. The advantage is that the programmer has maximum control over what the computer is doing.

Why is ASM a "low-level" language?

Assembly is called a low-level programming language because there's (nearly) a one-to-one relationship between what it tells the computer to do, and what the computer does. In general, one line of an assembly program contains a maximum of one instruction for the computer.

How is ASM different from a "high-level" language?

DIWAKAR EDUCATION HUB

High-level languages provide abstractions of low-level operations which allow the programmer to focus more on describing what they want to do, and less on how it should be done. Programming this way is more convenient and makes programs easier to read at the sacrifice of low-level control.

Programs written in high-level languages never match the raw speed and efficiency of programs written in assembly. Examples of high-level languages include Python, Java, JavaScript, Clojure, and Lisp.

What is a "mid-level" language?

Mid-level languages or **lower-level languages** provide some high-level abstractions to make the programmer's life easier, while still providing access to low-level operations. They are often used to write operating systems, so they are sometimes called **system programming languages**.

Programs written in mid-level languages can perform as well, or nearly as well, as programs written in assembly language. Examples of mid-level programming languages include C, C++, Ada, Nim, and Rust.

Is ASM portable?

No. Because assembly languages are tied to one specific computer architecture, they are not portable. A program written in one assembly language would need to be completely rewritten for it to run on another type of machine.

Portability is one of the main advantages of higher-level languages. The C programming language is often called "portable assembly" because C compilers exist for nearly every modern system architecture. A program written in C may require some changes before it will compile on another computer, but the core language is portable.

Generally speaking, the higher-level a language is, the fewer changes need to be made for it to run on another architecture. The lowest-level languages — machine language and assembly language — are not portable.

DIWAKAR EDUCATION HUB

Machine Language vs Assembly Language

Machine language and assembly language are both low-level languages, but machine code is below assembly in the hierarchy of computer languages. Assembly language includes human-readable commands, such as mov, add, and sub, while machine language does not contain any words or even letters. Some developers manually write assembly language to optimize a program, but they do not write machine code. Only developers who write software compilers need to worry about machine language.

NOTE: While machine code is technically comprised of binary data, it may also be represented in hexadecimal values. For example, the letter "Z," which is 01011010 in binary, may be displayed as 5A in hexadecimal code.

Assembler

Program used to convert or translate programs written in assembly code to machine code. Some users may also refer to assembly language or **assembler** language as assembler.

An assembler is a program that converts assembly language into machine code. It takes the basic commands and operations from assembly code and converts them into binary code that can be recognized by a specific type of processor.

Assemblers are similar to compilers in that they produce executable code. However, assemblers are more simplistic since they only convert low-level code (assembly language) to machine code. Since each assembly language is designed for a specific processor, assembling a program is performed using a simple one-to-one mapping from assembly code to machine code. Compilers, on the other hand, must convert generic high-level source code into machine code for a specific processor.

Most programs are written in high-level programming languages and are compiled directly to machine code using a compiler. However, in some cases, assembly code may be used to customize functions and ensure they perform in a

DIWAKAR EDUCATION HUB

specific way. Therefore, IDEs often include assemblers so they can build programs from both high and low-level languages.

How it works:

- Most computers come with a specified set of very basic instructions that correspond to the basic machine operations that the computer can perform. For example, a "Load" instruction causes the processor to move a string of bits from a location in the processor's memory to a special holding place called a register. Assuming the processor has at least eight registers, each numbered, the following instruction would move the value (string of bits of a certain length) at memory location 3000 into the holding place called register 8:

```
L       8,3000
```

- The programmer can write a program using a sequence of these assembler instructions.
- This sequence of assembler instructions, known as the source code or source program, is then specified to the assembler program when that program is started.
- The assembler program takes each program statement in the source program and generates a corresponding bit stream or pattern (a series of 0's and 1's of a given length).
- The output of the assembler program is called the object code or object program relative to the input source program. The sequence of 0's and 1's that constitute the object program is sometimes called machine code.
- The object program can then be run (or executed) whenever desired.

In the earliest computers, programmers actually wrote programs in machine code, but assembler languages or instruction sets were soon developed to speed up programming. Today, assembler programming is used only where very

DIWAKAR EDUCATION HUB

efficient control over processor operations is needed. It requires knowledge of a particular computer's instruction set, however. Historically, most programs have been written in "higher-level" languages such as COBOL, FORTRAN, PL/I, and C. These languages are easier to learn and faster to write programs with than assembler language. The program that processes the source code written in these languages is called a compiler. Like the assembler, a compiler takes higher-level language statements and reduces them to machine code.

Program Loops

Loops are among the most basic and powerful of programming concepts. A loop in a computer program is an instruction that repeats until a specified condition is reached. In a loop structure, the loop asks a question. If the answer requires action, it is executed. The same question is asked again and again until no further action is required. Each time the question is asked is called an iteration.

A computer programmer who needs to use the same lines of code many times in a program can use a loop to save time.

Just about every programming language includes the concept of a loop. High-level programs accommodate several types of loops. C, C++, and C# are all high-level computer programs and have the capacity to use several types of loops.

Types of Loops

- A **for** loop is a loop that runs for a preset number of times.
- A **while** loop is a loop that is repeated as long as an expression is true. An expression is a statement that has a value.
- A **do while** loop or **repeat until** loop repeats until an expression becomes false.
- An **infinite** or **endless** loop is a loop that repeats indefinitely because it has no terminating condition, the exit condition is never met or the loop is instructed to start over from the beginning. Although it is possible for a

DIWAKAR EDUCATION HUB

programmer to intentionally use an infinite loop, they are often mistakes made by new programmers.

- A **nested** loop appears inside any other **for**, **while** or **do while** loop.

A goto statement can create a loop by jumping backward to a label, although this is generally discouraged as a bad programming practice. For some complex code, it allows a jump to a common exit point that simplifies the code.

Loop Control Statements

A statement that alters the execution of a loop from its designated sequence is a loop control statement. C#, for example, provides two loop control statements.

- A **break** statement inside a loop terminates the loop immediately.
- A **continue** statement jumps to the next iteration of the loop, skipping any code in between.

Basic Structures of Computer Programming

Loop, selection, and sequence are the three basic structures of computer programming. These three logic structures are used in combination to form algorithms for solving any logic problem. This process is called structured programming.

Subroutine

A set of Instructions which are used repeatedly in a program can be referred to as Subroutine. Only one copy of this Instruction is stored in the memory. When a Subroutine is required it can be called many times during the Execution of a Particular program. A call Subroutine Instruction calls the Subroutine. Care Should be taken while returning a Subroutine as Subroutine can be called from a different place from the memory.

The content of the PC must be Saved by the call Subroutine Instruction to make a correct return to the calling program.

DIWAKAR EDUCATION HUB

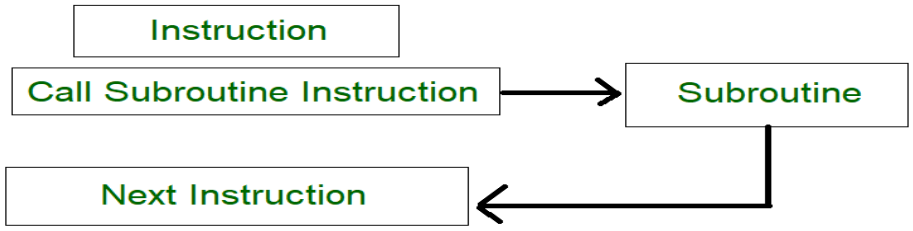


Figure – Process of subroutine in a program

Subroutine linkage method is a way in which computer call and return the Subroutine. The simplest way of Subroutine linkage is saving the return address in a specific location, such as register which can be called as link register call Subroutine.

2. Subroutine Nesting –

Subroutine nesting is a common Programming practice In which one Subroutine call another Subroutine.

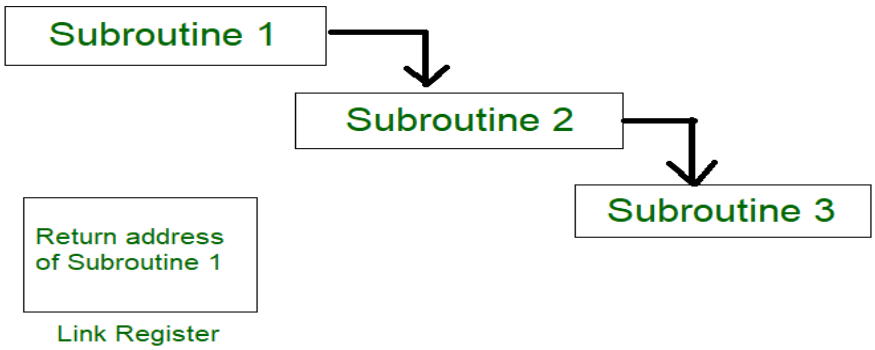


Figure – Subroutine calling another subroutine

From the above figure, assume that when Subroutine 1 calls Subroutine 2 the return address of Subroutine 2 should be saved somewhere. So if link register stores return address of Subroutine 1 this will be (destroyed/overwritten) by return address of Subroutine 2. As the last Subroutine called is the first one to be

returned (Last in first out format). So stack data structure is the most efficient way to store the return addresses of the Subroutines.

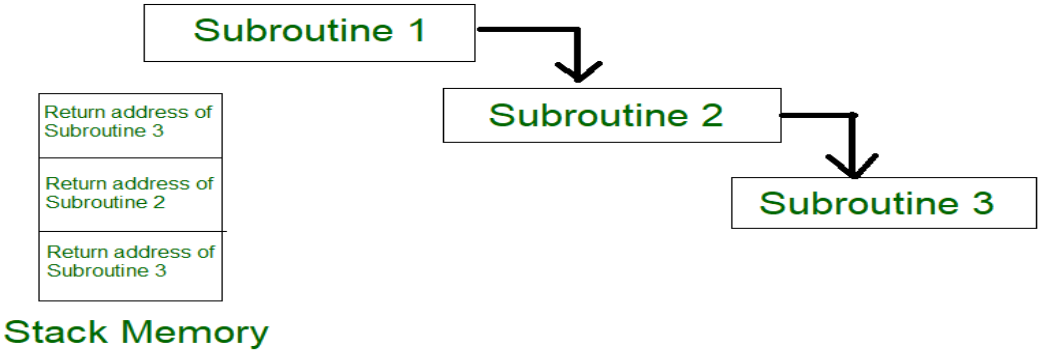


Figure – Return address of subroutine is stored in stack memory

3. Stack memory –

Stack is a basic data structure which can be implemented anywhere in the memory. It can be used to store variables which may be required afterwards in the program Execution. In a stack, the first data put will be last to get out of a stack. So the last data added will be the First one to come out of the stack (last in first out).

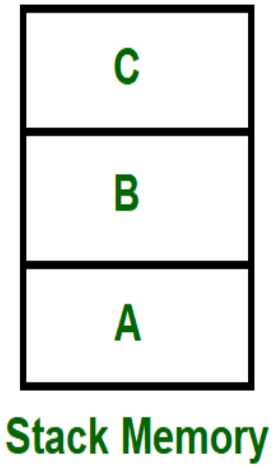


Figure – Stack memory having data A, B & C

DIWAKAR EDUCATION HUB

So from the diagram above first A is added then B & C. While removing first C is Removed then B & A.

Design of Control Unit

The Control Unit is classified into two major categories:

1. Hardwired Control
2. Microprogrammed Control

Hardwired Control Unit:

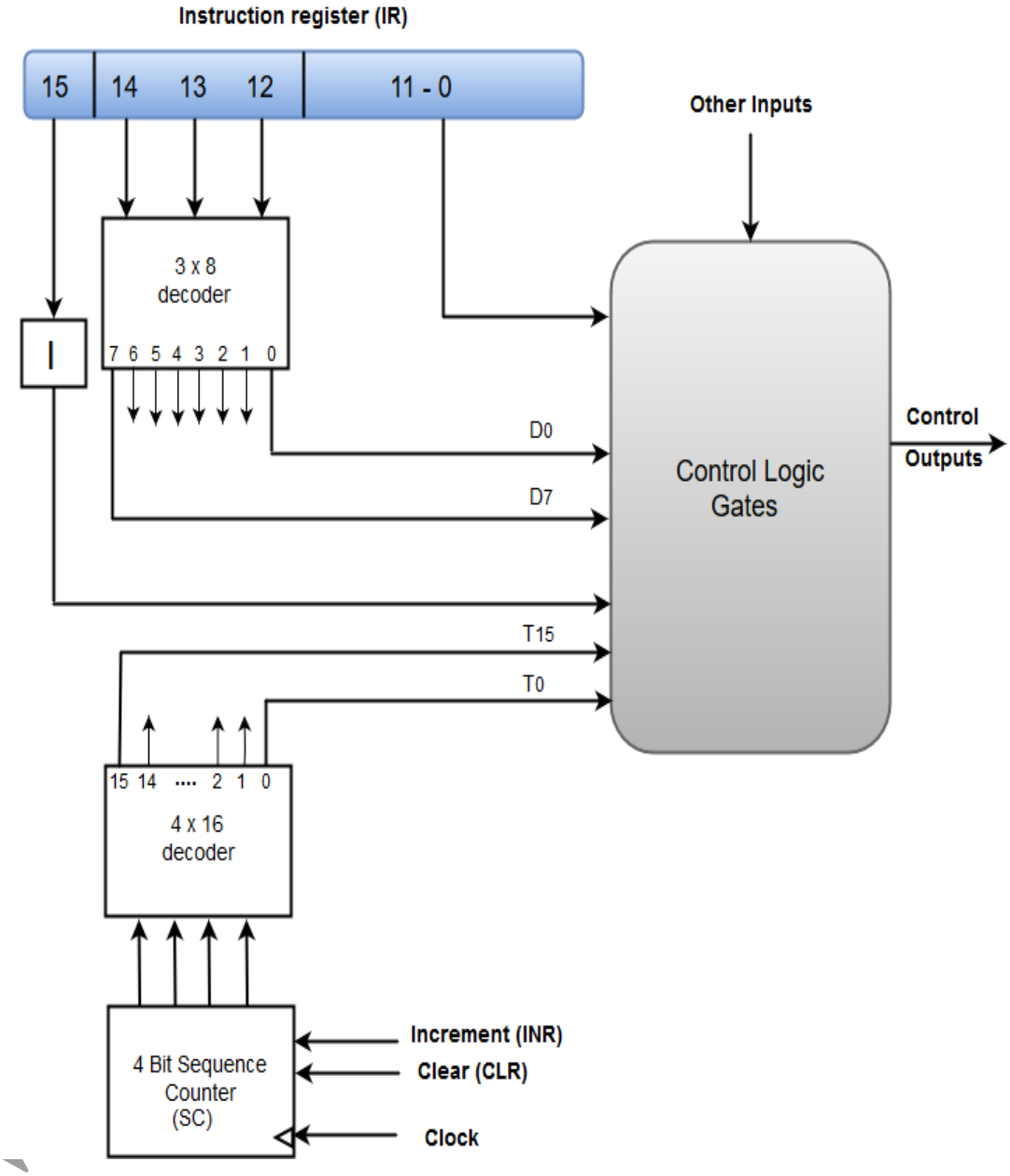
When the control signals are generated by hardware using conventional logic design techniques, the control unit is said to be hardwired.

The Hardwired Control organization involves the control logic to be implemented with gates, flip-flops, decoders, and other digital circuits.

The following image shows the block diagram of a Hardwired Control organization.

DIWAKAR EDUCATION HUB

Control Unit of a Basic Computer:



- A Hard-wired Control consists of two decoders, a sequence counter, and a number of logic gates.
- An instruction fetched from the memory unit is placed in the instruction register (IR).

- The component of an instruction register includes; I bit, the operation code, and bits 0 through 11.
- The operation code in bits 12 through 14 are coded with a 3 x 8 decoder.
- The outputs of the decoder are designated by the symbols D0 through D7.
- The operation code at bit 15 is transferred to a flip-flop designated by the symbol I.
- The operation codes from Bits 0 through 11 are applied to the control logic gates.
- The Sequence counter (SC) can count in binary from 0 through 15.

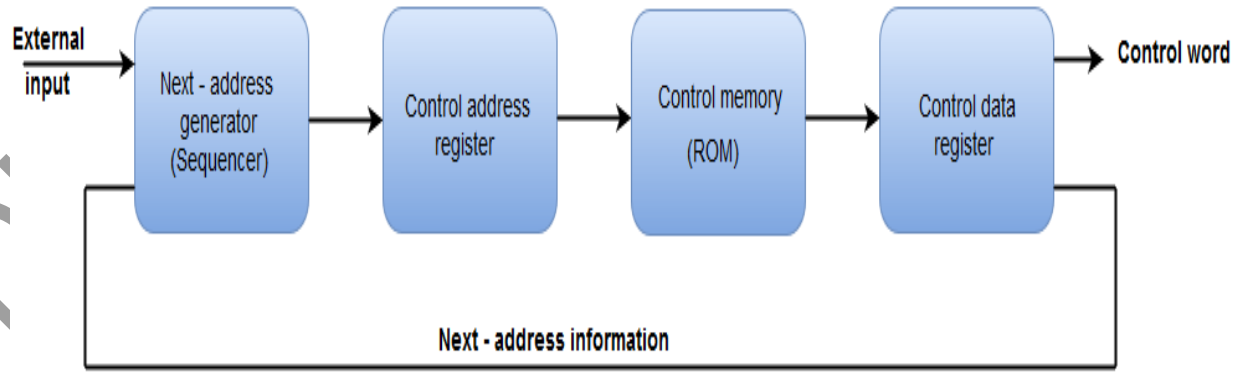
Micro-programmed Control

The Microprogrammed Control organization is implemented by using the programming approach.

In Microprogrammed Control, the micro-operations are performed by executing a program consisting of micro-instructions.

The following image shows the block diagram of a Microprogrammed Control organization.

Microprogrammed Control Unit of a Basic Computer:



EDUCATION HUB

- The Control memory address register specifies the address of the micro-instruction.
- The Control memory is assumed to be a ROM, within which all control information is permanently stored.
- The control register holds the microinstruction fetched from the memory.
- The micro-instruction contains a control word that specifies one or more micro-operations for the data processor.
- While the micro-operations are being executed, the next address is computed in the next address generator circuit and then transferred into the control address register to read the next microinstruction.
- The next address generator is often referred to as a micro-program sequencer, as it determines the address sequence that is read from control memory.

Difference between Hardwired Control and Microprogrammed Control

Hardwired Control	**Microprogrammed Control**
Technology is circuit based.	Technology is software based.
It is implemented through flip-flops, gates, decoders etc.	Microinstructions generate signals to control the execution of instructions.
Fixed instruction format.	Variable instruction format (16-64 bits per instruction).
Instructions are register based.	Instructions are not register based.
ROM is not used.	ROM is used.
It is used in RISC.	It is used in CISC.
Faster decoding.	Slower decoding.

DIWAKAR EDUCATION HUB

Difficult to modify.	Easily modified.
Chip area is less.	Chip area is large.

Dynamic microprogramming:

A more advanced development known as dynamic microprogramming permits a microprogram to be loaded initially from an auxiliary memory such as a magnetic disk. Control units that use dynamic microprogramming employ a writable control memory. This type of memory can be used for writing.

Control Memory:

Control Memory is the storage in the microprogrammed control unit to store the microprogram.

Writeable Control Memory:

Control Storage whose contents can be modified, allow the change in microprogram and Instruction set can be changed or modified is referred as Writeable Control Memory**.**

Control Word:

The control variables at any given time can be represented by a control word string of 1 's and 0's called a control word.

Microoperation, Microinstruction, Micro program, Microcode.

Microoperations:

In computer central processing units, micro-operations (also known as a micro-ops or μops) are detailed low-level instructions used in some designs to implement complex machine instructions (sometimes termed macro-instructions in this context).

Micro instruction:

DIWAKAR EDUCATION HUB

- A symbolic microprogram can be translated into its binary equivalent by means of an assembler.
- Each line of the assembly language microprogram defines a symbolic microinstruction.
- Each symbolic microinstruction is divided into five fields: label, microoperations, CD, BR, and AD.

Micro program:

- A sequence of microinstructions constitutes a microprogram.
- Since alterations of the microprogram are not needed once the control unit is in operation, the control memory can be a read-only memory (ROM).
- ROM words are made permanent during the hardware production of the unit.
- The use of a micro program involves placing all control variables in words of ROM for use by the control unit through successive read operations.
- The content of the word in ROM at a given address specifies a microinstruction.

Microcode:

- Microinstructions can be saved by employing subroutines that use common sections of microcode.
- For example, the sequence of micro operations needed to generate the effective address of the operand for an instruction is common to all memory reference instructions.
- This sequence could be a subroutine that is called from within many other routines to execute the effective address computation.

Organization of micro programmed control unit

DIWAKAR EDUCATION HUB

- The general configuration of a micro-programmed control unit is demonstrated in the block diagram of Figure 4.1.
- The control memory is assumed to be a ROM, within which all control information is permanently stored.

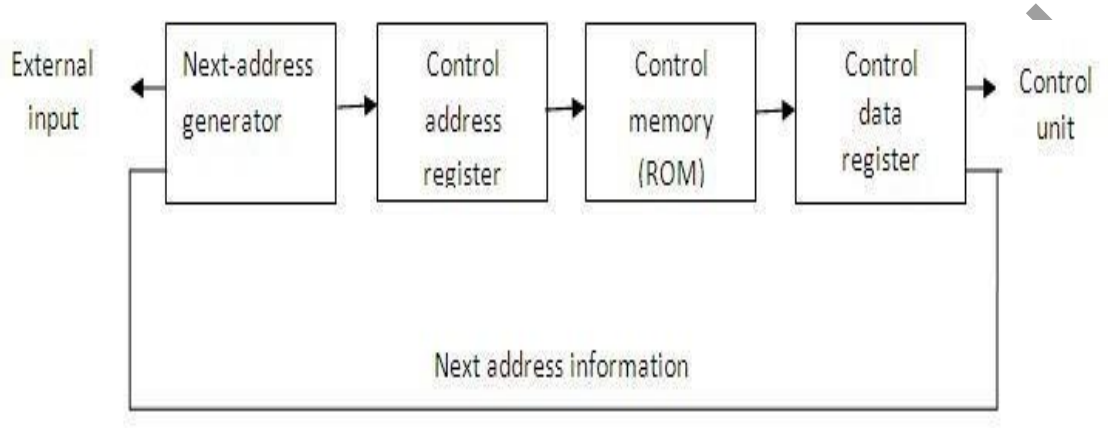


figure: Micro-programmed control organization

- The control memory address register specifies the address of the microinstruction, and the control data register holds the microinstruction read from memory.
- The microinstruction contains a control word that specifies one or more microoperations for the data processor. Once these operations are executed, the control must determine the next address.
- The location of the next microinstruction may be the one next in sequence, or it may be located somewhere else in the control memory.
- While the microoperations are being executed, the next address is computed in the next address generator circuit and then transferred into the control address register to read the next microinstruction.
- Thus a microinstruction contains bits for initiating microoperations in the data processor part and bits that determine the address sequence for the control memory.

DIWAKAR EDUCA JB

- The next address generator is sometimes called a **micro-program sequencer**, as it determines the address sequence that is read from control memory.
- Typical functions of a micro-program sequencer are incrementing the control address register by one, loading into the control address register an address from control memory, transferring an external address, or loading an initial address to start the control operations.
- The control data register holds the present microinstruction while the next address is computed and read from memory.
- The data register is sometimes called a **pipeline register**.
- It allows the execution of the microoperations specified by the control word simultaneously with the generation of the next microinstruction.
- This configuration requires a two-phase clock, with one clock applied to the address register and the other to the data register.
- The main advantage of the micro programmed control is the fact that once the hardware configuration is established; there should be no need for further hardware or wiring changes.
- If we want to establish a different control sequence for the system, all we need to do is specify a different set of microinstructions for control memory.

Address Sequencing

Microinstructions are stored in control memory in groups, with each group specifying a

routine.

To appreciate the address sequencing in a micro-program control unit, specify the steps that the control must undergo during the execution of a single computer instruction.

Step-1:

DIWAKAR EDUCATION HUB

- An initial address is loaded into the control address register when power is turned on in the computer.
- This address is usually the address of the first microinstruction that activates the instruction fetch routine.
- The fetch routine may be sequenced by incrementing the control address register through the rest of its microinstructions.
- At the end of the fetch routine, the instruction is in the instruction register of the computer.

Step-2:

- The control memory next must go through the routine that determines the effective address of the operand.
- A machine instruction may have bits that specify various addressing modes, such as indirect address and index registers.
- The effective address computation routine in control memory can be reached through a branch microinstruction, which is conditioned on the status of the mode bits of the instruction.
- When the effective address computation routine is completed, the address of the operand is available in the memory address register.

Step-3:

- The next step is to generate the microoperations that execute the instruction fetched from memory.
- The microoperation steps to be generated in processor registers depend on the operation code part of the instruction.
- Each instruction has its own micro-program routine stored in a given location of control memory.
- The transformation from the instruction code bits to an address in control memory where the routine is located is referred to as a **mapping** process.
- A mapping procedure is a rule that transforms the instruction code into a control memory address.

DIWAKAR EDUCATION HUB

Step-4:

- Once the required routine is reached, the microinstructions that execute the instruction may be sequenced by incrementing the control address register.
- Micro-programs that employ subroutines will require an external register for storing the return address.
- Return addresses cannot be stored in ROM because the unit has no writing capability.
- When the execution of the instruction is completed, control must return to the fetch routine.
- This is accomplished by executing an unconditional branch microinstruction to the first address of the fetch routine.

In summary, the address sequencing capabilities required in a control memory are:

1. Incrementing of the control address register.
2. Unconditional branch or conditional branch, depending on status bit conditions.
3. A mapping process from the bits of the instruction to an address for control memory.
4. A facility for subroutine call and return.

Selection of address for control memory

DIWAKAR EDUCATION HUB

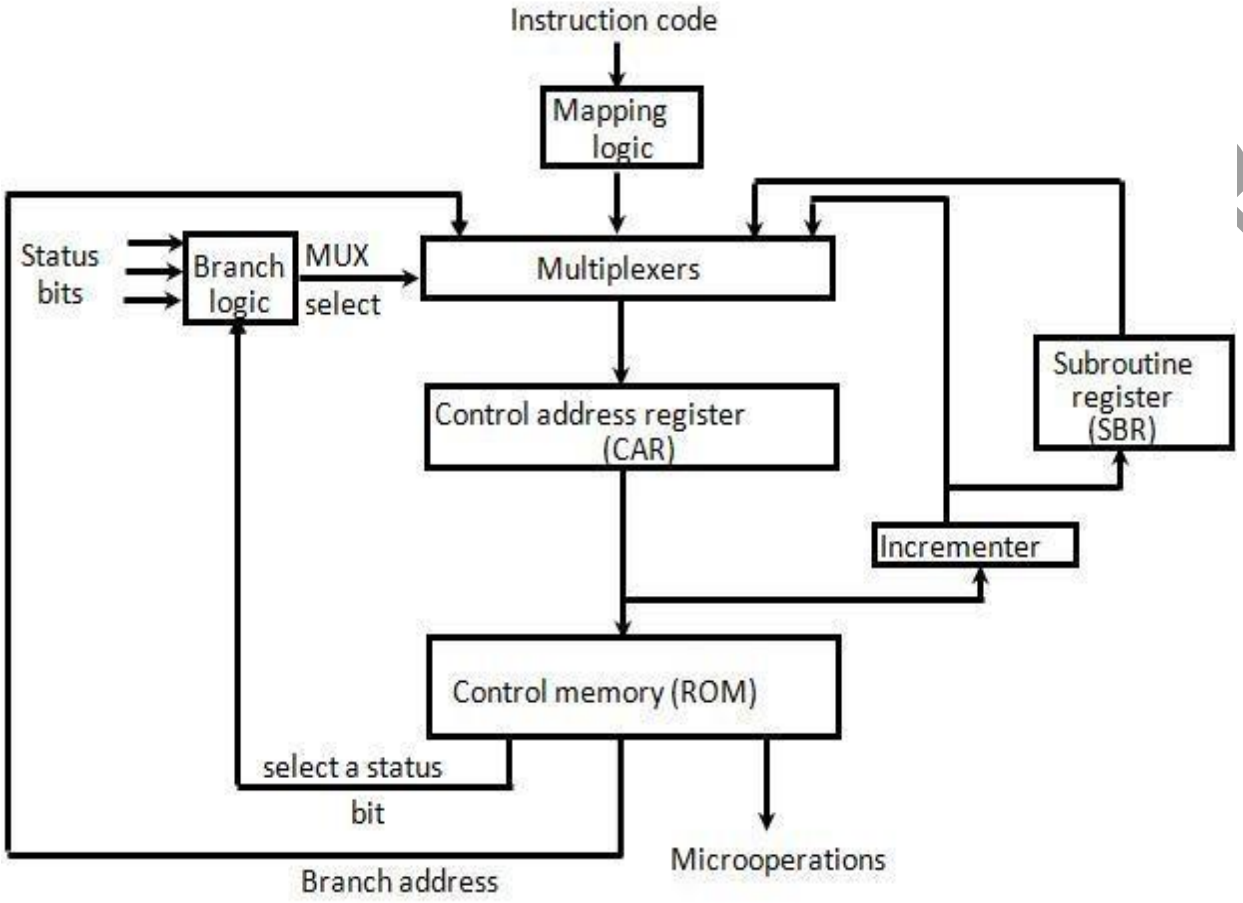


Figure 4.2: Selection of address for control memory

- Above figure 4.2 shows a block diagram of a control memory and the associated hardware needed for selecting the next microinstruction address.
- The microinstruction in control memory contains a set of bits to initiate microoperations in computer registers and other bits to specify the method by which the next address is obtained.
- The diagram shows four different paths from which the control address register (CAR) receives the address.
- The incrementer increments the content of the control address register by one, to select the next microinstruction in sequence.
- Branching is achieved by specifying the branch address in one of the fields of the microinstruction.

DIWAKAR EL

- Conditional branching is obtained by using part of the microinstruction to select a specific status bit in order to determine its condition.
- An external address is transferred into control memory via a mapping logic circuit.
- The return address for a subroutine is stored in a special register whose value is then used when the micro-program wishes to return from the subroutine.

Central Processing Unit (CPU)

The central processing unit (CPU) or processor, is the unit which performs most of the processing inside a computer. It processes all instructions received by software running on the PC and by other hardware components, and acts as a powerful calculator.

The CPU is placed into a specific square-shaped socket found on all motherboards by inserting its metallic connectors or pins found on the underside. Each socket is built with a specific pin layout to support only a specific type of processor.

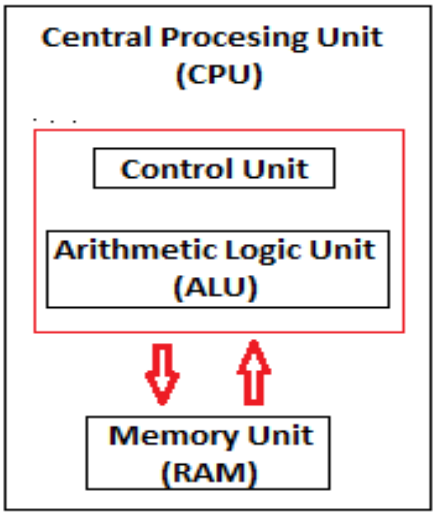
Since modern CPUs produce a lot of heat and are prone to overheating, they must be kept cool with appropriate fans or ventilation systems, and covered with heat sinks and thermal paste.

To control instructions and data flow to and from other parts of the computer, the CPU relies heavily on a chipset, which is a group of microchips located on the motherboard.

This term is also known as a central processor, microprocessor or chip

The central processing unit (CPU) has three components:

DIWAKAR EDUCATION HUB



Control Unit

The control unit extracts instructions from memory and decodes and executes them.

The control unit acts as an intermediary that decodes the instructions sent to the processor, tells the other units such as the Arithmetic Logic Unit (below) what to do by providing control signals, and then sends back the processed data back to memory.

Arithmetic Logic Unit (ALU)

An arithmetic logic unit (ALU) is a digital circuit inside the processor that handles arithmetic and logical operations by loading data from input registers.

After the control unit provides the ALU with the instruction on the operations that must be performed, the ALU completes them by connecting multiple transistors, and then stores the results in an output register.

The control unit will then move this data to memory.

To function properly, the CPU relies on the system clock, memory, secondary storage, and data and address buses.

DIWAKAR EDUCATION HUB

Smaller devices like mobile phones, calculators, held gaming systems, and tablets use smaller-sized processors known as ARM CPUs to accommodate their reduced size and space.

The CPU is the heart and brain of a computer. It receives data input, executes instructions, and processes information. It communicates with input/output (I/O) devices, which send and receive data to and from the CPU.

Additionally, the microprocessor has an internal bus for communication with the internal cache memory, called the backside bus. The main bus for data transfer to and from the CPU, memory, chipset, and AGP socket is called the front-side bus.

The CPU contains internal memory units, which are called registers. These registers contain data, instructions, counters and addresses used in the ALU's information processing.

Some computers utilize two or more processors. These consist of separate physical microprocessors located side by side on the same board or on separate boards. Each CPU has an independent interface, separate cache, and individual paths to the system front-side bus.

Multiple processors are ideal for intensive parallel tasks requiring multitasking. Multicore CPUs are also common, in which a single chip contains multiple CPUs.

Since the first microprocessor was released by Intel in November 1971, CPUs have increased their computing power severalfold.

The oldest Intel 4004 processor only performed 60,000 operations per second, while a modern Intel Pentium processor can perform about 188,000,000 instructions per second.

Memory or Storage Unit/ Registers

It is called Random access memory (RAM). It temporarily stores data, programs, and intermediate and final results of processing. So, it acts as a temporary storage area that holds the data temporarily, which is used to run the computer.

DIWAKAR EDUCATION HUB

CPU Clock Speed

The clock speed of a CPU or a processor refers to the number of instructions it can process in a second. It is measured in gigahertz. For example, a CPU with a clock speed of 4.0 GHz means it can process 4 billion instructions in a second.

Types of CPU:

CPUs are mostly manufactured by Intel and AMD, each of which manufactures its own types of CPUs. In modern times, there are lots of CPU types in the market. Some of the basic types of CPUs are described below:

Single Core CPU: Single Core is the oldest type of computer CPU, which was used in the 1970s. It has only one core to process different operations. It can start only one operation at a time; the CPU switches back and forth between different sets of data streams when more than one program runs. So, it is not suitable for multitasking as the performance will be reduced if more than one application runs. The performance of these CPUs is mainly dependent on the clock speed. It is still used in various devices, such as smartphones.

Dual Core CPU: As the name suggests, Dual Core CPU contains two cores in a single Integrated Circuit (IC). Although each core has its own controller and cache, they are linked together to work as a single unit and thus can perform faster than the single-core processors and can handle multitasking more efficiently than Single Core processors.

Quad Core CPU: This type of CPU comes with two dual-core processors in one integrated circuit (IC) or chip. So, a quad-core processor is a chip that contains four independent units called cores. These cores read and execute instructions of CPU. The cores can run multiple instructions simultaneously, thereby increases the overall speed for programs that are compatible with parallel processing.

Quad Core CPU uses a technology that allows four independent processing units (cores) to run in parallel on a single chip. Thus by integrating multiple cores in a single CPU, higher performance can be generated without boosting the clock

DIWAKAR EDUCATION HUB

speed. However, the performance increases only when the computer's software supports multiprocessing. The software which supports multiprocessing divides the processing load between multiple processors instead of using one processor at a time.

History of CPU:

Some of the important events in the development of CPU since its invention till date are as follows:

- In 1823, Baron Jons Jackob Berzelius discovered silicon that is the main component of CPU till date.
- In 1903, Nikola Tesla got gates or switches patented, which are electrical logic circuits.
- In December 1947, John Bardeen, William Shockley, and Walter Brattain invented the first transistor at the Bell Laboratories and got it patented in 1948.
- In 1958, the first working integrated circuit was developed by Robert Noyce and Jack Kilby.
- In 1960, IBM established the first mass-production facility for transistors in New York.
- In 1968, Robert Noyce and Gordon Moore founded Intel Corporation.
- AMD (Advanced Micro Devices) was founded in May 1969.
- In 1971, Intel introduced the first microprocessor, the Intel 4004, with the help of Ted Hoff.
- In 1972, Intel introduced the 8008 processor; in 1976, Intel 8086 was introduced, and in June 1979, Intel 8088 was released.
- In 1979, a 16/32-bit processor, the Motorola 68000, was released. Later, it was used as a processor for the Apple Macintosh and Amiga computers.
- In 1987, Sun introduced the SPARC processor.
- In March 1991, AMD introduced the AM386 microprocessor family.

DIWAKAR EDUCATION HUB

- In March 1993, Intel released the Pentium processor. In 1995, Cyrix introduced the Cx5x86 processor to give competition to Intel Pentium processors.
- In January 1999, Intel introduced the Celeron 366 MHz and 400 MHz processors.
- In April 2005, AMD introduced its first dual-core processor.
- In 2006, Intel introduced the Core 2 Duo processor.
- In 2007, Intel introduced different types of Core 2 Quad processors.
- In April 2008, Intel introduced the first series of Intel Atom processors, the Z5xx series. They were single-core processors with a 200 MHz GPU.
- In September 2009, Intel released the first Core i5 desktop processor with four cores.
- In January 2010, Intel released many processors such as Core 2 Quad processor Q9500, first Core i3 and i5 mobile processors, first Core i3 and i5 desktop processors. In the same year in July, it released the first Core i7 desktop processor with six cores.
- In June 2017, Intel introduced the first Core i9 desktop processor.
- In April 2018, Intel released the first Core i9 mobile processor.

General Register based CPU Organization

When we are using multiple general purpose registers, instead of single accumulator register, in the CPU Organization then this type of organization is known as General register based CPU Organization. In this type of organization, computer uses two or three address fields in their instruction format. Each address field may specify a general register or a memory word.If many CPU registers are available for heavily used variables and intermediate results, we can avoid memory references much of the time, thus vastly increasing program execution speed, and reducing program size.

For example:

MULT R1, R2, R3

DIWAKAR EDUCATION HUB

This is an instruction of an arithmatic multiplication written in assembly language. It uses three address fields R1, R2 and R3. The meaning of this instruction is:

R1 <-- R2 * R3

This instruction also can be written using only two address fields as:

MULT R1, R2

In this instruction, the destination register is the same as one of the source registers. This means the operation

R1 <-- R1 * R2

The use of large number of registers results in short program with limited instructions.

Some examples of General register based CPU Organization are **IBM 360 and PDP- 11**.

The advantages of General register based CPU organization –

- Efficiency of CPU increases as there are large number of registers are used in this organization.
- Less memory space is used to store the program since the instructions are written in compact way.

The disadvantages of General register based CPU organization –

- Care should be taken to avoid unnecessary usage of registers. Thus, compilers need to be more intelligent in this aspect.
- Since large number of registers are used, thus extra cost is required in this organization.

General register CPU organisation of two type:

1. Register-memory reference architecture (CPU with less register)– In this organisation Source 1 is always required in register, source 2 can be

DIWAKAR EDUCATION HUB

present either in register or in memory.Here two address instruction format is the compatible instruction format.

2. Register-register reference architecture(CPU with more register)– In this organisation ALU operations are performed only on a register data. So operands are required in the register. After manipulation result is also placed in register.Here three address instruction format is the compatible instruction format.

Stack based CPU Organization

The computers which use Stack-based CPU Organization are based on a data structure called **stack**. The stack is a list of data words. It uses **Last In First Out (LIFO)** access method which is the most popular access method in most of the CPU. A register is used to store the address of the topmost element of the stack which is known as **Stack pointer (SP)**. In this organisation, ALU operations are performed on stack data. It means both the operands are always required on the stack. After manipulation, the result is placed in the stack.

The main two operations that are performed on the operators of the stack are **Push** and **Pop**. These two operations are performed from one end only.

1. **Push –**
   This operation results in inserting one operand at the top of the stack and it decrease the stack pointer register. The format of the PUSH instruction is:

```
PUSH
```

It inserts the data word at specified address to the top of the stack. It can be implemented as:

```
//decrement SP by 1

SP <-- SP - 1

//store the content of specified memory address
```

DIWAKAR EDUCATION HUB

```
//into SP; i.e, at top of stack

SP <-- (memory address)
```

2. **Pop –**
   This operation results in deleting one operand from the top of the stack and it increase the stack pointer register. The format of the POP instruction is:

```
POP
```

It deletes the data word at the top of the stack to the specified address. It can be implemented as:

```
//transfer the content of  SP (i.e, at top most data)

//into specified memory location

(memory address) <-- SP

//increment SP by 1

SP <-- SP + 1
```

Operation type instruction does not need the address field in this CPU organization. This is because the operation is performed on the two operands that are on the top of the stack. For example:

```
SUB
```

This instruction contains the opcode only with no address field. It pops the two top data from the stack, subtracting the data, and pushing the result into the stack at the top.

PDP-11, Intel's 8085 and HP 3000 are some of the examples of the stack organized computers.

The advantages of Stack based CPU organization –

DIWAKAR EDUCATION HUB

- Efficient computation of complex arithmetic expressions.
- Execution of instructions is fast because operand data are stored in consecutive memory locations.
- Length of instruction is short as they do not have address field.

The disadvantages of Stack based CPU organization –

- The size of the program increases.

Note:*Stack based CPU organisation uses zero address instruction.*

Computer Instructions

The basic computer has three instruction code formats. The **Operation code** (opcode) part of the instruction contains 3 bits and remaining 13 bits depends upon the operation code encountered.

There are three types of formats:

1. Memory Reference Instruction

It uses 12 bits to specify the address and 1 bit to specify the addressing mode (**I**). **I** is equal to 0 for direct address and 1 for indirect address.

2. Register Reference Instruction

These instructions are recognized by the opcode 111 with a 0 in the left most bit of instruction. The other 12 bits specify the operation to be executed.

3. Input-Output Instruction

These instructions are recognized by the operation code 111 with a 1 in the left most bit of instruction. The remaining 12 bits are used to specify the input-output operation.

Format of Instruction

DIWAKAR EDUCATION HUB

The format of an instruction is depicted in a rectangular box symbolizing the bits of an instruction. Basic fields of an instruction format are given below:

4. An operation code field that specifies the operation to be performed.
5. An address field that designates the memory address or register.
6. A mode field that specifies the way the operand of effective address is determined.

Computers may have instructions of different lengths containing varying number of addresses. The number of address field in the instruction format depends upon the internal organization of its registers.

Adressing Modes

The operation field of an instruction specifies the operation to be performed. This operation will be executed on some data which is stored in computer registers or the main memory. The way any operand is selected during the program execution is dependent on the addressing mode of the instruction. The purpose of using addressing modes is as follows:

3. To give the programming versatility to the user.
4. To reduce the number of bits in addressing field of instruction.

Types of Addressing Modes

Below we have discussed different types of addressing modes one by one:

Immediate Mode

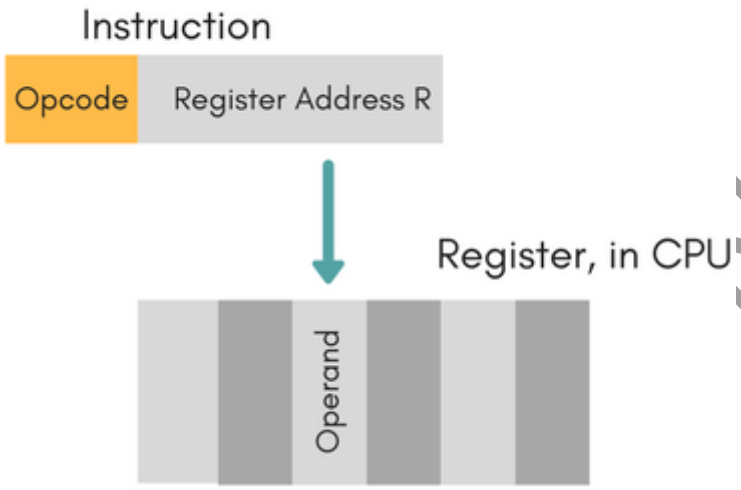
In this mode, the operand is specified in the instruction itself. An immediate mode instruction has an operand field rather than the address field.

For example: ADD 7, which says Add 7 to contents of accumulator. 7 is the operand here.

Register Mode

DIWAKAR EDUCATION HUB

In this mode the operand is stored in the register and this register is present in CPU. The instruction has the address of the Register where the operand is stored.



Advantages

- Shorter instructions and faster instruction fetch.
- Faster memory access to the operand(s)

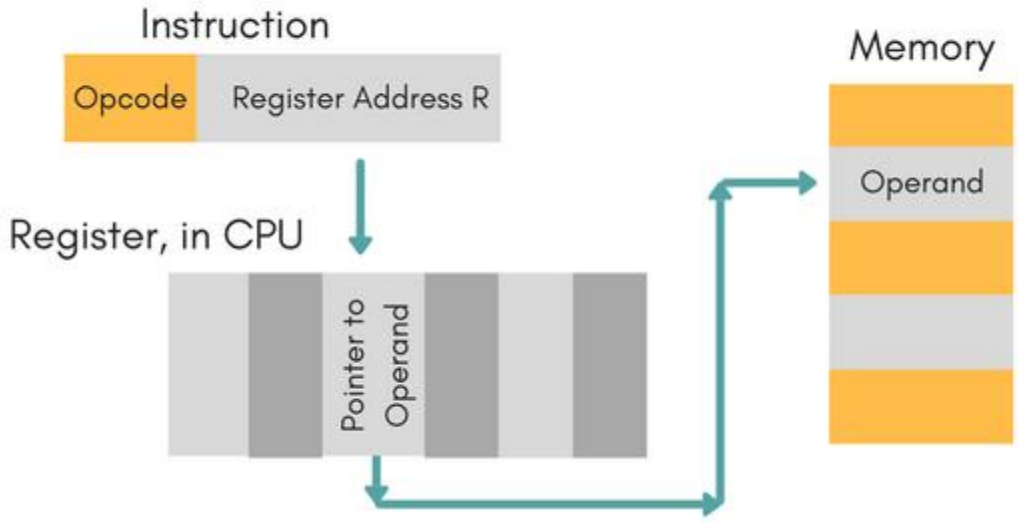
Disadvantages

- Very limited address space
- Using multiple registers helps performance but it complicates the instructions.

Register Indirect Mode

In this mode, the instruction specifies the register whose contents give us the address of operand which is in memory. Thus, the register contains the address of operand rather than the operand itself.

DIWAKAR EDUCATION HUB



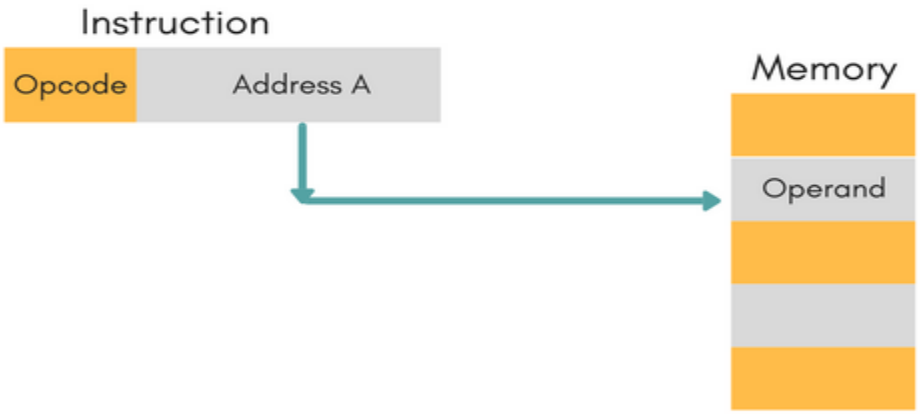
Auto Increment/Decrement Mode

In this the register is incremented or decremented after or before its value is used.

Direct Addressing Mode

In this mode, effective address of operand is present in instruction itself.

- Single memory reference to access data.
- No additional calculations to find the effective address of the operand.

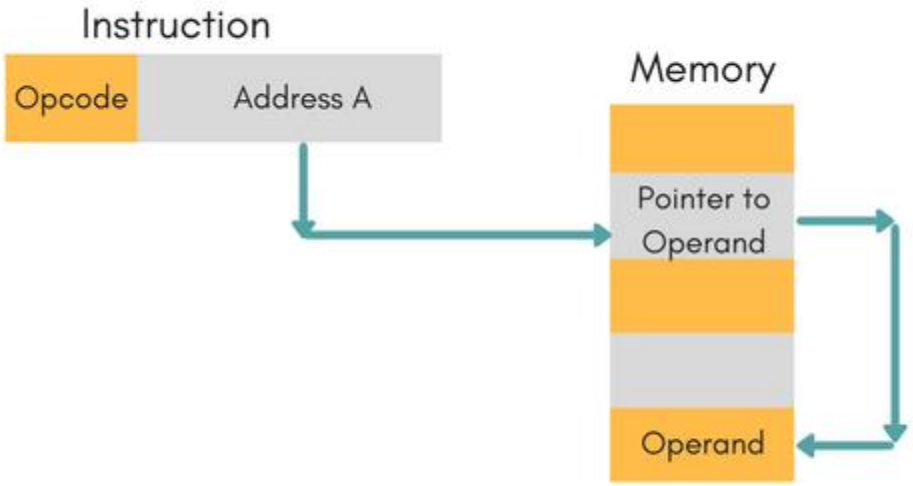


For Example: ADD R1, 4000 - In this the 4000 is effective address of operand.

NOTE: Effective Address is the location where operand is present.

Indirect Addressing Mode

In this, the address field of instruction gives the address where the effective address is stored in memory. This slows down the execution, as this includes multiple memory lookups to find the operand.

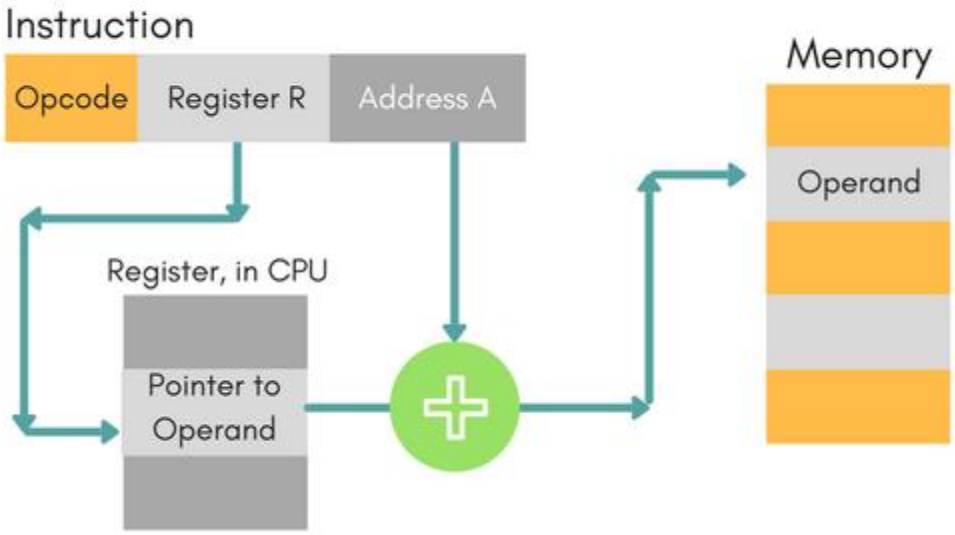


Displacement Addressing Mode

In this the contents of the indexed register is added to the Address part of the instruction, to obtain the effective address of operand.

EA = A + (R), In this the address field holds two values, A(which is the base value) and R(that holds the displacement), or vice versa.

DIWAKAR



Relative Addressing Mode

It is a version of Displacement addressing mode.

In this the contents of PC(Program Counter) is added to address part of instruction to obtain the effective address.

EA = A + (PC), where EA is effective address and PC is program counter.

The operand is A cells away from the current cell(the one pointed to by PC)

Base Register Addressing Mode

It is again a version of Displacement addressing mode. This can be defined as EA = A + (R), where A is displacement and R holds pointer to base address.

Stack Addressing Mode

In this mode, operand is at the top of the stack. For example: ADD, this instruction will POP top two items from the stack, add them, and will then PUSH the result to the top of the stack.

RISC Processor

DIWAKAR EDUCATION HUB

It is known as **Reduced Instruction Set Computer**. It is a type of microprocessor that has a limited number of instructions. They can execute their instructions very fast because instructions are very small and simple.

RISC chips require fewer transistors which make them cheaper to design and produce. In RISC, the instruction set contains simple and basic instructions from which more complex instruction can be produced. Most instructions complete in one cycle, which allows the processor to handle many instructions at same time.

In this instructions are register based and data transfer takes place from register to register.

CISC Processor

- It is known as **Complex Instruction Set Computer**.
- It was first developed by Intel.
- It contains large number of complex instructions.
- In this instructions are not register based.
- Instructions cannot be completed in one machine cycle.
- Data transfer is from memory to memory.
- Micro programmed control unit is found in CISC.
- Also they have variable instruction formats.

Difference Between CISC and RISC

Architectural Characterstics	**Complex Instruction Set Computer(CISC)**	**Reduced Instruction Set Computer(RISC)**
Instruction size and format	Large set of instructions with variable formats (16-64 bits per instruction).	Small set of instructions with fixed format (32 bit).

DIWAKAR EDUCATION HUB

Data transfer	Memory to memory.	Register to register.
CPU control	Most micro coded using control memory (ROM) but modern CISC use hardwired control.	Mostly hardwired without control memory.
Instruction type	Not register based instructions.	Register based instructions.
Memory access	More memory access.	Less memory access.
Clocks	Includes multi-clocks.	Includes single clock.
Instruction nature	Instructions are complex.	Instructions are reduced and simple.

Parallel Processing

Parallel processing can be described as a class of techniques which enables the system to achieve simultaneous data-processing tasks to increase the computational speed of a computer system.

A parallel processing system can carry out simultaneous data-processing to achieve faster execution time. For instance, while an instruction is being processed in the ALU component of the CPU, the next instruction can be read from memory.

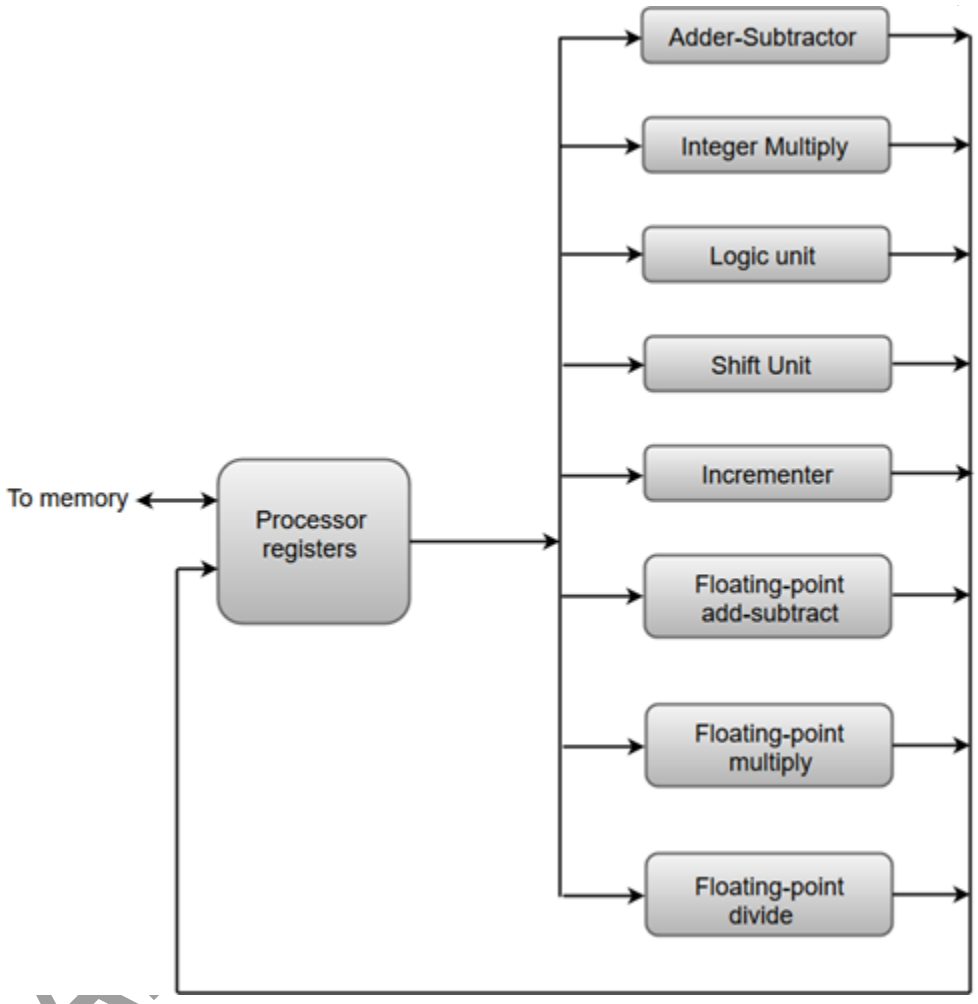
The primary purpose of parallel processing is to enhance the computer processing capability and increase its throughput, i.e. the amount of processing that can be accomplished during a given interval of time.

A parallel processing system can be achieved by having a multiplicity of functional units that perform identical or different operations simultaneously. The data can be distributed among various multiple functional units.

DIWAKAR EDUCATION HUB

The following diagram shows one possible way of separating the execution unit into eight functional units operating in parallel.

The operation performed in each functional unit is indicated in each block if the diagram:



- The adder and integer multiplier performs the arithmetic operation with integer numbers.
- The floating-point operations are separated into three circuits operating in parallel.
- The logic, shift, and increment operations can be performed concurrently on different data. All units are independent of each other,

so one number can be shifted while another number is being incremented.

Data Transfer Modes of a Computer System

According to the data transfer mode, computer can be divided into 4 major groups:

1. SISD
2. SIMD
3. MISD
4. MIMD

SISD (Single Instruction Stream, Single Data Stream)

It represents the organization of a single computer containing a control unit, processor unit and a memory unit. Instructions are executed sequentially. It can be achieved by pipelining or multiple functional units.

SIMD (Single Instruction Stream, Multiple Data Stream)

It represents an organization that includes multiple processing units under the control of a common control unit. All processors receive the same instruction from control unit but operate on different parts of the data.

They are highly specialized computers. They are basically used for numerical problems that are expressed in the form of vector or matrix. But they are not suitable for other types of computations

MISD (Multiple Instruction Stream, Single Data Stream)

It consists of a single computer containing multiple processors connected with multiple control units and a common memory unit. It is capable of processing several instructions over single data stream simultaneously. MISD structure is

DIWAKAR EDUCATION HUB

only of theoretical interest since no practical system has been constructed using this organization.

MIMD (Multiple Instruction Stream, Multiple Data Stream

It represents the organization which is capable of processing several programs at same time. It is the organization of a single computer containing multiple processors connected with multiple control units and a shared memory unit. The shared memory unit contains multiple modules to communicate with all processors simultaneously. Multiprocessors and multicomputer are the examples of MIMD. It fulfills the demand of large scale computations.

Pipelining

The term Pipelining refers to a technique of decomposing a sequential process into sub-operations, with each sub-operation being executed in a dedicated segment that operates concurrently with all other segments.

The most important characteristic of a pipeline technique is that several computations can be in progress in distinct segments at the same time. The overlapping of computation is made possible by associating a register with each segment in the pipeline. The registers provide isolation between each segment so that each can operate on distinct data simultaneously.

The structure of a pipeline organization can be represented simply by including an input register for each segment followed by a combinational circuit.

an example of combined multiplication and addition operation to get a better understanding of the pipeline organization.

The combined multiplication and addition operation is done with a stream of numbers such as:

$A_i * B_i + C_i$ for i = 1, 2, 3, ..., 7

DIWAKAR EDUCATION HUB

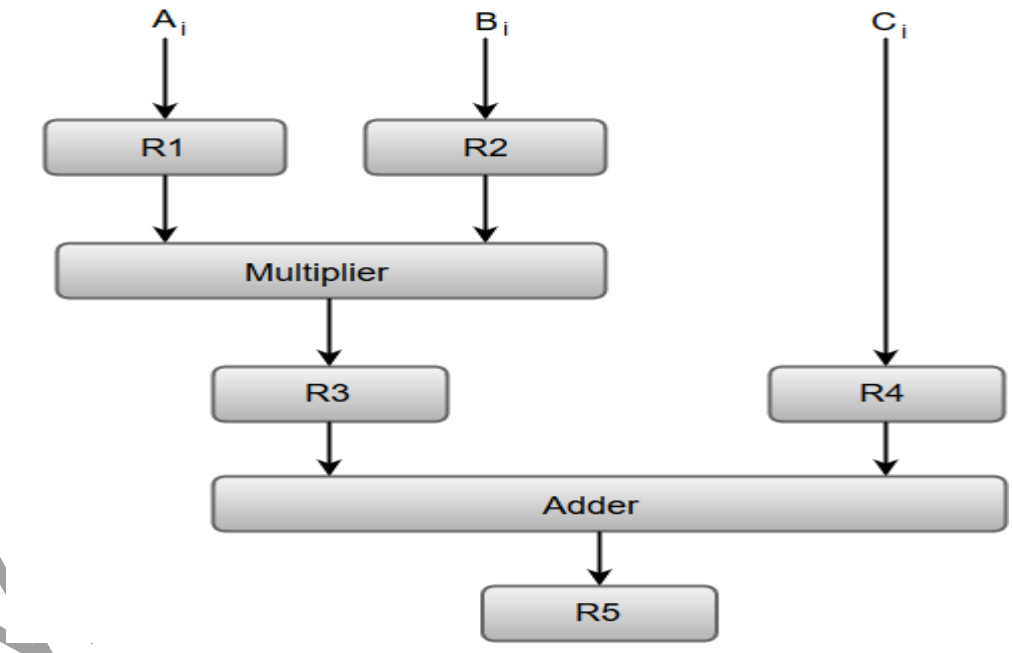
The operation to be performed on the numbers is decomposed into sub-operations with each sub-operation to be implemented in a segment within a pipeline.

The sub-operations performed in each segment of the pipeline are defined as:

$R1 \leftarrow A_i$, $R2 \leftarrow B_i$	Input A_i, and B_i
$R3 \leftarrow R1 * R2$, $R4 \leftarrow C_i$	Multiply, and input C_i
$R5 \leftarrow R3 + R4$	Add C_i to product

The following block diagram represents the combined as well as the sub-operations performed in each segment of the pipeline.

Pipeline Processing:



Registers R1, R2, R3, and R4 hold the data and the combinational circuits operate in a particular segment.

The output generated by the combinational circuit in a given segment is applied as an input register of the next segment. For instance, from the block diagram, we

can see that the register R3 is used as one of the input registers for the combinational adder circuit.

In general, the pipeline organization is applicable for two areas of computer design which includes:

1. Arithmetic Pipeline
2. Instruction Pipeline

Arithmetic Pipeline

Arithmetic Pipelines are mostly used in high-speed computers. They are used to implement floating-point operations, multiplication of fixed-point numbers, and similar computations encountered in scientific problems.

To understand the concepts of arithmetic pipeline in a more convenient way, an example of a pipeline unit for floating-point addition and subtraction.

The inputs to the floating-point adder pipeline are two normalized floating-point binary numbers defined as:

$$X = A * 2^a = 0.9504 * 10^3$$

$$Y = B * 2^b = 0.8200 * 10^2$$

Where **A** and **B** are two fractions that represent the mantissa and **a** and **b** are the exponents.

The combined operation of floating-point addition and subtraction is divided into four segments. Each segment contains the corresponding suboperation to be performed in the given pipeline. The suboperations that are shown in the four segments are:

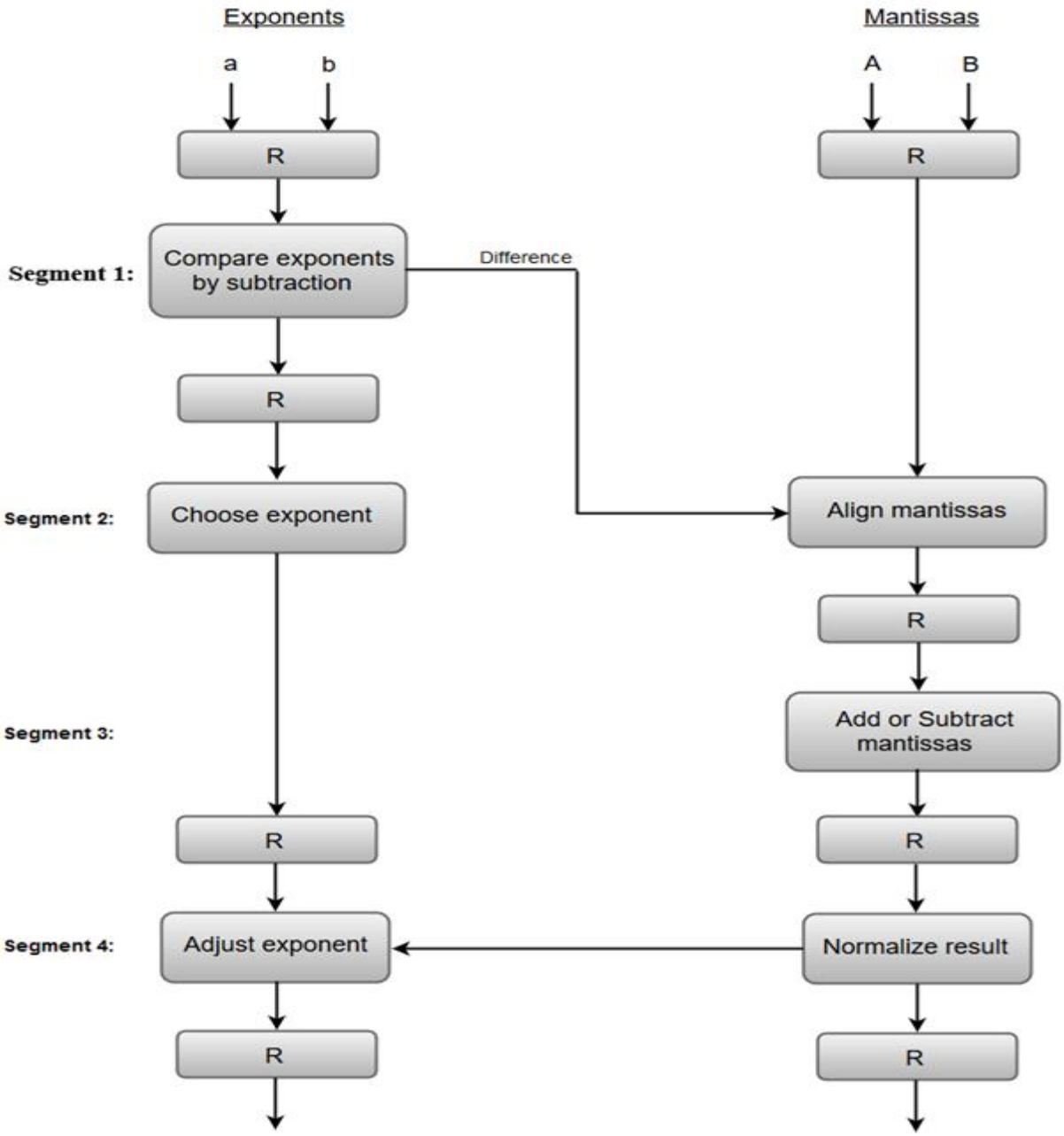
1. Compare the exponents by subtraction.
2. Align the mantissas.

DIWAKAR EDUCATION HUB

3. Add or subtract the mantissas.
4. Normalize the result.

The following block diagram represents the suboperations performed in each segment of the pipeline.

Pipeline organization for floating point addition and subtraction:



Note: Registers are placed after each suboperation to store the intermediate results.

1. Compare exponents by subtraction:

The exponents are compared by subtracting them to determine their difference. The larger exponent is chosen as the exponent of the result.

The difference of the exponents, i.e., **3** - **2** = **1** determines how many times the mantissa associated with the smaller exponent must be shifted to the right.

2. Align the mantissas:

The mantissa associated with the smaller exponent is shifted according to the difference of exponents determined in segment one.

$$X = 0.9504 * 10^3$$

$$Y = 0.08200 * 10^3$$

3. Add mantissas:

The two mantissas are added in segment three.

$$Z = X + Y = 1.0324 * 10^3$$

4. Normalize the result:

After normalization, the result is written as:

$$Z = 0.1324 * 10^4$$

Instruction Pipeline

Pipeline processing can occur not only in the data stream but in the instruction stream as well.

Most of the digital computers with complex instructions require instruction pipeline to carry out operations like fetch, decode and execute instructions.

DIWAKAR EDUCATION HUB

In general, the computer needs to process each instruction with the following sequence of steps.

1. Fetch instruction from memory.
2. Decode the instruction.
3. Calculate the effective address.
4. Fetch the operands from memory.
5. Execute the instruction.
6. Store the result in the proper place.

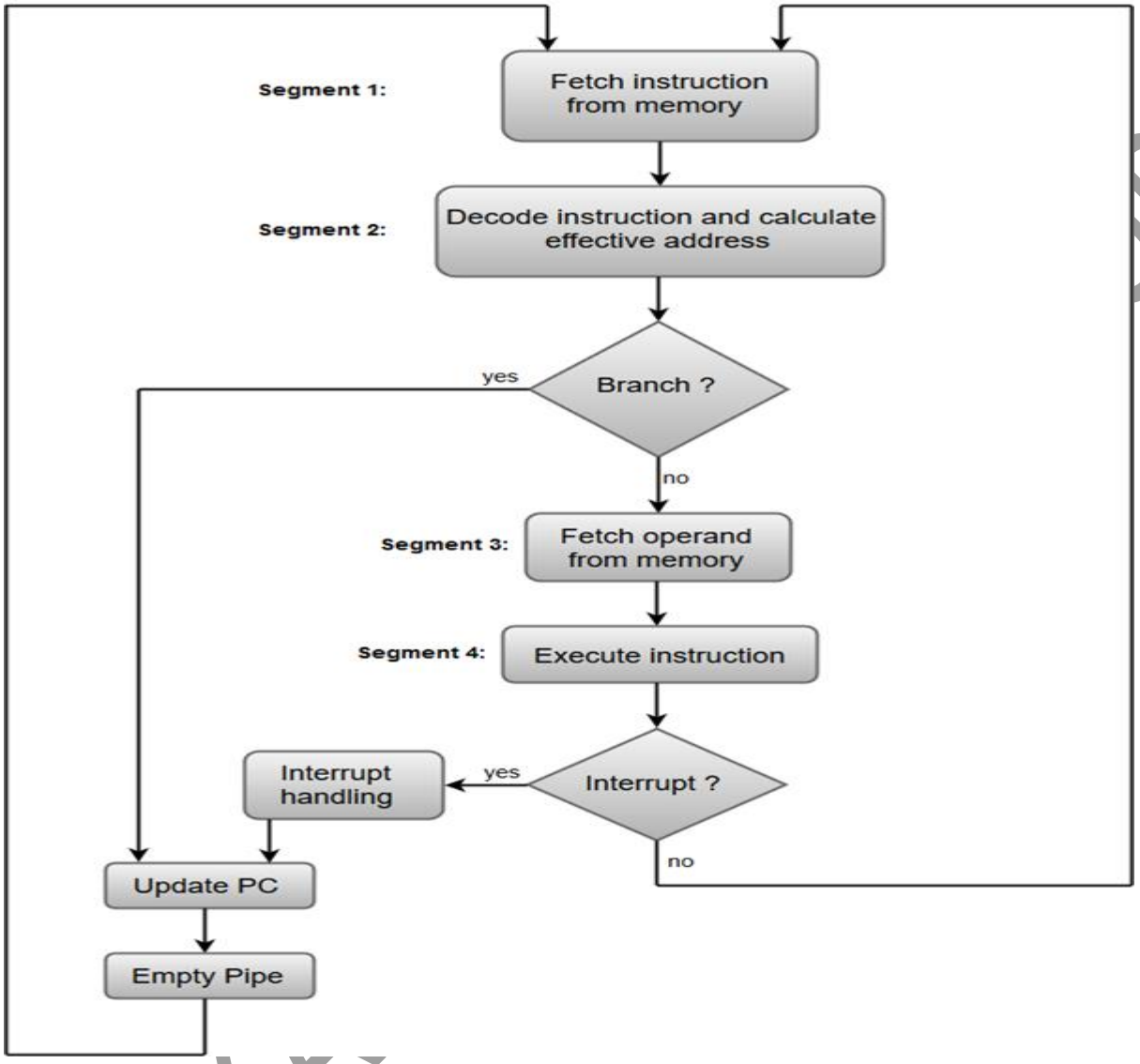
Each step is executed in a particular segment, and there are times when different segments may take different times to operate on the incoming information. Moreover, there are times when two or more segments may require memory access at the same time, causing one segment to wait until another is finished with the memory.

The organization of an instruction pipeline will be more efficient if the instruction cycle is divided into segments of equal duration. One of the most common examples of this type of organization is a **Four-segment instruction pipeline.**

A **four-segment instruction** pipeline combines two or more different segments and makes it as a single one. For instance, the decoding of the instruction can be combined with the calculation of the effective address into one segment.

The following block diagram shows a typical example of a four-segment instruction pipeline. The instruction cycle is completed in four segments.

DIWAKAR EDUCATION HUB



Segment 1:

The instruction fetch segment can be implemented using first in, first out (FIFO) buffer.

Segment 2:

The instruction fetched from memory is decoded in the second segment, and eventually, the effective address is calculated in a separate arithmetic circuit.

Segment 3:

An operand from memory is fetched in the third segment.

Segment 4:

The instructions are finally executed in the last segment of the pipeline organization.

Advantages of Pipelining

1. The cycle time of the processor is reduced.
2. It increases the throughput of the system
3. It makes the system reliable.

Disadvantages of Pipelining

1. The design of pipelined processor is complex and costly to manufacture.
2. The instruction latency is more.

Vector(Array) Processing

There is a class of computational problems that are beyond the capabilities of a conventional computer. These problems require vast number of computations on multiple data items, that will take a conventional computer(with scalar processor) days or even weeks to complete.

Such complex instructions, which operates on multiple data at the same time, requires a better way of instruction execution, which was achieved by Vector processors.

Scalar CPUs can manipulate one or two data items at a time, which is not very efficient. Also, simple instructions like **ADD A to B, and store into C** are not practically efficient.

Addresses are used to point to the memory location where the data to be operated will be found, which leads to added overhead of data lookup. So until

DIWAKAR EDUCATION HUB

the data is found, the CPU would be sitting ideal, which is a big performance issue.

Hence, the concept of **Instruction Pipeline** comes into picture, in which the instruction passes through several sub-units in turn. These sub-units perform various independent functions, **for example**: the **first** one decodes the instruction, the **second** sub-unit fetches the data and the **third** sub-unit performs the math itself. Therefore, while the data is fetched for one instruction, CPU does not sit idle, it rather works on decoding the next instruction set, ending up working like an assembly line.

Vector processor, not only use Instruction pipeline, but it also pipelines the data, working on multiple data at the same time.

A normal scalar processor instruction would be ADD A, B, which leads to addition of two operands, but what if we can instruct the processor to ADD a group of numbers(from 0 to n memory location) to another group of numbers(lets say, n to k memory location). This can be achieved by vector processors.

In vector processor a single instruction, can ask for multiple data operations, which saves time, as instruction is decoded once, and then it keeps on operating on different data items.

Applications of Vector Processors

Computer with vector processing capabilities are in demand in specialized applications. The following are some areas where vector processing is used:

1. Petroleum exploration.
2. Medical diagnosis.
3. Data analysis.
4. Weather forecasting.
5. Aerodynamics and space flight simulations.

DIWAKAR EDUCATION HUB

6. Image processing.
7. Artificial intelligence.

Vector(Array) Processor and its Types

Array processors are also known as multiprocessors or vector processors. They perform computations on large arrays of data. Thus, they are used to improve the performance of the computer.

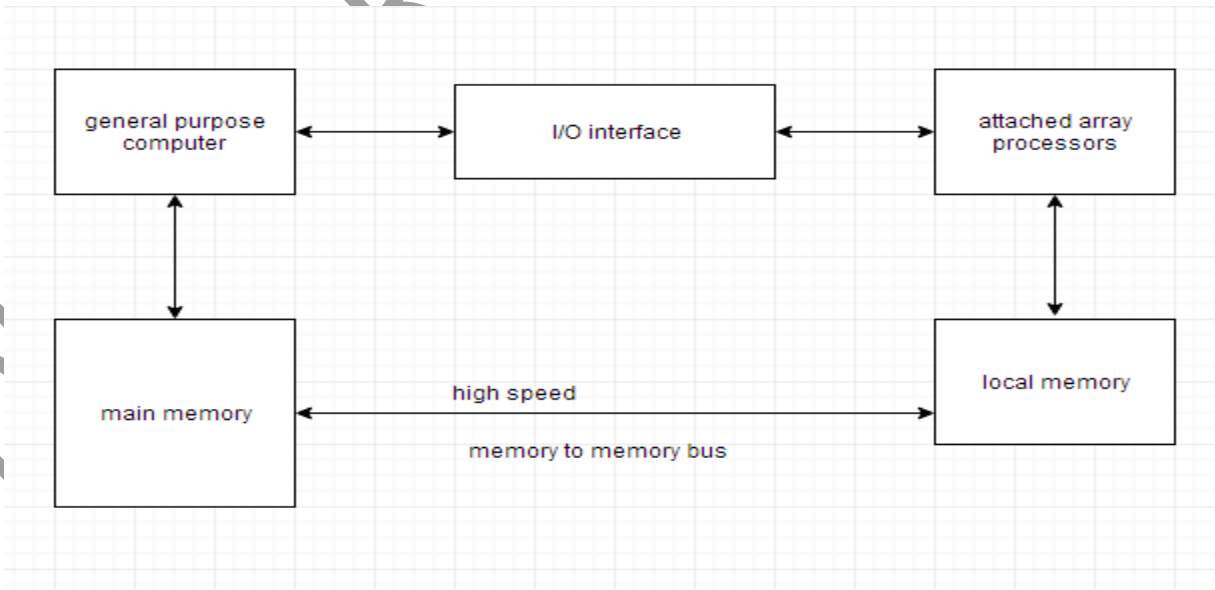
Types of Array Processors

There are basically two types of array processors:

1. Attached Array Processors
2. SIMD Array Processors

Attached Array Processors

An attached array processor is a processor which is attached to a general purpose computer and its purpose is to enhance and improve the performance of that computer in numerical computational tasks. It achieves high performance by means of parallel processing with multiple functional units.



EDUCATION HUB

SIMD Array Processors

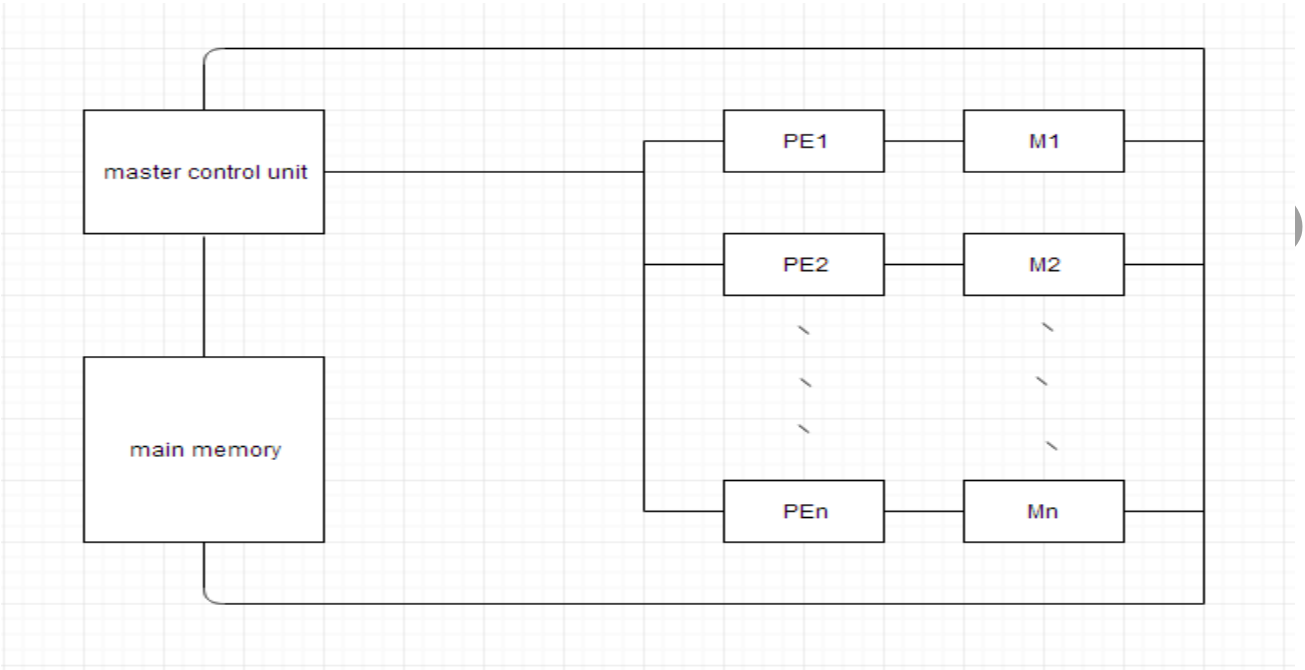
SIMD is the organization of a single computer containing multiple processors operating in parallel. The processing units are made to operate under the control of a common control unit, thus providing a single instruction stream and multiple data streams.

A general block diagram of an array processor is shown below. It contains a set of identical processing elements (PE's), each of which is having a local memory M. Each processor element includes an **ALU** and **registers**. The master control unit controls all the operations of the processor elements. It also decodes the instructions and determines how the instruction is to be executed.

The main memory is used for storing the program. The control unit is responsible for fetching the instructions. Vector instructions are send to all PE's simultaneously and results are returned to the memory.

The best known SIMD array processor is the **ILLIAC IV** computer developed by the **Burroughs corps**. SIMD processors are highly specialized computers. They are only suitable for numerical problems that can be expressed in vector or matrix form and they are not suitable for other types of computations.

DIWAKAR EDUCATION HUB



Why use the Array Processor

- Array processors increases the overall instruction processing speed.
- As most of the Array processors operates asynchronously from the host CPU, hence it improves the overall capacity of the system.
- Array Processors has its own local memory, hence providing extra memory for systems with low memory.

Input/Output Organisation

Input/Output Subsystem

The I/O subsystem of a computer provides an efficient mode of communication between the central system and the outside environment. It handles all the input-output operations of the computer system.

Peripheral Devices

Input or output devices that are connected to computer are called **peripheral devices**. These devices are designed to read information into or out of the

DIWAKAR EDUCA

memory unit upon command from the CPU and are considered to be the part of computer system. These devices are also called **peripherals**.

For example: *Keyboards*, *display units* and *printers* are common peripheral devices.

There are three types of peripherals:

1. **Input peripherals** : Allows user input, from the outside world to the computer. Example: Keyboard, Mouse etc.
2. **Output peripherals**: Allows information output, from the computer to the outside world. Example: Printer, Monitor etc
3. **Input-Output peripherals**: Allows both input(from outised world to computer) as well as, output(from computer to the outside world). Example: Touch screen etc.

Interfaces

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

There are two types of interface:

1. CPU Inteface
2. I/O Interface

Input-Output Interface

Peripherals connected to a computer need special communication links for interfacing with CPU. In computer system, there are special hardware components between the CPU and peripherals to control or manage the input-output transfers. These components are called **input-output interface units** because they provide communication links between processor bus and

DIWAKAR EDUCATION HUB

peripherals. They provide a method for transferring information between internal system and input-output devices.

Modes of I/O Data Transfer

Data transfer between the central unit and I/O devices can be handled in generally three types of modes which are given below:

1. Programmed I/O
2. Interrupt Initiated I/O
3. Direct Memory Access

Programmed I/O

Programmed I/O instructions are the result of I/O instructions written in computer program. Each data item transfer is initiated by the instruction in the program.

Usually the program controls data transfer to and from CPU and peripheral. Transferring data under programmed I/O requires constant monitoring of the peripherals by the CPU.

Interrupt Initiated I/O

In the programmed I/O method the CPU stays in the program loop until the I/O unit indicates that it is ready for data transfer. This is time consuming process because it keeps the processor busy needlessly.

This problem can be overcome by using **interrupt initiated I/O**. In this when the interface determines that the peripheral is ready for data transfer, it generates an interrupt. After receiving the interrupt signal, the CPU stops the task which it is processing and service the I/O transfer and then returns back to its previous processing task.

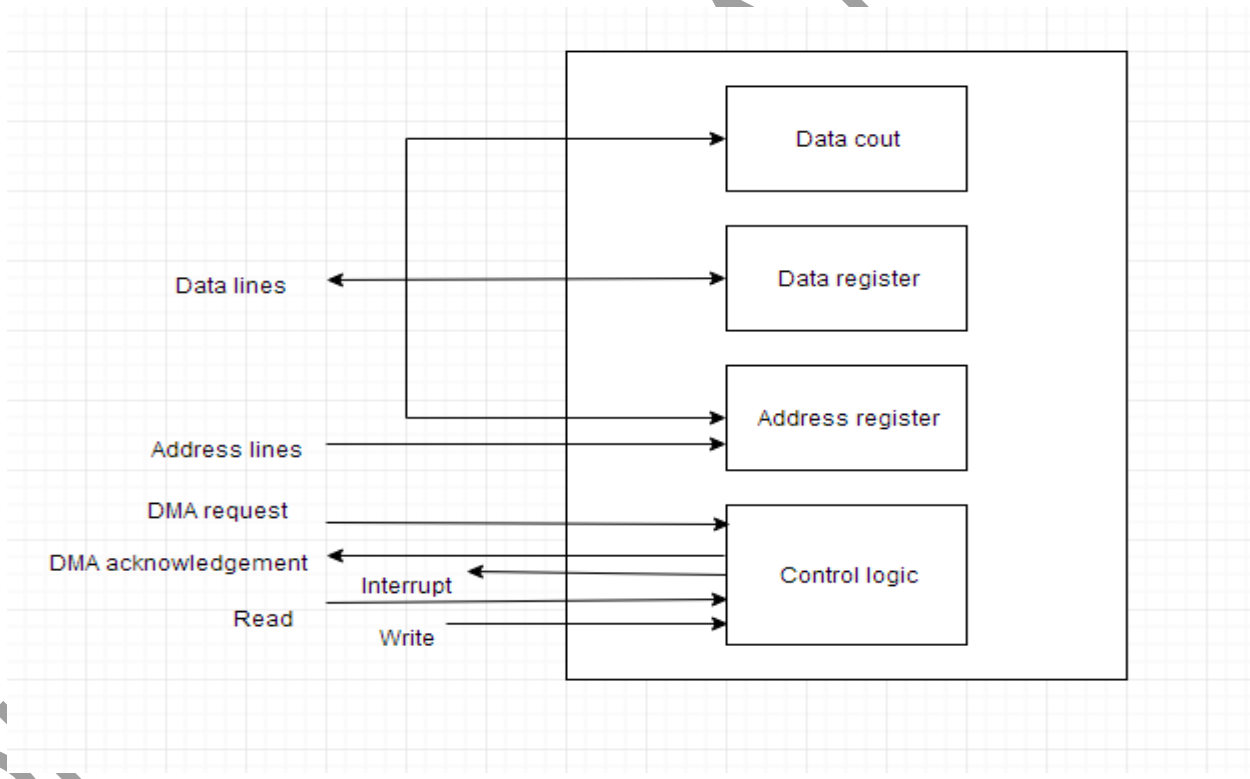
Direct Memory Access

DIWAKAR EDUCATION HUB

Removing the CPU from the path and letting the peripheral device manage the memory buses directly would improve the speed of transfer. This technique is known as **DMA**.

In this, the interface transfer data to and from the memory through memory bus. A DMA controller manages to transfer data between peripherals and memory unit.

Many hardware systems use DMA such as disk drive controllers, graphic cards, network cards and sound cards etc. It is also used for intra chip data transfer in multicore processors. In DMA, CPU would initiate the transfer, do other operations while the transfer is in progress and receive an interrupt from the DMA controller when the transfer has been completed.



Above figure shows block diagram of DMA

Input/Output Processor

An input-output processor (IOP) is a processor with direct memory access capability. In this, the computer system is divided into a memory unit and number of processors.

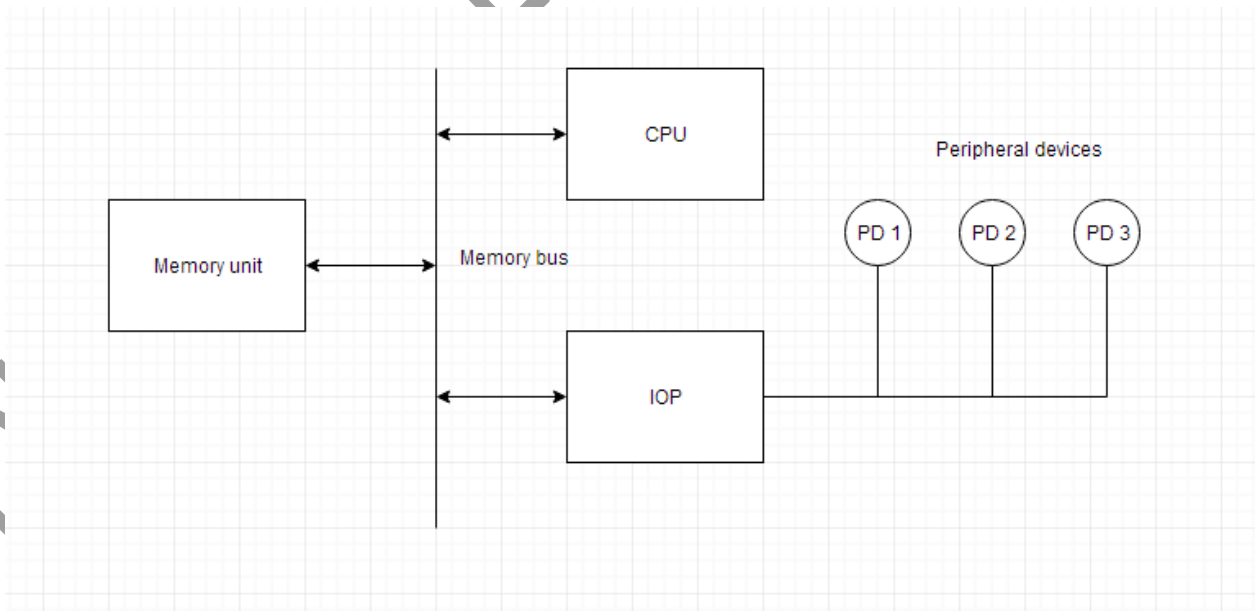
Each IOP controls and manage the input-output tasks. The IOP is similar to CPU except that it handles only the details of I/O processing. The IOP can fetch and execute its own instructions. These IOP instructions are designed to manage I/O transfers only.

Block Diagram Of I/O Processor

Below is a block diagram of a computer along with various I/O Processors. The memory unit occupies the central position and can communicate with each processor.

The CPU processes the data required for solving the computational tasks. The IOP provides a path for transfer of data between peripherals and memory. The CPU assigns the task of initiating the I/O program.

The IOP operates independent from CPU and transfer data between peripherals and memory.



EDUCATION HUB

The communication between the IOP and the devices is similar to the program control method of transfer. And the communication with the memory is similar to the direct memory access method.

In large scale computers, each processor is independent of other processors and any processor can initiate the operation.

The CPU can act as master and the IOP act as slave processor. The CPU assigns the task of initiating operations but it is the IOP, who executes the instructions, and not the CPU. CPU instructions provide operations to start an I/O transfer. The IOP asks for CPU through interrupt.

Instructions that are read from memory by an IOP are also called *commands* to distinguish them from instructions that are read by CPU. Commands are prepared by programmers and are stored in memory. Command words make the program for IOP. CPU informs the IOP where to find the commands in memory.

Interrupts

Data transfer between the CPU and the peripherals is initiated by the CPU. But the CPU cannot start the transfer unless the peripheral is ready to communicate with the CPU. When a device is ready to communicate with the CPU, it generates an interrupt signal. A number of input-output devices are attached to the computer and each device is able to generate an interrupt request.

The main job of the interrupt system is to identify the source of the interrupt. There is also a possibility that several devices will request simultaneously for CPU communication. Then, the interrupt system has to decide which device is to be serviced first.

Priority Interrupt

A priority interrupt is a system which decides the priority at which various devices, which generates the interrupt signal at the same time, will be serviced by the CPU. The system has authority to decide which conditions are allowed to interrupt the CPU, while some other interrupt is being serviced. Generally, devices

DIWAKAR EDUCATION HUB

with high speed transfer such as *magnetic disks* are given high priority and slow devices such as *keyboards* are given low priority.

When two or more devices interrupt the computer simultaneously, the computer services the device with the higher priority first.

Types of Interrupts:

Following are some different types of interrupts:

Hardware Interrupts

When the signal for the processor is from an external device or hardware then this interrupts is known as **hardware interrupt**.

an example: when we press any key on our keyboard to do some action, then this pressing of the key will generate an interrupt signal for the processor to perform certain action. Such an interrupt can be of two types:

- **Maskable Interrupt**

The hardware interrupts which can be delayed when a much high priority interrupt has occurred at the same time.

- **Non Maskable Interrupt**

The hardware interrupts which cannot be delayed and should be processed by the processor immediately.

Software Interrupts

The interrupt that is caused by any internal system of the computer system is known as a **software interrupt**. It can also be of two types:

- **Normal Interrupt**

The interrupts that are caused by software instructions are called **normal software interrupts**.

DIWAKAR EDUCATION HUB

- **Exception**

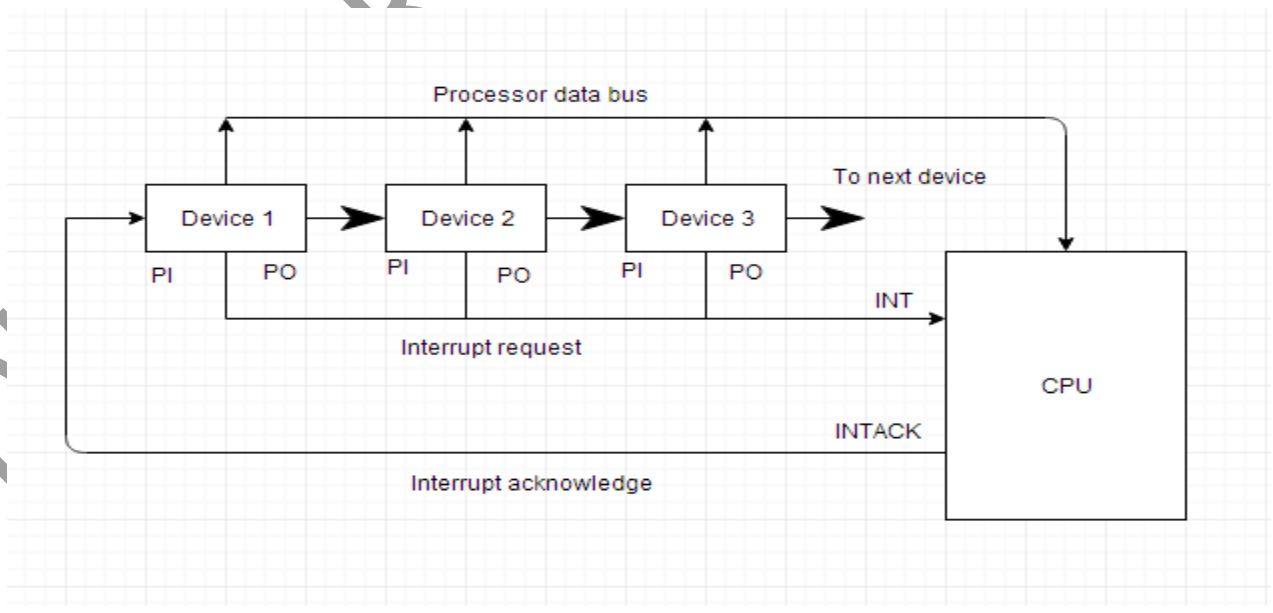
Unplanned interrupts which are produced during the execution of some program are called **exceptions**, such as division by zero.

Daisy Chaining Priority

This way of deciding the interrupt priority consists of serial connection of all the devices which generates an interrupt signal. The device with the highest priority is placed at the first position followed by lower priority devices and the device which has lowest priority among all is placed at the last in the chain.

In daisy chaining system all the devices are connected in a serial form. The interrupt line request is common to all devices. If any device has interrupt signal in low level state then interrupt line goes to low level state and enables the interrupt input in the CPU. When there is no interrupt the interrupt line stays in high level state. The CPU respond to the interrupt by enabling the interrupt acknowledge line. This signal is received by the device 1 at its PI input. The acknowledge signal passes to next device through PO output only if device 1 is not requesting an interrupt.

The following figure shows the block diagram for daisy chaining priority system.



Memory Organization in Computer Architecture

A memory unit is the collection of storage units or devices together. The memory unit stores the binary information in the form of bits. Generally, memory/storage is classified into 2 categories:

- **Volatile Memory**: This loses its data, when power is switched off.
- **Non-Volatile Memory**: This is a permanent storage and does not lose any data when power is switched off.

Memory Hierarchy

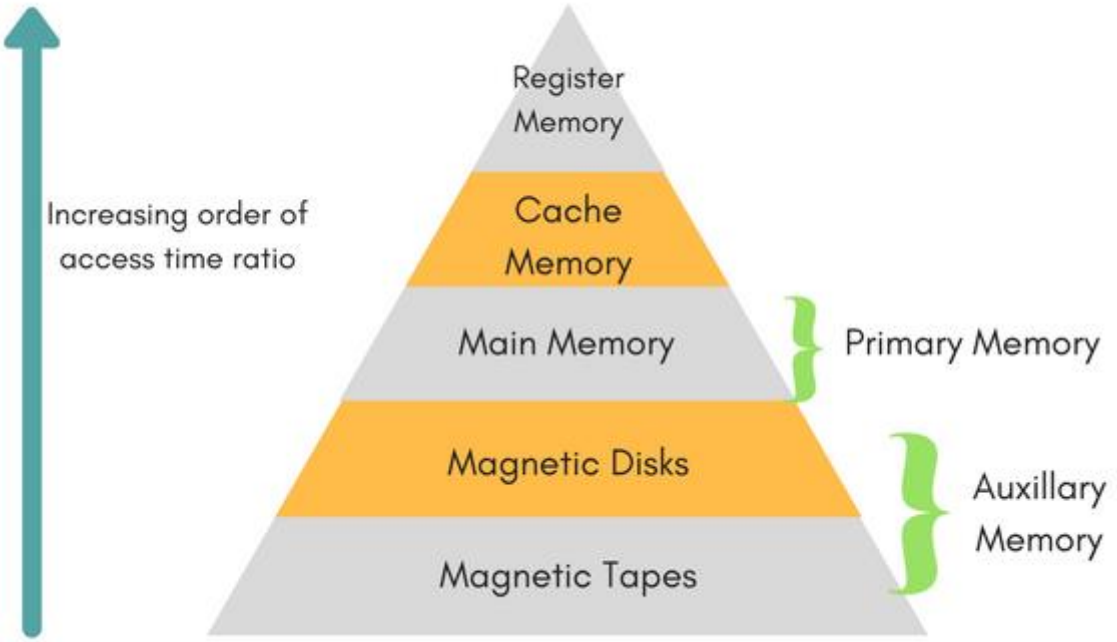
A memory unit is an essential component in any digital computer since it is needed for storing programs and data.

Typically, a memory unit can be classified into two categories:

1. The memory unit that establishes direct communication with the CPU is called **Main Memory**. The main memory is often referred to as RAM (Random Access Memory).
2. The memory units that provide backup storage are called **Auxiliary Memory**. For instance, magnetic disks and magnetic tapes are the most commonly used auxiliary memories.

Apart from the basic classifications of a memory unit, the memory hierarchy consists all of the storage devices available in a computer system ranging from the slow but high-capacity auxiliary memory to relatively faster main memory.

DIWAKAR EDUCATION HUB



The total memory capacity of a computer can be visualized by hierarchy of components. The memory hierarchy system consists of all storage devices contained in a computer system from the slow Auxiliary Memory to fast Main Memory and to smaller Cache memory.

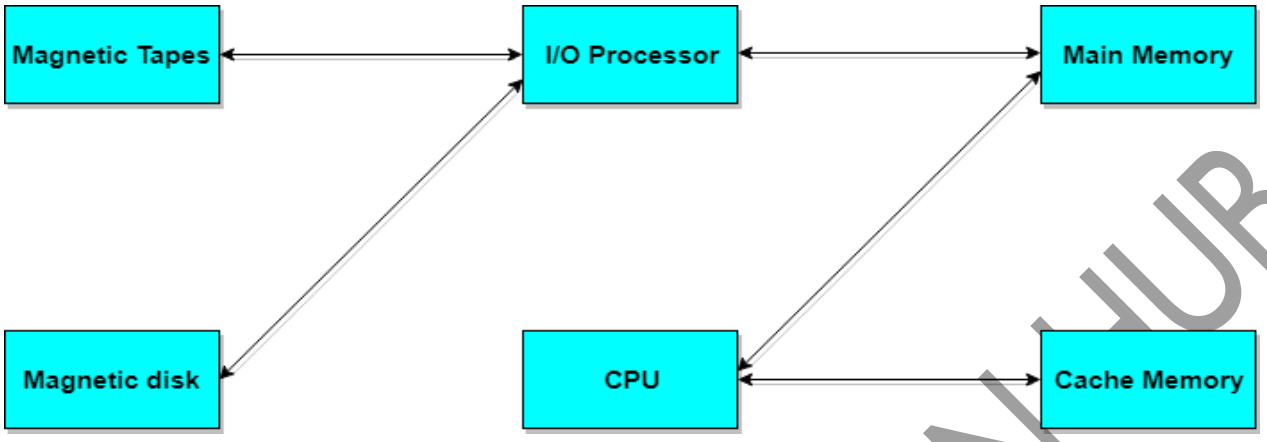
Auxillary memory access time is generally **1000 times** that of the main memory, hence it is at the bottom of the hierarchy.

The **main memory** occupies the central position because it is equipped to communicate directly with the CPU and with auxiliary memory devices through Input/output processor (I/O).

When the program not residing in main memory is needed by the CPU, they are brought in from auxiliary memory. Programs not currently needed in main memory are transferred into auxiliary memory to provide space in main memory for other programs that are currently in use.

The **cache memory** is used to store program data which is currently being executed in the CPU. Approximate access time ratio between cache memory and main memory is about **1 to 7~10**

DIWAKAR EDUCA



Memory Access Methods

Each memory type, is a collection of numerous memory locations. To access data from any memory, first it must be located and then the data is read from the memory location. Following are the methods to access information from memory locations:

1. **Random Access**: Main memories are random access memories, in which each memory location has a unique address. Using this unique address any memory location can be reached in the same amount of time in any order.
2. **Sequential Access**: This methods allows memory access in a sequence or in order.
3. **Direct Access**: In this mode, information is stored in tracks, with each track having a separate read/write head.

Main Memory

The main memory acts as the central storage unit in a computer system. It is a relatively large and fast memory which is used to store programs and data during the run time operations.

The primary technology used for the main memory is based on semiconductor integrated circuits.

DIWAKAR EDUCATION HUB

The memory unit that communicates directly within the CPU, Auxillary memory and Cache memory, is called main memory. It is the central storage unit of the computer system. It is a large and fast memory used to store data during computer operations. Main memory is made up of **RAM** and **ROM**, with RAM integrated circuit chips holing the major share.

- **RAM: Random Access Memory**
    - **DRAM**: Dynamic RAM, is made of capacitors and transistors, and must be refreshed every 10~100 ms. It is slower and cheaper than SRAM.
    - **SRAM**: Static RAM, has a six transistor circuit in each cell and retains data, until powered off.
    - **NVRAM**: Non-Volatile RAM, retains its data, even when turned off. Example: Flash memory.
- **ROM: Read Only Memory**
    - Read Only Memory, is non-volatile and is more like a permanent storage for information. It also stores the **bootstrap loader** program, to load and start the operating system when computer is turned on. **PROM**(Programmable ROM), **EPROM**(Erasable PROM) and **EEPROM**(Electrically Erasable PROM) are some commonly used ROMs.

Auxiliary Memory

Devices that provide backup storage are called auxiliary memory. **For example:** Magnetic disks and tapes are commonly used auxiliary devices. Other devices used as auxiliary memory are magnetic drums, magnetic bubble memory and optical disks.

It is not directly accessible to the CPU, and is accessed using the Input/Output channels.

DIWAKAR EDUCATION HUB

Auxiliary memory is known as the lowest-cost, highest-capacity and slowest-access storage in a computer system. It is where programs and data are kept for long-term storage or when not in immediate use. The most common examples of auxiliary memories are magnetic tapes and magnetic disks.

Magnetic Disks

A magnetic disk is a type of memory constructed using a circular plate of metal or plastic coated with magnetized materials. Usually, both sides of the disks are used to carry out read/write operations. However, several disks may be stacked on one spindle with read/write head available on each surface.

The following image shows the structural representation for a magnetic disk.

An Auxiliary memory is known as the lowest-cost, highest-capacity and slowest-access storage in a computer system. It is where programs and data are kept for long-term storage or when not in immediate use. The most common examples of auxiliary memories are magnetic tapes and magnetic disks.

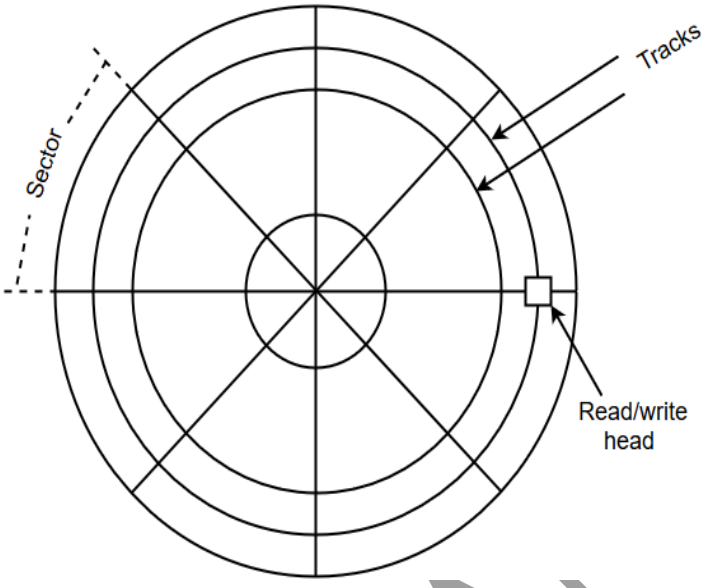
Magnetic Disks

A magnetic disk is a type of memory constructed using a circular plate of metal or plastic coated with magnetized materials. Usually, both sides of the disks are used to carry out read/write operations. However, several disks may be stacked on one spindle with read/write head available on each surface.

The following image shows the structural representation for a magnetic disk.

DIWAKAR EDUCATION HUB

Magnetic disks



- The memory bits are stored in the magnetized surface in spots along the concentric circles called tracks.
- The concentric circles (tracks) are commonly divided into sections called sectors.

Magnetic Tape

Magnetic tape is a storage medium that allows data archiving, collection, and backup for different kinds of data. The magnetic tape is constructed using a plastic strip coated with a magnetic recording medium.

The bits are recorded as magnetic spots on the tape along several tracks. Usually, seven or nine bits are recorded simultaneously to form a character together with a parity bit.

Magnetic tape units can be halted, started to move forward or in reverse, or can be rewound. However, they cannot be started or stopped fast enough between individual characters. For this reason, information is recorded in blocks referred to as records.

Cache Memory

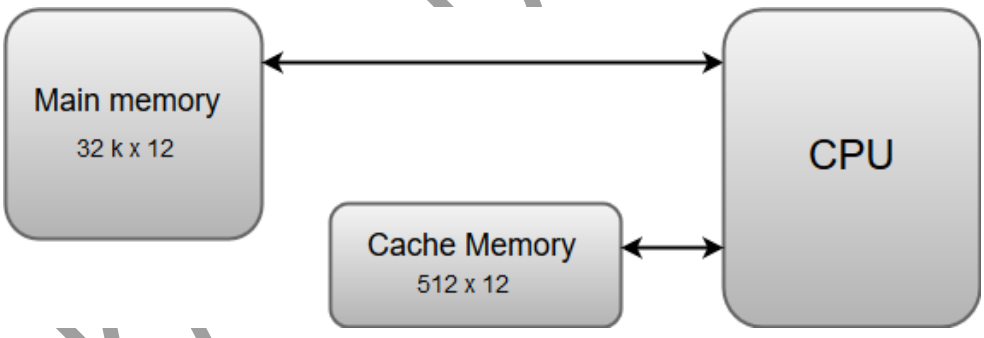
DIWAKAR EDUCATION HUB

The data or contents of the main memory that are used again and again by CPU, are stored in the cache memory so that we can easily access that data in shorter time.

Whenever the CPU needs to access memory, it first checks the cache memory. If the data is not found in cache memory then the CPU moves onto the main memory. It also transfers block of recent data into the cache and keeps on deleting the old data in cache to accomodate the new one.

The data or contents of the main memory that are used frequently by CPU are stored in the cache memory so that the processor can easily access that data in a shorter time. Whenever the CPU needs to access memory, it first checks the cache memory. If the data is not found in cache memory, then the CPU moves into the main memory.

Cache memory is placed between the CPU and the main memory. The block diagram for a cache memory can be represented as:



The cache is the fastest component in the memory hierarchy and approaches the speed of CPU components.

The basic operation of a cache memory is as follows:

- When the CPU needs to access memory, the cache is examined. If the word is found in the cache, it is read from the fast memory.
- If the word addressed by the CPU is not found in the cache, the main memory is accessed to read the word.

DIWAKAR EDUCATION HUB

- A block of words one just accessed is then transferred from main memory to cache memory. The block size may vary from one word (the one just accessed) to about 16 words adjacent to the one just accessed.
- The performance of the cache memory is frequently measured in terms of a quantity called **hit ratio**.
- When the CPU refers to memory and finds the word in cache, it is said to produce a **hit**.
- If the word is not found in the cache, it is in main memory and it counts as a **miss**.
- The ratio of the number of hits divided by the total CPU references to memory (hits plus misses) is the hit ratio.

Hit Ratio

The performance of cache memory is measured in terms of a quantity called **hit ratio**. When the CPU refers to memory and finds the word in cache it is said to produce a **hit**. If the word is not found in cache, it is in main memory then it counts as a **miss**.

The ratio of the number of hits to the total CPU references to memory is called hit ratio.

Hit Ratio = Hit/(Hit + Miss)

Associative Memory

It is also known as **content addressable memory (CAM)**. It is a memory chip in which each bit position can be compared. In this the content is compared in each bit cell which allows very fast table lookup. Since the entire chip can be compared, contents are randomly stored without considering addressing scheme. These chips have less storage capacity than regular memory chips.

An associative memory can be considered as a memory unit whose stored data can be identified for access by the content of the data itself rather than by an address or memory location.

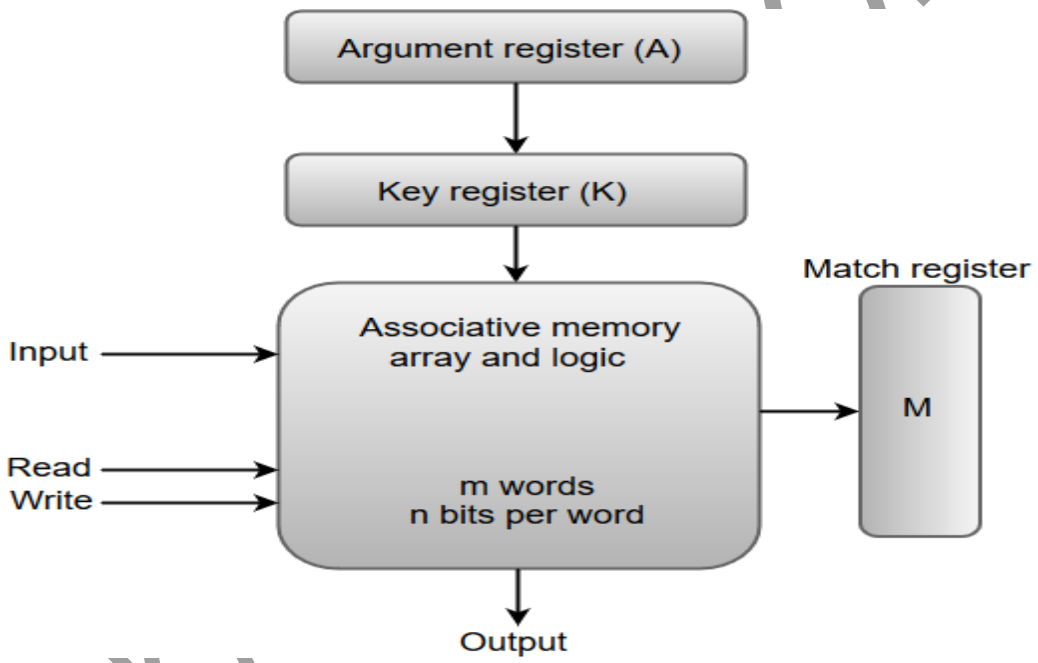
DIWAKAR EDUCATION HUB

Associative memory is often referred to as **Content Addressable Memory (CAM)**.

When a write operation is performed on associative memory, no address or memory location is given to the word. The memory itself is capable of finding an empty unused location to store the word.

On the other hand, when the word is to be read from an associative memory, the content of the word, or part of the word, is specified. The words which match the specified content are located by the memory and are marked for reading.

The following diagram shows the block representation of an Associative memory.



From the block diagram, we can say that an associative memory consists of a memory array and logic for 'm' words with 'n' bits per word.

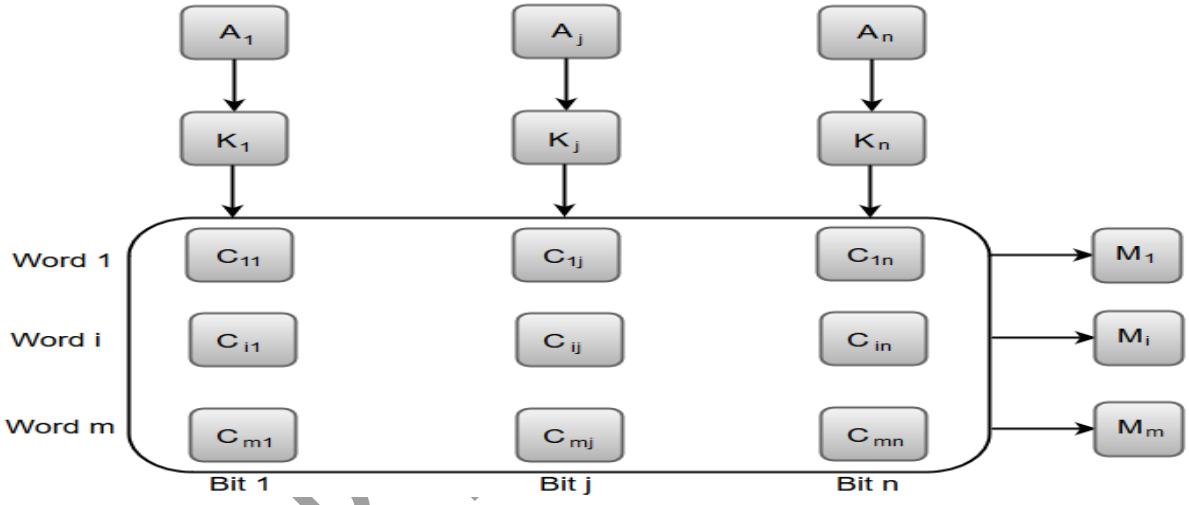
The functional registers like the argument register **A** and key register **K** each have **n** bits, one for each bit of a word. The match register **M** consists of **m** bits, one for each memory word.

The words which are kept in the memory are compared in parallel with the content of the argument register.

The key register (K) provides a mask for choosing a particular field or key in the argument word. If the key register contains a binary value of all 1's, then the entire argument is compared with each memory word. Otherwise, only those bits in the argument that have 1's in their corresponding position of the key register are compared. Thus, the key provides a mask for identifying a piece of information which specifies how the reference to memory is made.

The following diagram can represent the relation between the memory array and the external registers in an associative memory.

Associative memory of m word, n cells per word:



The cells present inside the memory array are marked by the letter C with two subscripts. The first subscript gives the word number and the second specifies the bit position in the word. For instance, the cell C_{ij} is the cell for bit **j** in word **i**.

A bit A_j in the argument register is compared with all the bits in column **j** of the array provided that $K_j = 1$. This process is done for all columns **j** = 1, 2, 3....., n.

If a match occurs between all the unmasked bits of the argument and the bits in word **i**, the corresponding bit M_i in the match register is set to 1. If one or more unmasked bits of the argument and the word do not match, M_i is cleared to 0.

Memory Mapping and Concept of Virtual Memory

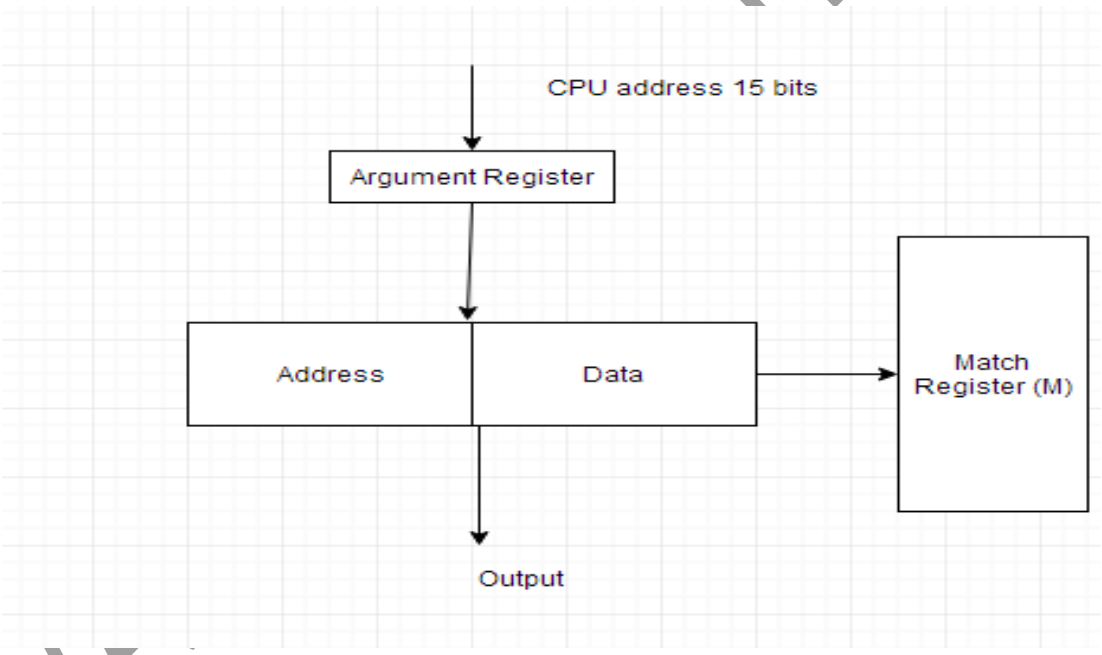
DIWAKAR

The transformation of data from main memory to cache memory is called mapping. There are 3 main types of mapping:

- Associative Mapping
- Direct Mapping
- Set Associative Mapping

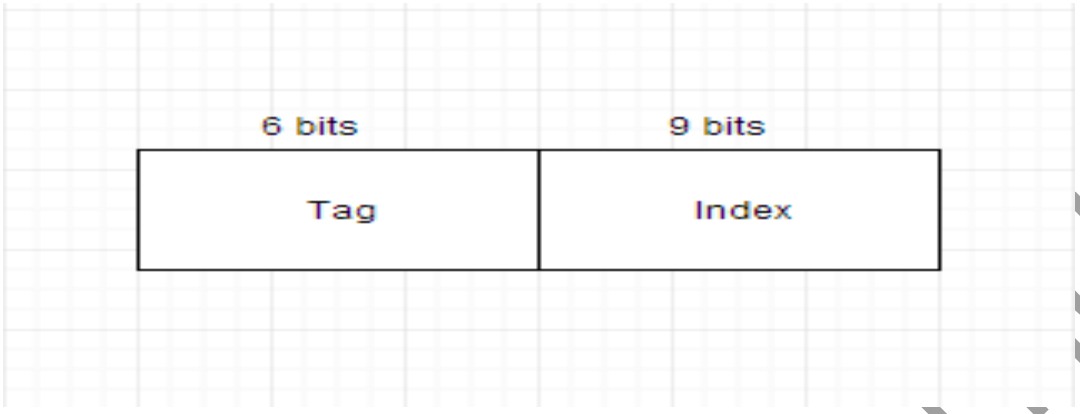
Associative Mapping

The associative memory stores both address and data. The address value of 15 bits is 5 digit octal numbers and data is of 12 bits word in 4 digit octal number. A CPU address of 15 bits is placed in **argument register** and the associative memory is searched for matching address.



Direct Mapping

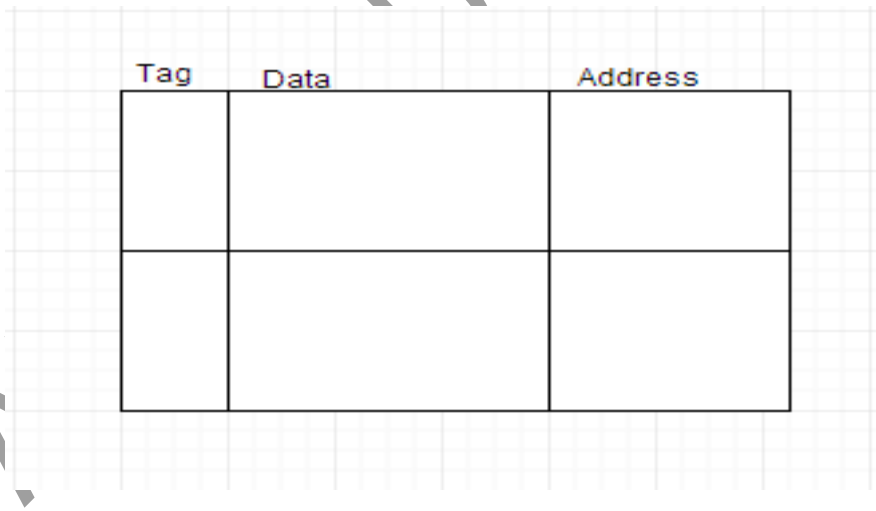
The CPU address of 15 bits is divided into 2 fields. In this the 9 least significant bits constitute the **index** field and the remaining 6 bits constitute the **tag** field. The number of bits in index field is equal to the number of address bits required to access cache memory.



Set Associative Mapping

The disadvantage of direct mapping is that two words with same index address can't reside in cache memory at the same time. This problem can be overcome by set associative mapping.

In this we can store two or more words of memory under the same index address. Each data word is stored together with its tag and this forms a set.



Replacement Algorithms

Data is continuously replaced with new data in the cache memory using replacement algorithms. Following are the 2 replacement algorithms used:

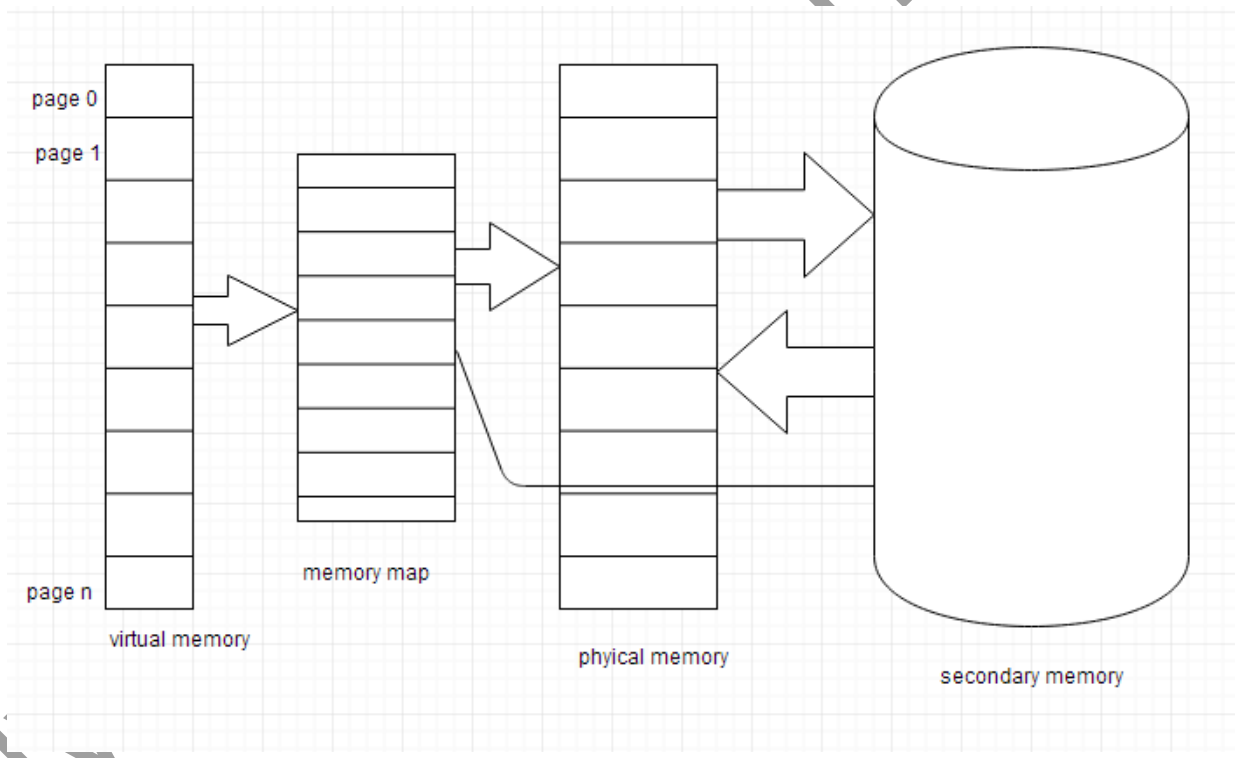
- FIFO - First in First out. Oldest item is replaced with the latest item.

- LRU - Least Recently Used. Item which is least recently used by CPU is removed.

Virtual Memory

Virtual memory is the separation of logical memory from physical memory. This separation provides large virtual memory for programmers when only small physical memory is available.

Virtual memory is used to give programmers the illusion that they have a very large memory even though the computer has a small main memory. It makes the task of programming easier because the programmer no longer needs to worry about the amount of physical memory available.



Memory Management Unit (MMU)

A computer's memory management unit (MMU) is the physical hardware that handles its virtual memory and caching operations. The MMU is usually located within the computer's central processing unit (CPU), but sometimes operates in a

separate integrated chip (IC). All data request inputs are sent to the MMU, which in turn determines whether the data needs to be retrieved from RAM or ROM storage.

A memory management unit is also known as a paged memory management unit.

The memory management unit performs three major functions:

- Hardware memory management
- Operating system (OS) memory management
- Application memory management

Hardware memory management deals with a system's RAM and cache memory, OS memory management regulates resources among objects and data structures, and application memory management allocates and optimizes memory among programs.

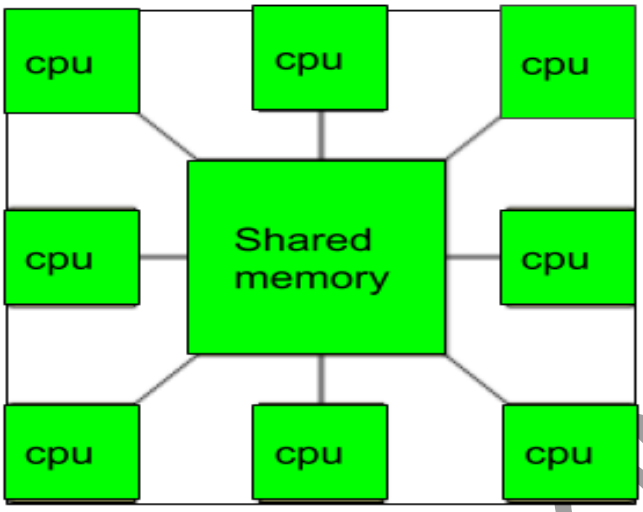
The MMU also includes a section of memory that holds a table that matches virtual addresses to physical addresses, called the translation lookaside buffer (TLB).

Multiprocessor

A Multiprocessor is a computer system with two or more central processing units (CPUs) share full access to a common RAM. The main objective of using a multiprocessor is to boost the system's execution speed, with other objectives being fault tolerance and application matching.

There are two types of multiprocessors, one is called shared memory multiprocessor and another is distributed memory multiprocessor. In shared memory multiprocessors, all the CPUs shares the common memory but in a distributed memory multiprocessor, every CPU has its own private memory.

DIWAKAR EDUCATION HUB



Applications of Multiprocessor –

1. As a uniprocessor, such as single instruction, single data stream (SISD).
2. As a multiprocessor, such as single instruction, multiple data stream (SIMD), which is usually used for vector processing.
3. Multiple series of instructions in a single perspective, such as multiple instruction, single data stream (MISD), which is used for describing hyper-threading or pipelined processors.
4. Inside a single system for executing multiple, individual series of instructions in multiple perspectives, such as multiple instruction, multiple data stream (MIMD).

Benefits of using a Multiprocessor –

- Enhanced performance.
- Multiple applications.
- Multi-tasking inside an application.
- High throughput and responsiveness.
- Hardware sharing among CPUs.

DIWAKAR EDUCATION HUB

Characteristics of multiprocessor

A multiprocessor system has two or more CPUs. It is an interconnection of two or more CPUs with memory and input-output equipment. The term “processor” in multiprocessor can mean either a central processing unit (CPU) or an input-output processor (IOP). However, a system with a single CPU and one or more IOPs is usually not included in the definition of a multiprocessor system unless the IOP has computational facilities comparable to a CPU. As it is most commonly defined, a multiprocessor system implies the existence of multiple CPUs, although usually there will be one or more IOPs as well. As mentioned earlier multiprocessors are classified as multiple instruction stream, multiple data stream (MIMD) systems.

There are some similarities between multiprocessor and multicomputer systems since both support concurrent operations. However, there exists an important distinction between a system with multiple computers and a system with multiple processors. Computers are interconnected with each other by means of communication lines to form a computer network. The network consists of several autonomous computers that may or may not communicate with each other. A multiprocessor system is controlled by one operating system that provides interaction between processors and all the components of the system cooperate in the solution of a problem.

Multiprocessing improves the reliability of the system so that a failure or error in one part has a limited effect on the rest of the system. If a fault cause one processor to fail, a second processor can be assigned to perform the functions of the disabled processor. The system as a whole can continue to function correctly with perhaps some loss in efficiency.

The benefit derived from a multiprocessor organization is an improved system performance. The system derives its high performance from the fact that computations can proceed in parallel in one of two ways.

1. Multiple independent jobs can be made to operate in parallel.

DIWAKAR EDUCATION HUB

2. A single job can be partitioned into multiple parallel tasks.

An overall function can be partitioned into a number of tasks that each processor can handle individually. System tasks may be allocated to special purpose processors whose design is optimized to perform certain types of processing efficiently. An example is a computer system where one processor performs the computations for an industrial process control while other monitor and control the various parameters, such as temperature and flow rate.

Multiprocessors are classified by the way their memory is organized. A multiprocessor system with common shared memory is classified as a shared-memory or tightly coupled multiprocessor. This does not preclude each processor from having its own local memory. In fact, most commercial tightly coupled multiprocessors provide a cache memory with each CPU. In addition, there is a global common memory that all CPUs can access. Information can therefore be shared among the CPUs by placing it in the common global memory.

An alternative model of microprocessor is the distributed-memory or loosely coupled system. Each processor element in a loosely coupled system has its own private local memory. The processors are tied together by a switching scheme designed to route information from one processor to another through a message-passing scheme. The processors relay program and data to other processors in packets. A packet consists of an address, the data content, and some error detection code. The packets are addressed to a specific processor or taken by the first available processor, depending on the communication system used. Loosely coupled systems are most efficient when the interaction between tasks is minimal, whereas tightly coupled systems can tolerate a higher degree of interaction between tasks.

Interconnection Structures

CPUs, TOPs connect' to input-output device, and a memory unit form a multiprocessor system, they may be partitioned into a number of separate modules. The interconnection between the components can have different

DIWAKAR EDUCATION HUB

physical configurations, depending on the number of transfer paths that are available between the processors and memory in a shared memory system or among the processing elements in a loosely coupled system. There are several physical forms available for establishing an interconnection network. Some of these schemes are presented in this section:

1. Time-shared common bus
2. Multiport memory
3. Crossbar switch
4. Multistage switching network
5. Hypercube system

Time Shared Common bus

A common-bus multiprocessor system consists of a number of processors connected through a common path to a memory unit. A time-shared common bus for five processors is shown in figure below. Only one processor can communicate with the memory or another processor at any given time. Transfer operations are conducted by the processor that is in control of the bus at the time. Any other processor wishing to initiate a transfer must first determine the availability status of the bus, and only after the bus becomes available can the processor address the destination unit to initiate the transfer. A command is issued to inform the destination unit what operation is to be performed. The receiving unit recognizes its address in the bus and responds to the control signals from the sender, after which the transfer is initiated. The system may exhibit transfer conflicts since one common bus is shared by all processors. These conflicts must be resolved by incorporating a bus controller that establishes priorities among the requesting units.

A single common-bus system is restricted to one transfer at a time. This means that when one processor is communicating with the memory, all other processors are either busy with internal operations or must be idle waiting for the bus. As a

DIWAKAR EDUCATION HUB

consequence, the total overall transfer rate within the system is limited by the speed of the single path. The processors in the system can be kept busy more often through the implementation of two or more independent buses to permit multiple simultaneous bus transfers. However, this increases the system cost and complexity.

Multiport Memory

A multiport memory system employs separate buses between each memory module and each CPU. This is shown in figure below for four CPUs and four memory modules (MMs). Each processor bus is connected to each memory module. A processor bus consists of the address, data, and control lines required to communicate with memory. The memory module is said to have four ports and each port accommodates one of the buses. The module must have internal control logic to determine which port will have access to memory at any given time. Memory access conflicts are resolved by assigning fixed priorities to each memory port. The priority for memory access associated with each processor may be established by the physical port position that its bus occupies in each module. Thus CPU1 will have priority over CPU2, CPU2 will have priority over CPU3, and CPU4 will have the lowest priority. The advantage of the multiport memory organization is the high transfer rate that can be achieved because of the multiple paths between processors and memory. The disadvantage is that it requires expensive memory control logic and a large number of cables and connectors. As a consequence, this interconnection structure is usually appropriate for systems with a small number of processors.

Crossbar Switch

The crossbar switch organization consists of a number of crosspoints that are placed at intersections between processor buses and memory module paths. Figure below shows a crossbar switch interconnection between four CPUs and four memory modules. The small square in each crosspoint is a switch that determines the path from a processor to a memory module. Each switch point has control logic to set up the transfer path between a processor and memory. It

DIWAKAR EDUCATION HUB

examines the address that is placed in the bus to determine whether its particular module is being addressed. It also resolves multiple requests for access to the same memory module on a predetermined priority basis.

Multistage Switching network

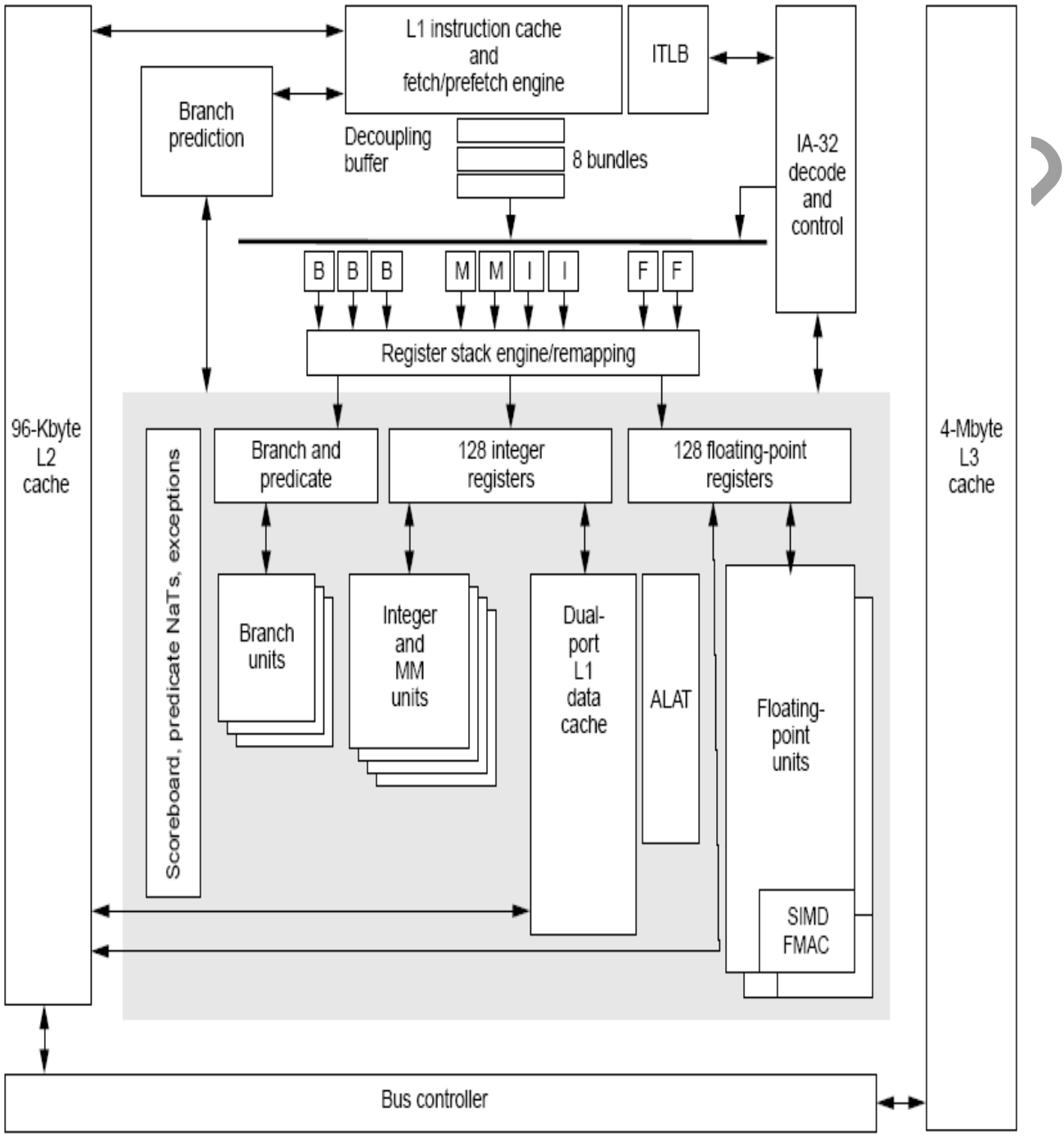
The basic component of a multistage network is a two-input, two-output interchange switch. There are control signals (not shown) associated with the switch that establish the interconnection between the input and output terminals. The switch has the capability of connecting input A to either of the outputs. Terminal B of the switch behaves in a similar fashion. The switch also has the capability to arbitrate between conflicting requests. If inputs A and B both request the same output terminal, only one of them will be connected; the other will be blocked.

Hypercube Interconnection

The hypercube or binary n-cube multiprocessor structure is a loosely coupled system composed of N = 2n processors interconnected in an n-dimensional binary cube. Each processor forms a node of the cube. Although it is customary to refer to each node as having a processor, in effect it contains not only a CPU but also local memory and I/O interface. Each processor has direct communication paths to n other neighbor processors. These paths correspond to the edges of the cube. There are 2^n distinct n-bit binary addresses that can be assigned to the processors. Each processor address differs from that of each of its n neighbors by exactly one bit position.

Interconnection Structures

DIWAKAR EDUCATION HUB



System bus

The processor uses a multidrop, shared system bus to provide four-way glueless multiprocessor system support. No additional bridges are needed for building up

to a four-way system. Systems with eight or more processors are designed through clusters of these nodes using high-speed interconnects. Note that multidrop buses are a cost-effective way to build high-performance four-way systems for commercial transaction processing and e-business workloads. These workloads often have highly shared writeable data and demand high throughput and low latency on transfers of modified data between caches of multiple processors. In a four-processor system, the transactionbased bus protocol allows up to 56 pending bus transactions (including 32 read transactions) on the bus at any given time. An advanced MESI coherence protocol helps in reducing bus invalidation transactions and in providing faster access to writeable data. The cache-to-cache transfer latency is further improved by an enhanced defer mechanism, which permits efficient out-of-order data transfers and out-of-order transaction completion on the bus. A deferred transaction on the bus can be completed without reusing the address bus. This reduces data return latency for deferred transactions and efficiently uses the address bus. This feature is critical for scalability beyond four-processor systems. The 64-bit system bus uses a source-synchronous data transfer to achieve 266-Mtransfers/ s, which enables a bandwidth of 2.1 Gbytes/s. The combination of these features makes the Itanium processor system a scalable building block for large multiprocessor systems.

Source Synchronous Mode

In source synchronous mode, the clock to data phase relationship at the input pins is maintained at the clock and data ports of the IOE input register. This mode is recommended for source synchronous data transfers. Data and clock signals at the IOE experience similar buffer delays as long as the same I/O standard is used.

Interprocessor Arbitration

Computer systems contain a number of buses at various levels to facilitate the transfer of information between components. The CPU contains a number of internal buses for transferring information between processor registers and ALU. A memory bus consists of lines for transferring data, address, and read/write information. An I/O bus is used to transfer information to and from input and

DIWAKAR EDUCATION HUB

output devices. A bus that connects major components in a multi-processor system, such as CPUs, lOPs, and memory, is called a system bus. The physical circuits of a system bus are contained in a number of identical printed circuit boards. Each board in the system belongs to a particular module. The board consists of circuits connected in parallel through connectors. Each pin of each circuit connector is connected by a wire to the corresponding pin of all other connectors in other boards. Thus any board can be plugged into a slot in the back-pane that forms the system bus.

The processors in a shared memory multiprocessor system request access to common memory or other common resources through the system bus. If n other processor is currently utilizing the bus, the requesting processor may be granted access immediately. However, the requesting processor must wait if another processor is currently utilizing the system bus. Furthermore, other processors may request the system bus at the same time. Arbitration must then be performed to resolve this multiple contention for the shared resources. The arbitration logic would be part of the system bus controller placed between the local bus and the system bus.

System Bus

A typical system bus consists of approximately 100 signal lines. These lines are divided into three functional groups: data, address, and control. In addition, there are power distribution lines that supply power to the components. For example, the IEEE standard 796 multibus system has 16 data lines, 24 address lines, 26 control lines, and 20 power lines, for a total of 86 lines.

The data lines provide a path for the transfer of data between processors and common memory. The number of data lines is usually a multiple of 8, with 16 and 32 being most common. The address lines are used to identify a memory address or any other source or destination, such as input or output ports. The number of address lines determines the maximum possible memory capacity in the system. For example, an address of 24 lines can access up to 2” (16 mega) words of memory. The data and address lines are terminated with three-state buffers. The

DIWAKAR EDUCATION HUB

address buffers are unidirectional from processor to memory. The data lines are bi- directional, allowing the transfer of data in either direction.

Data transfers over the system bus may be synchronous or asynchronous. In a synchronous bus, each data item is transferred during a time slice known in advance to both source and destination units; synchronization is achieved by driving both units from a common clock source. An alternative procedure is to have separate clocks of approximately the same frequency in each unit. Synchronization signals are transmitted periodically in order to keep all clocks asynchronous bus in the system in step with each other. In an asynchronous bus, each data item being transferred is accompanied by handshaking control signals to indicate when the data are transferred from the source and received by the destination.

The control lines provide signals for controlling the information transfer between units. Timing signals indicate the validity of data and address information. Command signals specify operations to be performed. Typical control lines include transfer signals such as memory read and write, acknowledge of a transfer, interrupt requests, bus control signals such as bus request and bus grant, and signals for arbitration procedures.

Serial Arbitration Procedure

Arbitration procedures service all processor requests on the basis of established priorities. A hardware bus priority resolving technique can be established by means of a serial or parallel connection of the units requesting control the system bus. The serial priority resolving technique is obtained from daisy-chain connection of bus arbitration circuits similar to the priority interrupt logic.

The processors connected to the system bus are assigned priority according to their position along the priority control line. The device closest to the priority line is assigned the highest priority. When multiple devices concurrently request the use of the bus, the device with the highest priority is granted access to it.

Parallel Arbitration Logic

DIWAKAR EDUCATION HUB

The parallel bus arbitration technique uses an external priority encoder and decoder as shown in figure below. Each bus arbiter in the parallel scheme has a bus request output line and a bus acknowledge input line. Each arbiter enables the request line when its processor is requesting access to the system bus. The processor takes control of the bus if its acknowledge input line is enabled. The bus busy line provides an orderly transfer of control, as in the daisy-chaining case.

Dynamic Arbitration Algorithms

The two bus arbitration procedures just described use a static priority algorithm since the priority of each device is fixed by the way it is connected to the bus. In contrast, a dynamic priority algorithm gives the system the capability for changing the priority of the devices while the system is in operation. We now discuss a few arbitration procedures that use dynamic priority algorithms.

The time slice algorithm allocates a fixed-length time slice of bus time that is offered sequentially to each processor, in round-robin fashion. The service given to each system component with this scheme is independent of its location along the bus. No preference is given to any particular device since each is allotted the same amount of time to communicate with the bus.

In a bus system that uses polling, the bus-grant signal is replaced by a set of lines called poll lines, which are connected to all units. These lines are used by the bus controller to define an address for each device connected to the bus. The bus controller sequences through the addresses in a prescribed manner. When a processor that requires access recognizes its address, it activates the bus busy-line and then accesses the bus. After a number of bus cycles, the polling process continues by choosing a different processor. The polling sequence is normally programmable, and as a result, the selection priority can be altered under program control.

The least recently used (LRU) algorithm gives the highest priority to the requesting device that has not used the bus for the longest interval. The priorities are adjusted after a number of bus cycles according to the LRU algorithm. With

DIWAKAR EDUCATION HUB

this procedure, no processor is favored over any other since the priorities are dynamically changed to give every device an opportunity to access the bus.

In the first-come, first-serve scheme, requests are served in the order received. To implement this algorithm, the bus controller establishes a queue arranged according to the time that the bus requests arrive. Each processor must wait for its turn to use the bus on a first-in, first-out (FIFO) basis. The rotating daisy-chain procedure is a dynamic extension of the daisy-chain algorithm. In this scheme there is no central bus controller, and the priority line is connected from the priority-out of the last device back to the priority-in of the first device in a closed loop. This is similar to the connections shown in figure for serial arbitration except that the PO output of arbiter 4 is connected to the PI input of arbiter 1. Whichever device has access to the bus serves as a bus controller for the following arbitration. Each arbiter priority for a given bus cycle is determined by its position along the bus priority line from the arbiter whose processor is currently controlling the bus. Once an arbiter releases the bus, it has the lowest priority.

Interprocessor Communication & Synchronization

The various processors in a multiprocessor system must be provided with a facility for communicating with each other. A communication path can be established through common input-output channels. In a shared memory multiprocessor system, the most common procedure is to set aside a portion of memory that is accessible to all processors. The primary use of the common memory is to act as a message center similar to a mailbox, where each processor can leave messages for other processors and pick up messages intended for it.

The sending processor structures a request, a message, or a procedure, and places it in the memory mailbox. Status bits residing in common memory are generally used to indicate the condition of the mailbox, whether it has meaningful information, and for which processor it is intended. The receiving processor can check the mailbox periodically to determine if there are valid messages for it. The response time of this procedure can be time consuming since a processor will

DIWAKAR EDUCATION HUB

recognize a request only when polling messages. A more efficient procedure is for the sending processor to alert the receiving processor directly by means of an interrupt signal. This can be accomplished through a software-initiated interprocessor interrupt by means of an instruction in the program of one processor which when executed produces an external interrupt condition in a second processor. This alerts the interrupted processor of the fact that a new message has been inserted by the interrupting processor. In addition to shared memory, a multiprocessor system may have other shared resources. For example, a magnetic disk storage unit connected to an IOP may be available to all CPUs. This provides a facility for sharing of system programs stored in the disk.

To prevent conflicting use of shared resources by several processors there must be a provision for assigning resources to processors. This task is given to the operating system. There are three organizations that have been used in the design of operating system for multiprocessors: master-slave configuration, separate operating system, and distributed operating system. In a master-slave mode, one processor, designated the master, always executes the operating system functions. The remaining processors, denoted as slaves, do not perform operating system functions. If a slave processor needs an operating system service, it must request it by interrupting the master and waiting until the current program can be interrupted. In the separate operating system organization, each processor can execute the operating system routines it needs. This organization is more suitable for loosely coupled systems where every processor may have its own copy of the entire operating system. In the distributed operating system organization, the operating system routines are distributed among the available processors. However, each particular operating system function is assigned to only one processor at a time. This type of organization is also referred to as a floating operating system since the routines float from one processor to another and the execution of the routines may be assigned to different processors at different times.

Inter Process Communication

DIWAKAR EDUCATION HUB

Inter Process Communication (IPC) is a mechanism that involves communication of one process with another process. This usually occurs only in one system.

Communication can be of two types –

- Between related processes initiating from only one process, such as parent and child processes.
- Between unrelated processes, or two or more different processes.

Following are some important terms that we need to know before proceeding further on this topic.

Pipes − Communication between two related processes. The mechanism is half duplex meaning the first process communicates with the second process. To achieve a full duplex i.e., for the second process to communicate with the first process another pipe is required.

FIFO − Communication between two unrelated processes. FIFO is a full duplex, meaning the first process can communicate with the second process and vice versa at the same time.

Message Queues − Communication between two or more processes with full duplex capacity. The processes will communicate with each other by posting a message and retrieving it out of the queue. Once retrieved, the message is no longer available in the queue.

Shared Memory − Communication between two or more processes is achieved through a shared piece of memory among all processes. The shared memory needs to be protected from each other by synchronizing access to all the processes.

Semaphores − Semaphores are meant for synchronizing access to multiple processes. When one process wants to access the memory (for reading or writing), it needs to be locked (or protected) and released when the access is removed. This needs to be repeated by all the processes to secure data.

DIWAKAR EDUCATION HUB

Signals − Signal is a mechanism to communication between multiple processes by way of signaling. This means a source process will send a signal (recognized by number) and the destination process will handle it accordingly.

Note − Almost all the programs in this tutorial are based on system calls under Linux Operating System (executed in Ubuntu).

Interprocessor Synchronization

The instruction set of a multiprocessor contains basic instructions that are used to implement communication and synchronization between cooperating processes. Communication refers to the exchange of data between different processes. For example, parameters passed to a procedure in a different processor constitute interprocessor communication. Synchronization refers to the special case where the data used to communicate between processors is control information. Synchronization is needed to enforce the correct sequence of processes and to ensure mutually exclusive access to shared writable data.

Multiprocessor systems usually include various mechanisms to deal with the synchronization of resources. Low-level primitives are implemented directly by the hardware. These primitives are the basic mechanisms that enforce mutual exclusion for more complex mechanisms implemented in software. A number of hardware mechanisms for mutual exclusion have been developed. One of the most popular methods is through the use of a binary semaphore.

Mutual Exclusion with a Semaphore

A binary variable called a semaphore is often used to indicate whether or not a processor is executing a critical section. A semaphore is a software-controlled flag that is stored in a memory location that all processors can access. When the semaphore is equal to 1, it means that a processor is executing a critical program, so that the shared memory is not available to other processors. When the semaphore is equal to 0, the shared memory is available to any requesting processor. Processors that share the same memory segment agree by convention not to use the memory segment unless the semaphore is equal to 0, indicating

DIWAKAR EDUCATION HUB

that memory is available. They also agree to set the semaphore to 1 when they are executing a critical section and to clear it to 0 when they are finished.

A semaphore can be initialized by means of a test and set instruction in conjunction with a hardware lock mechanism. A hardware lock is a processor-generated signal that serves to prevent other processors from using the system bus as long as the signal is active. The test-and-set instruction tests and sets a semaphore and activates the lock mechanism during the time that the instruction is being executed. This prevents other processors from changing the semaphore between the time that the processor is testing it and the time that it is setting it. Assume that the semaphore is a bit in the least significant position of a memory word whose address is symbolized by SEM. Let the mnemonic TSL designate the "test and set while locked" operation. The instruction

TSL SEM

will be executed in two memory cycles (the first to read and the second to write) without interference as follows:

R ← M[SEM] Test semaphore
M [SEM] ← 1 Set semaphore

The semaphore is tested by transferring its value to a processor register R and then it is set to 1. The value in R determines what to do next. If the processor finds that R = 1, it knows that the semaphore was originally set (the fact that it is set again does not change the semaphore value). That means another processor is executing a critical section, so the processor that checked the semaphore does not access the shared memory. R = 0 means that the common memory (or the shared resource that the semaphore represents) is available. The semaphore is set to 1 to prevent other processors from accessing memory. The processor can now execute the critical section. The last instruction in the program must clear location SEM to zero to release the share resource to other processors.

Cache Coherence

DIWAKAR EDUCATION HUB

We know that the primary advantage of cache is its ability to reduce the average access time in uniprocessor systems. When the processor finds a word in cache during a read operation, the main memory is not involved in the transfer. If the operation is to write, there are two commonly used procedures to update memory. In the write-through policy, both cache and main memory are updated with every write operation. In the write-back policy, only the cache is updated and the location is marked so that it can be copied later into main memory. In a shared memory multiprocessor system, all the processors share a common memory. In addition, each processor may have a local memory, part or all of which may be a cache. The compelling reason for having separate caches for each processor is to reduce the average access time in each processor .The same information may reside in a number of copies in some caches and main memory. To ensure the ability of the system to execute memory operations correctly, the multiple copies must be kept identical. This requirement imposes a cache coherence problem. A memory scheme is coherent if the value returned on a load instruction is always the value given by the latest store instruction with the same address. Without a proper solution to the cache coherence problem, caching cannot be used in bus-oriented multiprocessor with two or more processors.

Conditions for Incoherence

Cache-coherence problems occur in multiprocessors with private caches because of the need to share writable data. Read-only data can safely be replicated without cache coherence enforcement mechanism to illustrate the problem, consider the three-processor configuration with private caches

Solution to the Cache Coherence System

There are various schemes to solve the cache coherence problem in shared memory multiprocessors.

A simple scheme is to disallow private caches for each processor and have a shared cache memory associated with main memory. Every data access is made to the shared cache. This method violates the principle of closeness of CPU to

DIWAKAR EDUCATION HUB

cache and increases the average memory access time. In effect, this scheme solves the problem by avoiding it.

For performance considerations it is desirable to attach a private cache to each processor. One scheme that has been used allows only non-shared and read-only to be stored in caches. Such items are called cacheable. Shared writable data are non-cachable. The compiler must tag data as either cachable or non-cachable, and the system hardware makes sure that only cachable data are stored in caches. The non-cachable data remain in main memory. This method restricts the type of data stored in caches and introduces an extra software overhead that may degrade performance.

A scheme that allows writable data to exist in at least one cache is a method that employs a centralized global table in its compiler. The status of memory blocks is stored in the central global table. Each block is identified as read-only (RO) or read-write (RW). All caches can have copies of blocks identified as RO.

Only one cache can have a copy of an RW block. Thus if the data are updated in the cache with an RW block, the other caches are not affected because they do not have a copy of this block. The cache coherence problem can be solved by means of a combination of software and hardware or by means of hardware-only schemes. The two methods mentioned previously use software-based procedures that require the ability to tag information in order to disable caching of shared writable data. Hardware-only solutions are handled by the hardware automatically and have the advantage of higher speed and program transparency. In the hardware solution, the cache controller is speedily designed to allow it to monitor all bus requests from CPUs and IOPs. All caches attached to the bus constantly monitor the network for possible write operations. Depending on the method used, they must then either update or invalidate their own cache copies when a match is detected. The bus controller that monitors this action is referred to snoopy cache as a snoopy cache controller. This is basically a hardware unit designed to maintain a bus-watching mechanism over all the caches attached to the bus.

DIWAKAR EDUCATION HUB

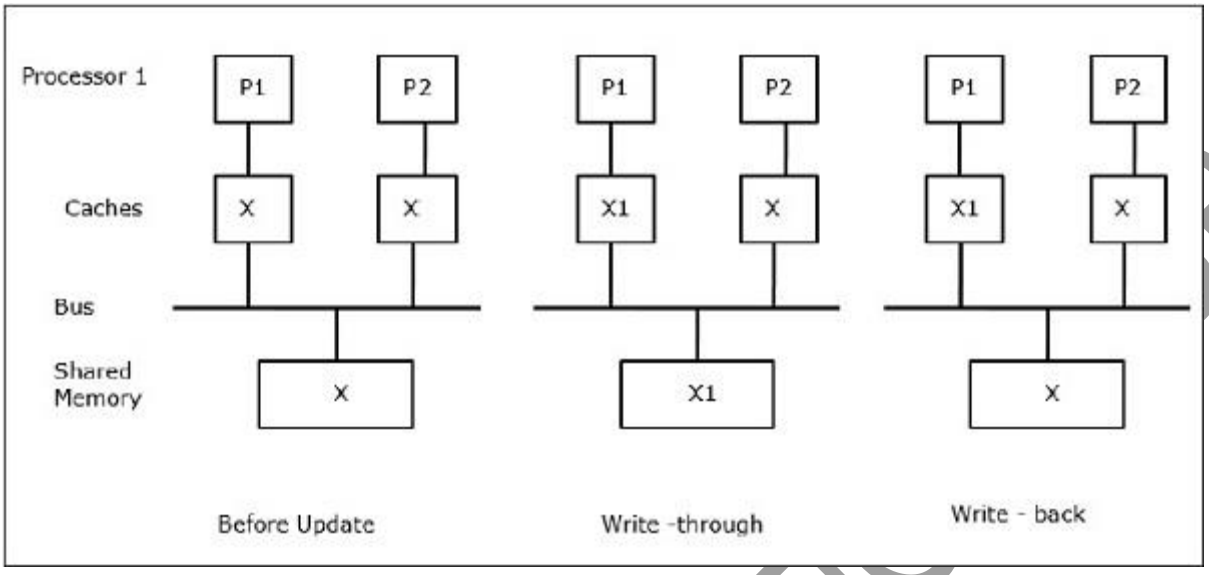
Various schemes have been proposed to solve the cache coherence problem by means of snoopy cache protocol. The simplest method is to adopt a write-through policy and use the following procedure. All the snoopy controllers watch the bus for memory store operations. When a word in a cache is updated by writing into it, the corresponding location in main memory is also updated. The local snoopy controllers in all other caches check their memory to determine if they have a copy of the word that has been overwritten. If a copy exists in a remote cache, that location is marked invalid. Because all caches snoop on all bus writes, whenever a word is written, the net effect is to update it in the original cache and main memory and remove it from all other caches. If at some future time a processor accesses the invalid item from its cache, the response is equivalent to a cache miss, and the updated item is transferred from main memory. In this way, inconsistent versions are prevented

The Cache Coherence Problem

In a multiprocessor system, data inconsistency may occur among adjacent levels or within the same level of the memory hierarchy. For example, the cache and the main memory may have inconsistent copies of the same object.

As multiple processors operate in parallel, and independently multiple caches may possess different copies of the same memory block, this creates **cache coherence problem**. **Cache coherence schemes** help to avoid this problem by maintaining a uniform state for each cached block of data.

DIWAKAR EDUCATION HUB



Let X be an element of shared data which has been referenced by two processors, P1 and P2. In the beginning, three copies of X are consistent. If the processor P1 writes a new data X1 into the cache, by using **write-through policy**, the same copy will be written immediately into the shared memory. In this case, inconsistency occurs between cache memory and the main memory. When a **write-back policy** is used, the main memory will be updated when the modified data in the cache is replaced or invalidated.

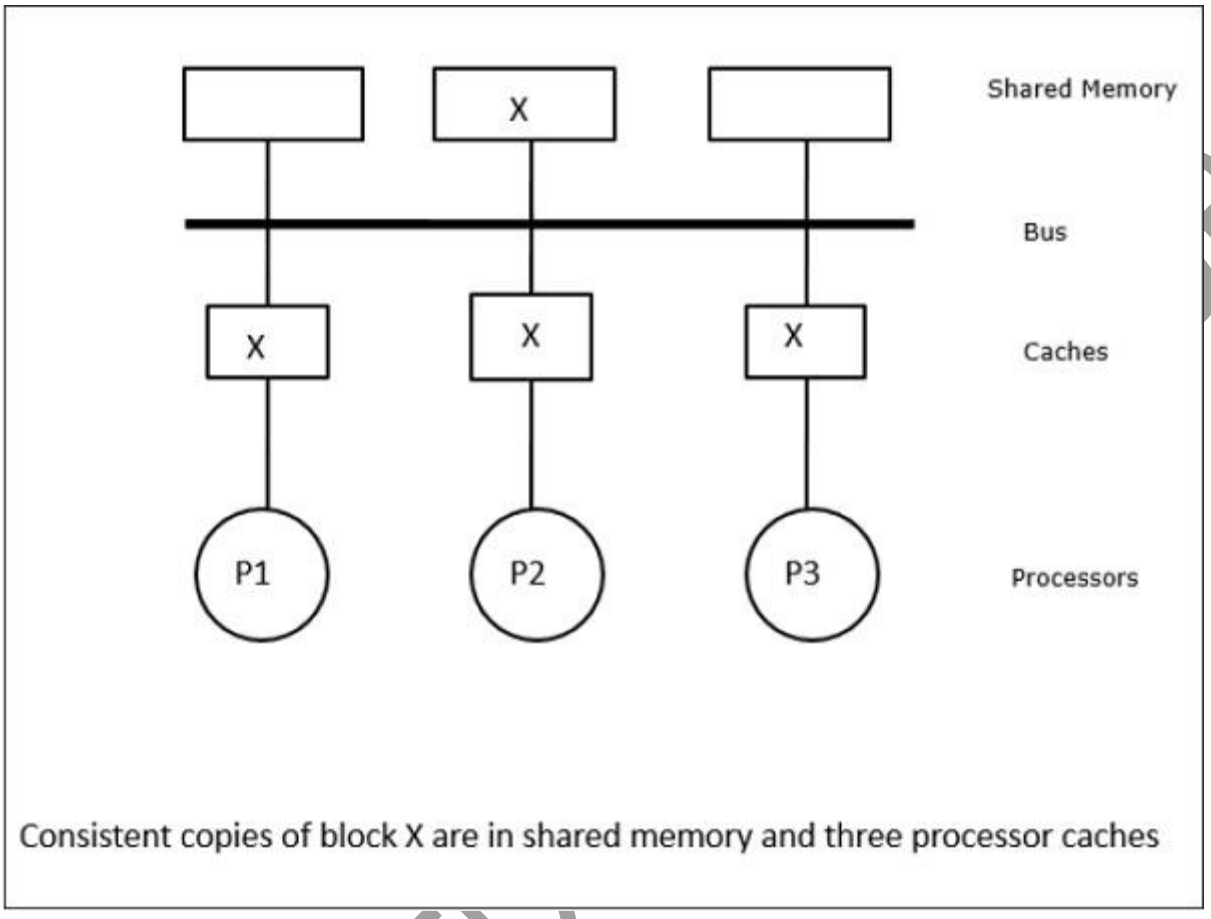
In general, there are three sources of inconsistency problem –

- Sharing of writable data
- Process migration
- I/O activity

Snoopy Bus Protocols

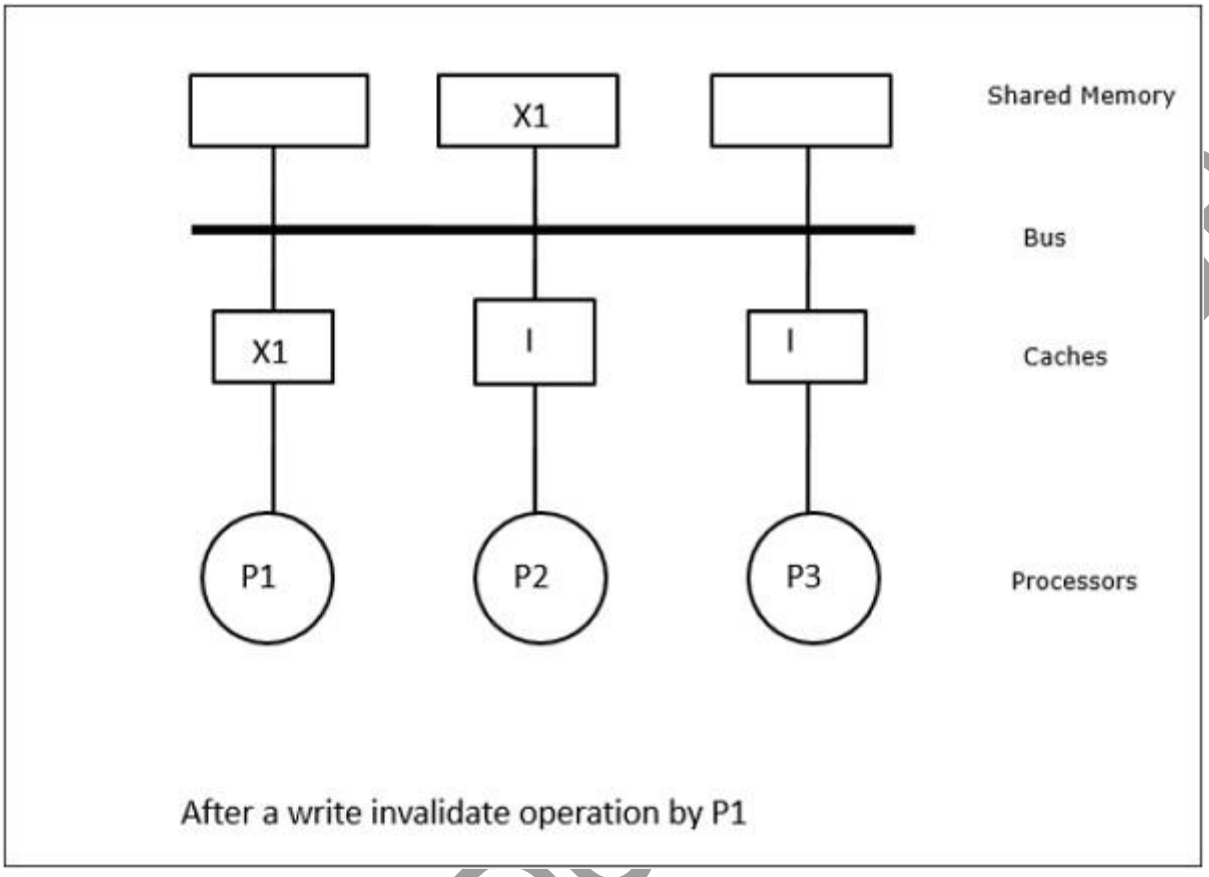
Snoopy protocols achieve data consistency between the cache memory and the shared memory through a bus-based memory system. **Write-invalidate** and **write-update** policies are used for maintaining cache consistency.

DIWAKAR EDUCATION



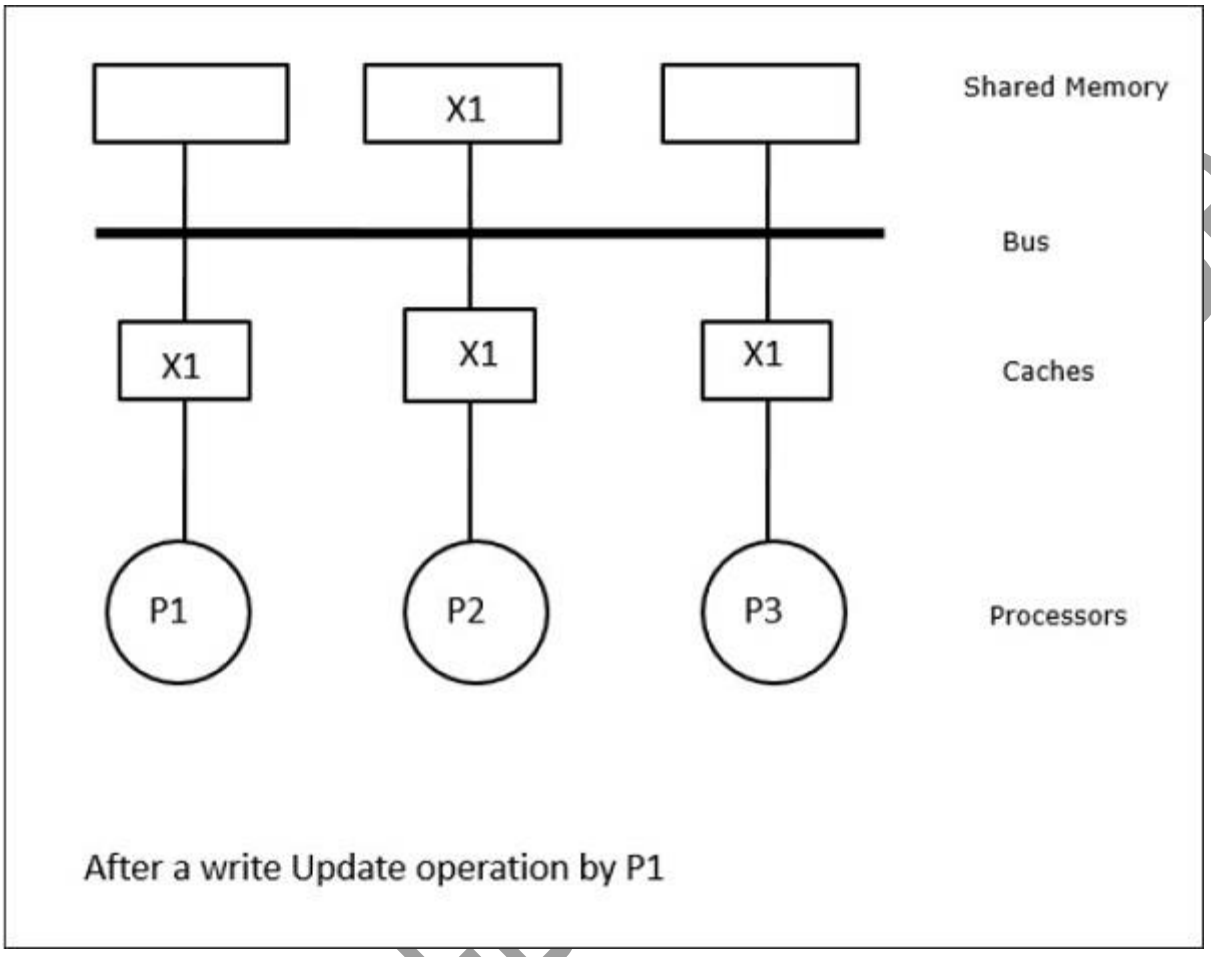
Consistent copies of block X are in shared memory and three processor caches

DIWAKAR ED



In this case, we have three processors P1, P2, and P3 having a consistent copy of data element 'X' in their local cache memory and in the shared memory (Figure-a). Processor P1 writes X1 in its cache memory using **write-invalidate protocol**. So, all other copies are invalidated via the bus. It is denoted by 'I' (Figure-b). Invalidated blocks are also known as **dirty**, i.e. they should not be used. The **write-update protocol** updates all the cache copies via the bus. By using **write back cache**, the memory copy is also updated (Figure-c).

DIWAKAR EDUCATION HUB



Cache Events and Actions

Following events and actions occur on the execution of memory-access and invalidation commands –

- **Read-miss** – When a processor wants to read a block and it is not in the cache, a read-miss occurs. This initiates a **bus-read** operation. If no dirty copy exists, then the main memory that has a consistent copy, supplies a copy to the requesting cache memory. If a dirty copy exists in a remote cache memory, that cache will restrain the main memory and send a copy to the requesting cache memory. In both the cases, the cache copy will enter the valid state after a read miss.

DIWAKAR EDUCATION HUB

- **Write-hit** − If the copy is in dirty or **reserved** state, write is done locally and the new state is dirty. If the new state is valid, write-invalidate command is broadcasted to all the caches, invalidating their copies. When the shared memory is written through, the resulting state is reserved after this first write.

- **Write-miss** − If a processor fails to write in the local cache memory, the copy must come either from the main memory or from a remote cache memory with a dirty block. This is done by sending a **read-invalidate** command, which will invalidate all cache copies. Then the local copy is updated with dirty state.

- **Read-hit** − Read-hit is always performed in local cache memory without causing a transition of state or using the snoopy bus for invalidation.

- **Block replacement** − When a copy is dirty, it is to be written back to the main memory by block replacement method. However, when the copy is either in valid or reserved or invalid state, no replacement will take place.

Directory-Based Protocols

By using a multistage network for building a large multiprocessor with hundreds of processors, the snoopy cache protocols need to be modified to suit the network capabilities. Broadcasting being very expensive to perform in a multistage network, the consistency commands is sent only to those caches that keep a copy of the block. This is the reason for development of directory-based protocols for network-connected multiprocessors.

In a directory-based protocols system, data to be shared are placed in a common directory that maintains the coherence among the caches. Here, the directory acts as a filter where the processors ask permission to load an entry from the primary memory to its cache memory. If an entry is changed the directory either updates it or invalidates the other caches with that entry.

Hardware Synchronization Mechanisms

DIWAKAR EDUCATION HUB

Synchronization is a special form of communication where instead of data control, information is exchanged between communicating processes residing in the same or different processors.

Multiprocessor systems use hardware mechanisms to implement low-level synchronization operations. Most multiprocessors have hardware mechanisms to impose atomic operations such as memory read, write or read-modify-write operations to implement some synchronization primitives. Other than atomic memory operations, some inter-processor interrupts are also used for synchronization purposes.

Cache Coherency in Shared Memory Machines

Maintaining cache coherency is a problem in multiprocessor system when the processors contain local cache memory. Data inconsistency between different caches easily occurs in this system.

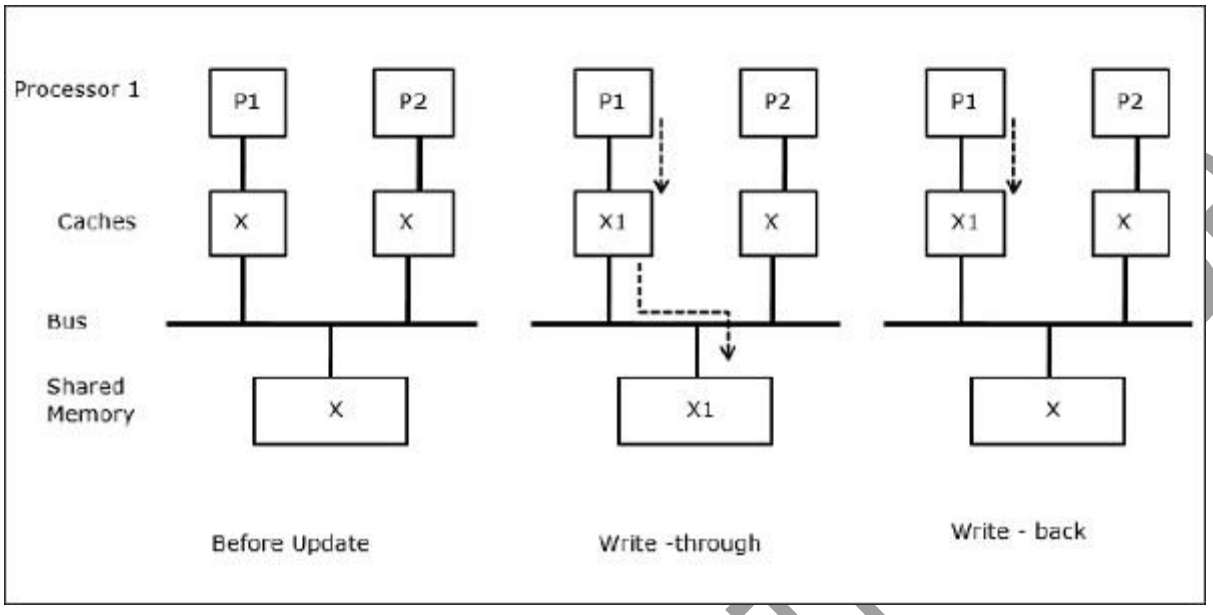
The major concern areas are –

- Sharing of writable data
- Process migration
- I/O activity

Sharing of writable data

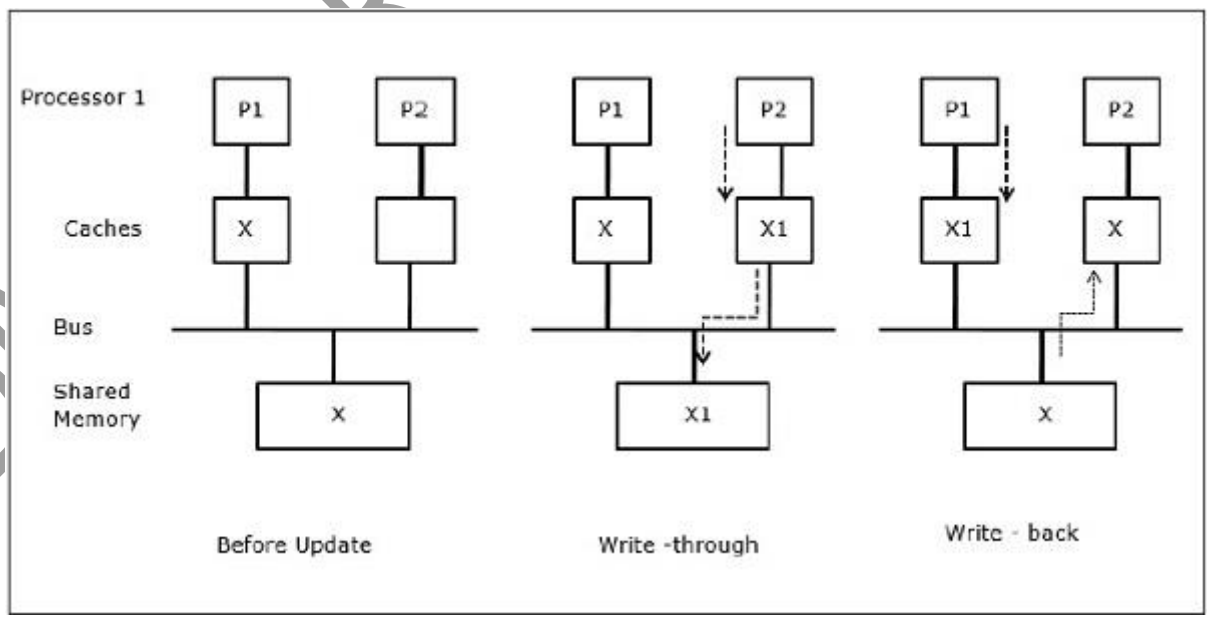
When two processors (P1 and P2) have same data element (X) in their local caches and one process (P1) writes to the data element (X), as the caches are write-through local cache of P1, the main memory is also updated. Now when P2 tries to read data element (X), it does not find X because the data element in the cache of P2 has become outdated.

DIWAKAR EDUCATION HUB



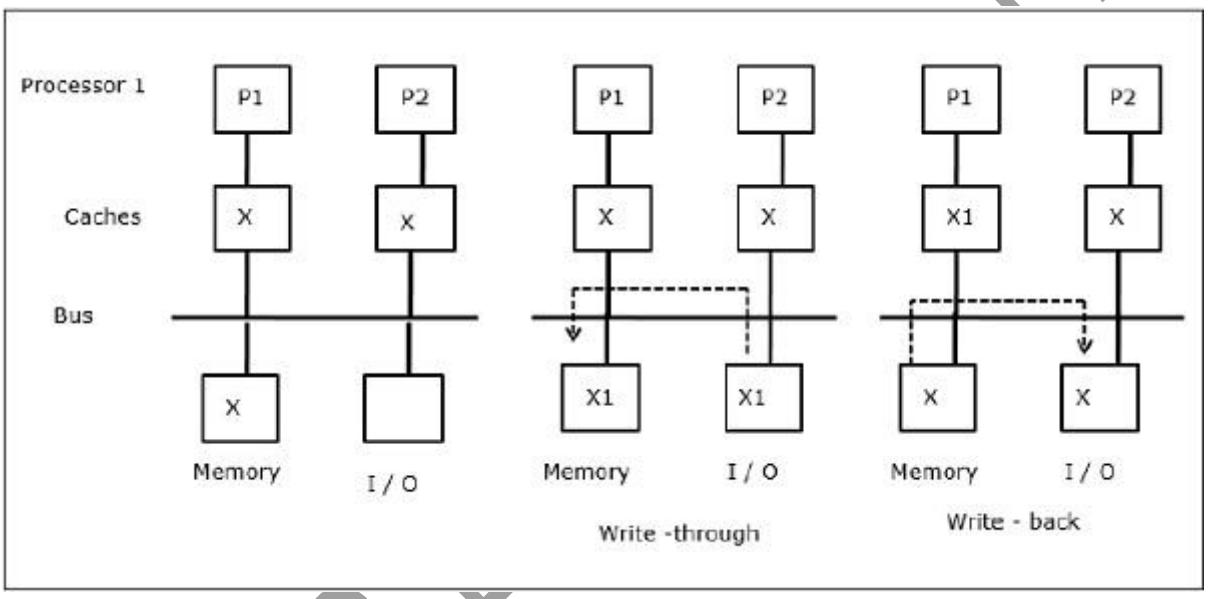
Process migration

In the first stage, cache of P1 has data element X, whereas P2 does not have anything. A process on P2 first writes on X and then migrates to P1. Now, the process starts reading data element X, but as the processor P1 has outdated data the process cannot read it. So, a process on P1 writes to the data element X and then migrates to P2. After migration, a process on P2 starts reading the data element X but it finds an outdated version of X in the main memory.



I/O activity

As illustrated in the figure, an I/O device is added to the bus in a two-processor multiprocessor architecture. In the beginning, both the caches contain the data element X. When the I/O device receives a new element X, it stores the new element directly in the main memory. Now, when either P1 or P2 (assume P1) tries to read element X it gets an outdated copy. So, P1 writes to element X. Now, if I/O device tries to transmit X it gets an outdated copy.



Uniform Memory Access (UMA)

Uniform Memory Access (UMA) architecture means the shared memory is the same for all processors in the system. Popular classes of UMA machines, which are commonly used for (file-) servers, are the so-called Symmetric Multiprocessors (SMPs). In an SMP, all system resources like memory, disks, other I/O devices, etc. are accessible by the processors in a uniform manner.

Non-Uniform Memory Access (NUMA)

In NUMA architecture, there are multiple SMP clusters having an internal indirect/shared network, which are connected in scalable message-passing

network. So, NUMA architecture is logically shared physically distributed memory architecture.

In a NUMA machine, the cache-controller of a processor determines whether a memory reference is local to the SMP's memory or it is remote. To reduce the number of remote memory accesses, NUMA architectures usually apply caching processors that can cache the remote data. But when caches are involved, cache coherency needs to be maintained. So these systems are also known as CC-NUMA (Cache Coherent NUMA).

Cache Only Memory Architecture (COMA)

COMA machines are similar to NUMA machines, with the only difference that the main memories of COMA machines act as direct-mapped or set-associative caches. The data blocks are hashed to a location in the DRAM cache according to their addresses. Data that is fetched remotely is actually stored in the local main memory. Moreover, data blocks do not have a fixed home location, they can freely move throughout the system.

COMA architectures mostly have a hierarchical message-passing network. A switch in such a tree contains a directory with data elements as its sub-tree. Since data has no home location, it must be explicitly searched for. This means that a remote access requires a traversal along the switches in the tree to search their directories for the required data. So, if a switch in the network receives multiple requests from its subtree for the same data, it combines them into a single request which is sent to the parent of the switch. When the requested data returns, the switch sends multiple copies of it down its subtree.

COMA versus CC-NUMA

Following are the differences between COMA and CC-NUMA.

- COMA tends to be more flexible than CC-NUMA because COMA transparently supports the migration and replication of data without the need of the OS.

DIWAKAR EDUCATION HUB

- COMA machines are expensive and complex to build because they need non-standard memory management hardware and the coherency protocol is harder to implement.
- Remote accesses in COMA are often slower than those in CC-NUMA since the tree network needs to be traversed to find the data.

Multicore processor

A **multicore processor** is a single computing component comprised of two or more CPUs that read and execute the actual program instructions. The individual cores can execute multiple instructions in parallel, increasing the performance of software which is written to take advantage of the unique architecture.

The first multicore processors were produced by Intel and AMD in the early 2000s. Today, processors are created with two cores ("dual core"), four cores ("quad core"), six cores ("hexa core"), and eight cores ("octo core"). Processors are made with as many as 100 physical cores, as well as 1000 effective independent cores by using FPGAs (Field Programmable Gate Arrays).

A **multicore processor** is a single integrated circuit (a.k.a., chip multiprocessor or CMP) that contains multiple core processing units, more commonly known as *cores*. There are many different multicore processor architectures, which vary in terms of

- **Number of cores**. Different multicore processors often have different numbers of cores. For example, a quad-core processor has four cores. The number of cores is usually a power of two.
- **Number of core types**.
    - **Homogeneous (symmetric) cores.** All of the cores in a homogeneous multicore processor are of the same type; typically the core processing units are general-purpose central processing units that run a single multicore operating system.

DIWAKAR EDUCATION HUB

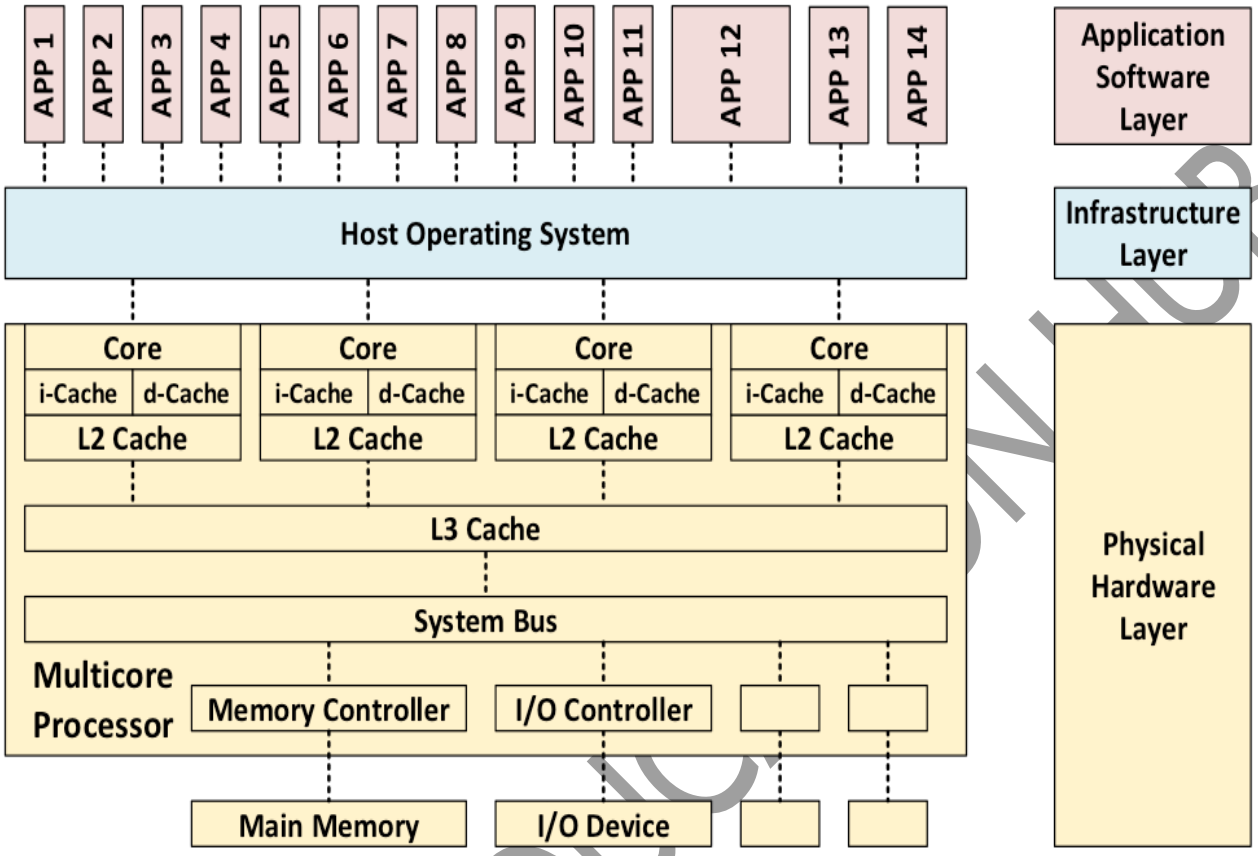
- **Heterogeneous (asymmetric) cores**. Heterogeneous multicore processors have a mix of core types that often run different operating systems and include graphics processing units.

- **Number and level of caches**. Multicore processors vary in terms of their instruction and data caches, which are relatively small and fast pools of local memory.
- **How cores are interconnected.** Multicore processors also vary in terms of their bus architectures.
- **Isolation**. The amount, typically minimal, of in-chip support for the spatial and temporal isolation of cores:
  - **Physical isolation** ensures that different cores cannot access the same physical hardware (e.g., memory locations such as caches and RAM).
  - **Temporal isolation** ensures that the execution of software on one *core* does not impact the temporal behavior of software running on another *core*.

Homogeneous Multicore Processor

The following figure notionally shows the architecture of a system in which 14 software applications are allocated by a single host operating system to the cores in a homogeneous quad-core processor. In this architecture, there are three levels of cache, which are progressively larger but slower: L1 (consisting of an instruction cache and a data cache), L2, and L3. Note that the L1 and L2 caches are local to a single core, whereas L3 is shared among all four cores.

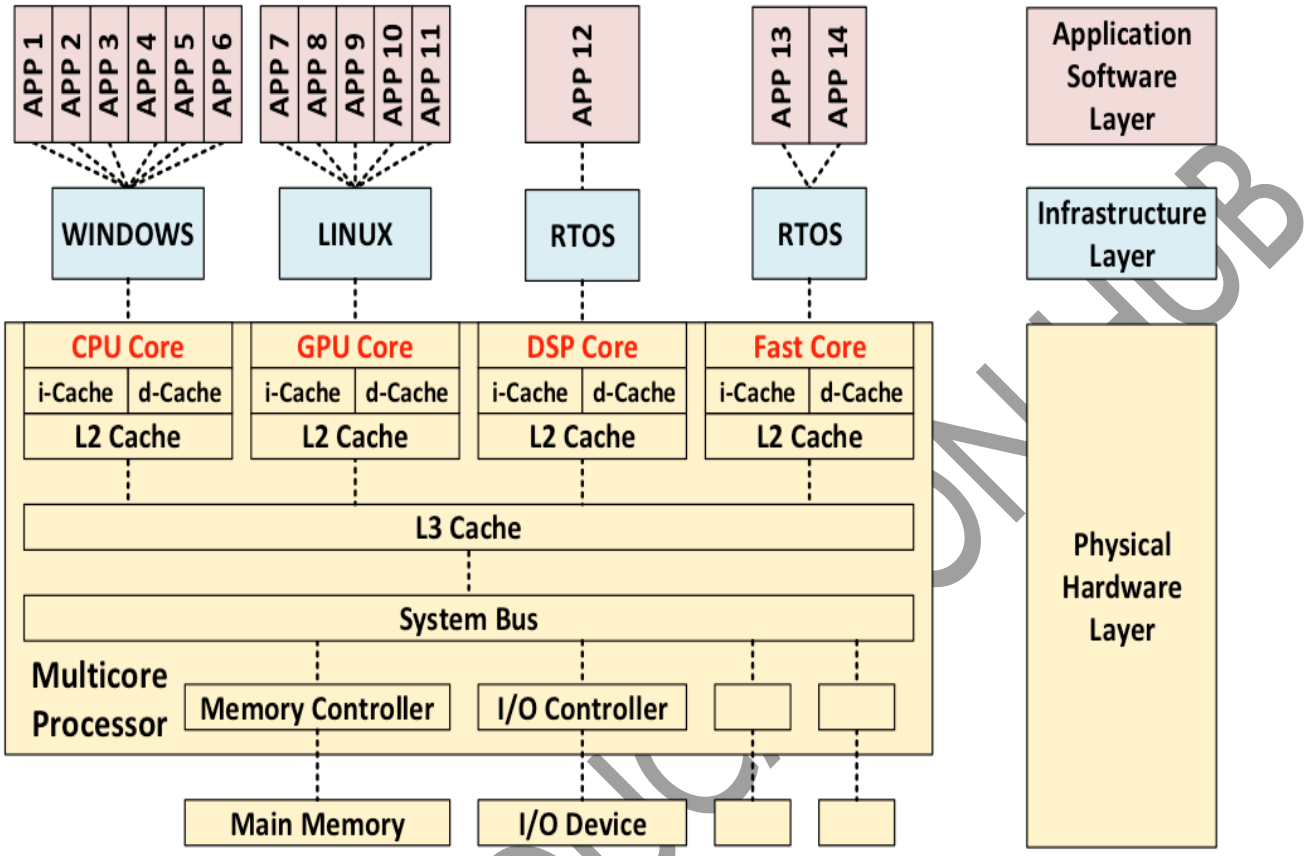
DIWAKAR EDUCATION HUB



Heterogeneous Multicore Processor

The following figure notionally shows how these 14 applications could be allocated to four different operating systems, which in turn are allocated to four different cores, in a heterogeneous, quad-core processor. From left to right, the cores include a general-purpose central processing unit core running Windows; a graphical processing unit (GPU) core running graphics-intensive applications on Linux; a digital signal processing (DSP) core running a real-time operating system (RTOS); and a high-performance core also running an RTOS.

DIWAKAR EDUCATION HUB



Current Trends in Multicore Processing

Multicore processors are replacing traditional, single-core processors so that fewer single-core processors are being produced and supported. Consequently, single-core processors are becoming technologically obsolete. Heterogeneous multicore processors, such as computer-on-a-chip processors, are becoming more common.

Although multicore processors have largely saturated some application domains (e.g., cloud computing, data warehousing, and on-line shopping), they are just starting to be used in real-time, safety- and security-critical, cyber-physical systems. One area in which multicore processing is becoming popular is in environments constrained by size, weight, and power, and cooling (SWAP-C), in which significantly increased performance is required.

Pros of Multicore Processing

DIWAKAR EDUCATION HUB

Multicore processing is typically commonplace because it offers advantages in the following seven areas:

1. **Energy Efficiency**. By using multicore processors, architects can decrease the number of embedded computers. They overcome increased heat generation due to Moore's Law (i.e., smaller circuits increase electrical resistance, which creates more heat), which in turn decreases the need for cooling. The use of multicore processing reduces power consumption (less energy wasted as heat), which increases battery life.

2. **True Concurrency**. By allocating applications to different cores, multicore processing increases the intrinsic support for actual (as opposed to virtual) parallel processing within individual software applications across multiple applications.

3. **Performance**. Multicore processing can increase performance by running multiple applications concurrently. The decreased distance between cores on an integrated chip enables shorter resource access latency and higher cache speeds when compared to using separate processors or computers. However, the size of the performance increase depends on the number of cores, the level of real concurrency in the actual software, and the use of shared resources.

4. **Isolation**. Multicore processors *may* improve (but do *not* guarantee) spatial and temporal isolation (segregation) compared to single-core architectures. Software running on one core is less likely to affect software on another core than if both are executing on the same single core. This decoupling is due to both spatial isolation (of data in core-specific cashes) and temporal isolation, because threads on one core are not delayed by threads on another core. Multicore processing may also improve robustness by localizing the impact of defects to single core. This increased isolation is particularly important in the independent execution of mixed-criticality applications (mission-critical, safety critical, and security-critical).

DIWAKAR EDUCATION HUB

5. **Reliability and Robustness**. Allocating software to multiple cores increases reliability and robustness (i.e., fault and failure tolerance) by limiting fault and/or failure propagation from software on one core to software on another. The allocation of software to multiple cores also supports failure tolerance by supporting failover from one core to another (and subsequent recovery).

6. **Obsolescence Avoidance**. The use of multicore processors enables architects to avoid technological obsolescence and improve maintainability. Chip manufacturers are applying the latest technical advances to their multicore chips. As the number of cores continues to increase, it becomes increasingly hard to obtain single-core chips.

7. **Hardware Costs**. By using multicore processors, architects can produce systems with fewer computers and processors.

Cons of Multicore Processing

Although there are many advantages to moving to multicore processors, architects must address disadvantages and associated risks in the following six areas:

1. **Shared Resources**. Cores on the same processor share both *processor-internal* resources (L3 cache, system bus, memory controller, I/O controllers, and interconnects) and *processor-external* resources (main memory, I/O devices, and networks). These shared resources imply (1) the existence of single points of failure, (2) two applications running on the *same* core can interfere with each other, and (3) software running on one core can impact software running on *another* core (i.e., interference can violate spatial and temporal isolation because multicore support for isolation is limited). The diagram below uses the color red to illustrate six shared resources.

2. **Interference**. Interference occurs when software executing on one core impacts the behavior of software executing on other cores in the same

DIWAKAR EDUCATION HUB

processor. This interference includes failures of both *spatial* isolation (due to shared memory access) and failure of *temporal* isolation (due to interference delays and/or penalties). Temporal isolation is a bigger problem than spatial isolation since multicore processors may have special hardware that can be used to enforce spatial isolation (to prevent software running on different cores from accessing the same processor-internal memory). The number of interference paths increases rapidly with the number of cores and the exhaustive analysis of all interference paths is often impossible. The impracticality of exhaustive analysis necessitates the selection of representative interference paths when analyzing isolation. The following diagram uses the color red to illustrate three possible interference paths between pairs of applications involving six shared resources.

3. **Concurrency Defects**. Cores execute concurrently, creating the potential for concurrency defects including deadlock, livelock, starvation, suspension, (data) race conditions, priority inversion, order violations, and atomicity violations. Note that these are essentially the same types of concurrency defects that can occur when software is allocated to multiple threads on a single core.

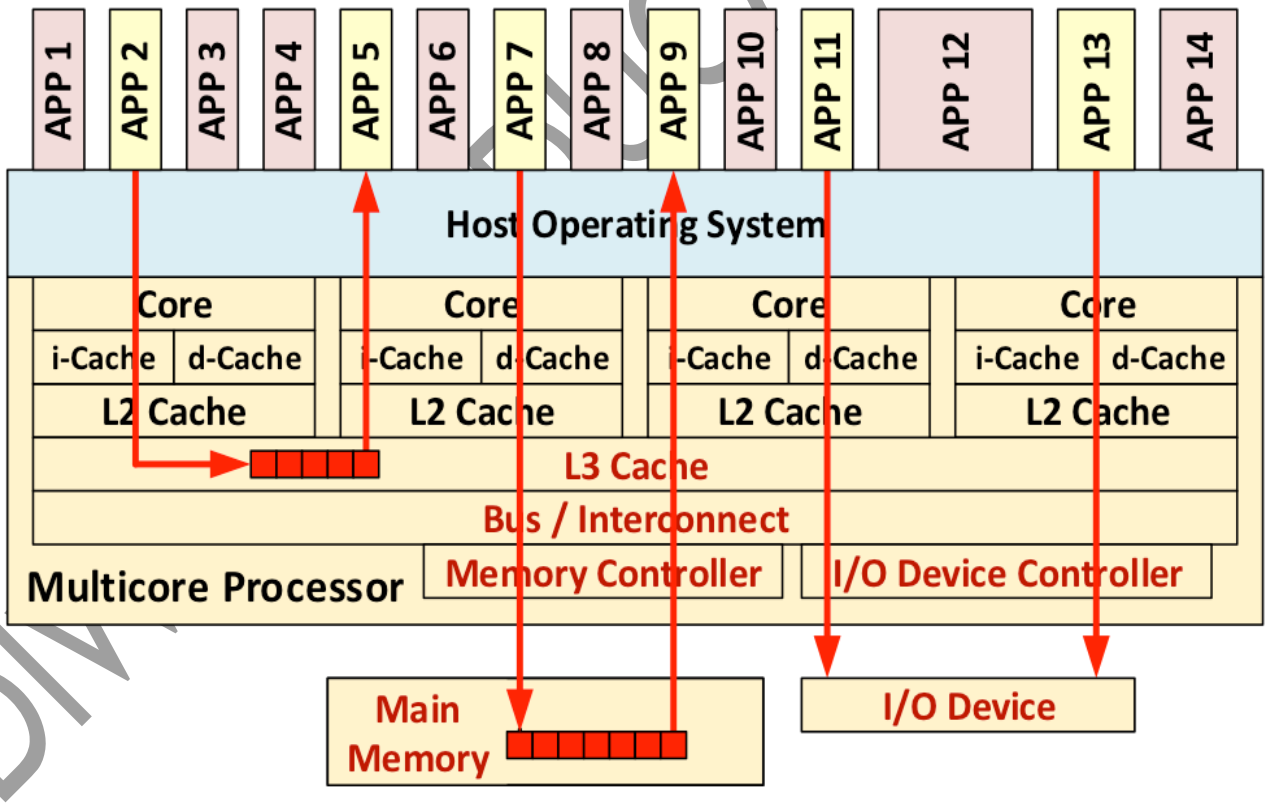
4. **Non-determinism**. Multicore processing increases non-determinism. For example, I/O Interrupts have top-level hardware priority (also a problem with single core processors). Multicore processing is also subject to lock trashing, which stems from excessive lock conflicts due to simultaneous access of kernel services by different cores (resulting in decreased concurrency and performance). The resulting non-deterministic behavior can be unpredictable, can cause related faults and failures, and can make testing more difficult (e.g., running the same test multiple times may not yield the same test result).

5. **Analysis Difficulty**. The real concurrency due to multicore processing requires different memory consistency models than virtual interleaved

DIWAKAR EDUCATION HUB

concurrency. It also breaks traditional analysis approaches for work on single core processors. The analysis of maximum time limits is harder and may be overly conservative. Although interference analysis becomes more complex as the number of cores-per-processor increases, overly-restricting the core number may not provide adequate performance.

6. **Accreditation and Certification**. Interference between cores can cause missed deadlines and excessive jitter, which in turn can cause faults (hazards) and failures (accidents). Verifying a multicore system requires proper real-time scheduling and timing analysis and/or specialized performance testing. Moving from a single-core to a multicore architecture may require recertification. Unfortunately, current safety policy guidelines are based on single-core architectures and must be updated based on the recommendations that will be listed in the final blog entry in this series.



The Division Algorithm

For all positive integers a and b,
where b ≠ 0,

$$a = qb + r$$

$$\frac{a}{b} = q + \frac{r}{b}$$

a divided by b gives a quotient and remainder.

The quotient q and remainder r are integers

$$0 \le r < |b|$$

Example

Use the division algorithm to find
the quotient and remainder when
a = 158 and b = 17

$$a = qb + r$$
$$158 = 9 \times 17 + 5$$

Convention uses a dot • to show multiplication
instead of a ×
This is fine , since we are purely dealing with integers.
No decimals are involved

$$158 = 9 \cdot 17 + 5$$
$$so \quad q = 9 \text{ and } r = 5$$

The Euclidean Algorithm

EDUCATION HUB

This uses the division algorithm to:-

- find the greatest common divisor (gcd)

[aka highest common factor (hcf)]

The greatest common denominator (gcd)
of two integers a and b , at least one of which $\neq 0$
is found when the remainder in the division algorithm is zero

$$a = qb + r$$

so $$a = qb + 0$$

- find the lowest common multiple (lcm) of two numbers

$$lcm = \frac{ab}{(a,b)}$$

- reduce a fraction to its simplest form

(just divide top and bottom by the gcd)

- find relatively prime (coprime) integers

These occur when the gcd (a,b) = 1

- Solve equations of the form

gcd (a,b) =ax +by

DIWAKAR EDUCATION HUB

If $ax + by = d$ where $\gcd(a,b) = d$

then

$$x_n = x_0 + \left(\frac{b}{d}\right)m$$

$$y_n = y_0 + \left(\frac{a}{d}\right)m$$

describes the general solution x_n, y_n

when the particular solutions x_0, y_0 are known

Example

Find the gcd of 135 and 1780

$$a = qb + r$$

$$1780 = 13 \bullet 135 + 25$$

Now continue, replacing

a with *b* and *b* with *r*

$$135 = 5 \bullet 25 + 10$$

$$25 = 2 \bullet 10 + 5$$

$$10 = 2 \bullet 5 + 0$$

$$\gcd(135, 1780) = 5$$

Example

Find the lcm of 135 and 1780

D' EDUCATION HUB

$$lcm = \frac{ab}{(a,b)}$$
$$= \frac{135 \bullet 1780}{5}$$
$$= \frac{240300}{5}$$
$$= 48060$$

$$\text{lcm}\ \ (135, 1780) = 48060$$

Example

Reduce the fraction 1480/128600 to its simplest form

$$a = qb + r$$
$$128600 = 86 \bullet 1480 + 1320$$
$$1480 = 1 \bullet 1320 + 160$$
$$1320 = 8 \bullet 160 + 40$$
$$160 = 4 \bullet 40 + 0$$

$$\gcd(128600, 1480) = 40$$

Divide top and bottom by the gcd

$$\frac{1480 \div 40}{128600 \div 40} = \frac{37}{3215}$$

Example

Show that 34 and 111 are co prime

DIWA... DUCATION HUB

$$a = qb + r$$
$$111 = 3 \bullet 34 + 9$$
$$34 = 3 \bullet 9 + 7$$
$$9 = 1 \bullet 7 + 2$$
$$7 = 3 \bullet 2 + 1$$
$$2 = 2 \bullet 1 + 0$$

$\gcd(34, 111) = 1$

$\therefore$ 34 and 111 are coprime.

Example

Solve 34x + 111y = 1 ,
where x and y are integers

$$a = qb + r$$
$$111 = 3 \bullet 34 + 9$$
$$34 = 3 \bullet 9 + 7$$
$$9 = 1 \bullet 7 + 2$$
$$7 = 3 \bullet 2 + 1$$
$$2 = 2 \bullet 1 + 0$$

DIWAKAR EDUCATION HUB

change subject to r and substitute
Start with the second last equation
and work backwards

$$\begin{aligned}1 &= 7-3\bullet 2\\ &= 7-3\bullet(9-1\bullet 7)\\ &= 7-3\bullet 9+3\bullet 7 = 4\bullet 7-3\bullet 9\\ &= 4\bullet(34-3\bullet 9)-3\bullet 9 = 4\bullet 34-15\bullet 9\\ &= 4\bullet 34-15\bullet(111-3\bullet 34)\\ &= 49\bullet 34-15\bullet 111\end{aligned}$$

compare to original

$34x+111y=1$

$\Rightarrow x=49 \qquad y=-15$

Diophantine Equations

These are of the form

$$ax^n+by^n=c^n$$

where all numbers are integers

DIWAKAR EDUCATION HUB

$ax + by = c$

has a solution only if the gcd is a factor of c

To solve

1) Find $\gcd(a,b) = d$

then $d \mid c$

so $c = dn$ for some integer n.

Express c in terms of d

2) Express d in the form $d = as + bt$ for some integers s and t

3) Multiply by n to get $x = sn \quad y = tn$

Example

Solve the linear Diophantine Equation

69x +27y = 1332, if it exists

$$69x + 27y = 1332$$

Find the gcd of 69 and 27

$$69 = 2 \bullet 27 + 15$$
$$27 = 1 \bullet 15 + 12$$
$$15 = 1 \bullet 12 + 3$$
$$12 = 4 \bullet 3 + 0$$

$\gcd(69,\ 27) = 3$

since $3 \mid 1332$, a solution exists

$c = dn \Rightarrow 1332 = 3n \Rightarrow n = 444$

$3 = 15 - 1 \bullet 12$

$= 15 - 1 \bullet (27 - 1 \bullet 15) = 15 - 1 \bullet 27 + 1 \bullet 15 = 2 \bullet 15 - 1 \bullet 27$

$= 2 \bullet (69 - 2 \bullet 27) - 1 \bullet 27$

$= 2 \bullet 69 - 4 \bullet 27 - 1 \bullet 27$

$= 2 \bullet 69 - 5 \bullet 27$

$d = 69s + 27t$

$\Rightarrow s = 2 \quad t = -5$

Multiply through by n

$x = 2n$ $\qquad$ $y = -5n$

$= 2 \bullet 444$ $\qquad$ $= -5 \bullet 444$

One solution is $x = 888, \quad y = -2220$

$$x_n = x_0 + \left(\frac{b}{d}\right)m \quad = 888 + \left(\frac{27}{3}\right)m = 888 + 9m$$

$$y_n = y_0 + \left(\frac{a}{d}\right)m \quad = -2220 + 13m$$

for some multiple m

Example

Find the positive integer values of x and y that satisfy

69x +27y = 1332

DUCATION HUB

From above, a solution exists

$69x + 27y = 1332$

$\gcd(69, 27) = 3$

$23x + 9y = 444$

solving for x

$$x = \frac{444 - 9y}{23}$$

$$= 19\frac{7}{23} - \frac{9y}{23}$$

$$= 19 - \frac{9y}{23} + \frac{7}{23}$$

$$= 19 - \frac{9y - 7}{23}$$

$$\therefore \quad \frac{9y - 7}{23} \leq 18$$

$$\Rightarrow y \leq \frac{23 \bullet 18 + 7}{9}$$

$$\Rightarrow y \leq \frac{421}{9} \leq 46\frac{7}{9}$$

$0 < y \leq 46, \quad \Delta y = 9$

Lowest possible is

$y = 11$

thus $x = \frac{444 - 99}{23} = 15$

D AR EDUCATION HUB

Alternatively, solving for y

$$y = \frac{444 - 23x}{9}$$

$$= 49 - \frac{23x - 3}{9}$$

$$\therefore \frac{23x - 3}{9} \le 48$$

$$\Rightarrow x \le \frac{48 \bullet 9 + 3}{23}$$

$$\Rightarrow x \le \frac{435}{23} \le 18\frac{21}{23}$$

$$0 < x \le 18, \quad \Delta x = 23$$

Lowest possible answers

$x = 15$

$$y = 49 - \frac{23 \bullet 15 - 3}{9}$$

$$= 49 - 38$$

$y = 11$

check

$$69x + 17y = 69 \bullet 15 + 17 \bullet 11 = 1035 + 187 = 1332$$

Pythagorean Triples

$$ax^2 + by^2 = c^2$$

To find these,

Pick an odd positive number

Divide its square into two integers which are as close to being equal as is possible

DIWAKAR EDUCATION HUB

e.g. $7^2 = 49 = 24 + 25$

gives triples 7, 24, 25

$7^2 + 24^2 = 25^2$

Alternatively, pick any even integer n

triples are 2n , n^2- 1 and $n^2 + 1$

e.g. picking 8 gives 16, 63 and 65

Indeed $16^2 + 63^2 = 65^2$

Fermat's LastTheorem

$$ax^n + by^n = c^n, \quad n > 2$$

cannot be solved with all as integers

Number Bases

To convert a number into a different base, use the Division Algorithm , taking b as the required base.

Example

Convert 36 into binary

$$a = qb + r$$

$$36 = 18 \bullet 2 + 0$$

Now continue, replacing *a* with *q*

$$18 = 9 \bullet 2 + 0$$

$$9 = 4 \bullet 2 + 1$$

$$4 = 2 \bullet 2 + 0$$

$$2 = 1 \bullet 2 + 0$$

$$1 = 0 \bullet 2 + 1$$

DIWAKAR EDUCATION HUB

Read the remainder upwards

$36 = 100100$ in base 2

The base is often shown as a subscript

$36 = 100100_2$

Example

Convert 36 into hexadecimal

$$a = qb + r$$

$$36 = 2 \bullet 16 + 4$$

$$2 = 0 \bullet 16 + 2$$

$36 = 24$ in base 16

Example

Convert 503793 into hexadecimal

(Remember that hexadecimal uses letters)

$$a = qb + r$$

$$503793 = 31487 \bullet 16 + 1$$

$$31487 = 1967 \bullet 16 + 15$$

$$1967 = 122 \bullet 16 + 15$$

$$122 = 7 \bullet 16 + 10$$

$$7 = 0 \bullet 16 + 7$$

$503793 = 7AFF1$ in base 16

$503793_{dec} = 7AFF1_{hex}$

DIWAKAR EDUCATION HUB

DIWAKAR EDUCATION HUB

Computer System Architecture Unit – 2 MCQs

As per updated syllabus

DIWAKAR EDUCATION HUB

2020

THE LEARN WITH EXPERTIES

1. Any signed negative binary number is recognised by its _____
a) MSB
b) LSB
c) Byte
d) Nibble

Answer: a
Explanation: Any negative number is recognized by its MSB (Most Significant Bit).
If it's 1, then ít's negative, else if it's 0, then positive.

2. The parameter through which 16 distinct values can be represented is known as _____
a) Bit
b) Byte
c) Word
d) Nibble

Answer: c
Explanation: It can be represented up to 16 different values with the help of a Word. Nibble is a combination of four bits and Byte is a combination of 8 bits. It is "word" which is said to be a collection of 16-bits on most of the systems.

3. If the decimal number is a fraction then its binary equivalent is obtained by _____ the number continuously by 2.
a) Dividing
b) Multiplying
c) Adding
d) Subtracting

Answer: b
Explanation: On multiplying the decimal number continuously by 2, the binary equivalent is obtained by the collection of the integer part. However, if it's an integer, then it's binary equivalent is determined by dividing the number by 2 and collecting the remainders.

4. The representation of octal number (532.2)8 in decimal is _____
a) (346.25)10
b) (532.864)10
c) (340.67)10
d) (531.668)10

Answer: a
Explanation: Octal to Decimal conversion is obtained by multiplying 8 to the power of base index along with the value at that index position.
$(532.2)8 = 5 * 8^2 + 3 * 8^1 + 2 * 8^0 + 2 * 8^{-1} = (346.25)10$

5. Which of the following is not a data type?
a) Symbolic Data
b) Alphanumeric Data
c) Numeric Data
d) Alphabetic Data

Answer: a
Explanation: Data types are of three basic types: Numeric, Alphabetic and Alphanumeric. Numeric Data consists of only numbers.
Alphabetic Data consists of only letters and a blank character and alphanumeric data consists of symbols.

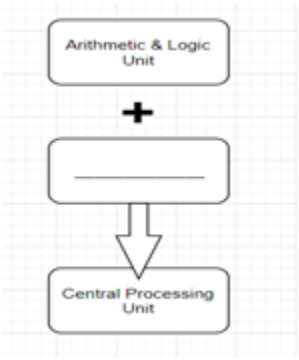
6. _____ is the raw material used as input and _____ is the processed data obtained as output of data processing.
a) Data, Instructions
b) Instructions, Program
c) Data, Program
d) Program, Code

Answer: a
Explanation: Data can be assumed as a raw material which, in turns after processing gives the desired output in the form of instructions. Further, a set of ordered and meaningful instructions is known as a program.

7. Which of the following is not a characteristic of a computer?
a) Diligence
b) I.Q.
c) Accuracy
d) Versatility

Answer: b
Explanation: The Computer system has no I.Q. of its own. It does only what it is programmed to do. It cannot take decisions of its own.
A computer is diligent because it can work continuously for hours without getting any errors or without getting grumbled.
The accuracy of a computer is consistently high and its level of accuracy depends on its design. A computer can perform any task if, it can be broken down into a series of logical steps. Therefore, a computer is versatile.

8. Fill in the blank in the diagram.



a) Input Unit
b) Memory Unit
c) Control Unit
d) I/O Unit

Answer: c
Explanation: The control unit manages and coordinates the operations of a computer system.
The ALU is responsible for performing all the arithmetic and bitwise operations . Therefore, both these

units combine to form the brain of the computer ,which is the central processing unit.

9. The part of a processor which contains hardware necessary to perform all the operations required by a computer:
a) Data path
b) Controller
c) Registers
d) Cache

Answer: a
Explanation: A processor is a part of the computer which does all the data manipulation and decision making. A processor comprises of:
A data path which contains the hardware necessary to perform all the operations. A controller tells the data path what needs to be done. The registers act as intermediate storage for the data.

10. What does MAR stand for?
a) Main Address Register
b) Memory Access Register
c) Main Accessible Register
d) Memory Address Register

Answer: d
Explanation: MAR is a type of register which is responsible for the fetch operation. MAR is connected to the address bus and it specifies the address for the read and write operations.

11. If the control signals are generated by combinational logic, then they are generated by a type of _____ controlled unit.
a) Micro programmed
b) Software
c) Logic
d) Hardwired

Answer: d
Explanation: The main task of a control unit is to generate control signals. There are two main types of control units:
A hardwired control unit generates control signals by using combinational logic circuits and the Micro programmed control unit generates control signals by using some softwares.

12. Which is the simplest method of implementing hardwired control unit?
a) State Table Method
b) Delay Element Method
c) Sequence Counter Method
d) Using Circuits

Answer: a
Explanation: There are 3 ways of implementing hardwired control unit:
A state table is the simplest method

in which a number of circuits are designed based on the cells in the table.
A delay element method consists of a flowchart drawn for the circuit. A D-flip flop is used as a delay element.
A sequence counter method used k-modulo counter as a replacement for k delay elements.

13. A set of microinstructions for a single machine instruction is called _____
a) Program
b) Command
c) Micro program
d) Micro command

Answer: c
Explanation: For every micro-operation, a set of microinstructions are written which indicate the control signals to be activated. A set of microinstructions is a micro program. The address of the next microinstruction is given by a Micro-program counter.

14. Micro-program consists of a set of microinstructions which are strings of 0s and 1s.
a) True
b) False

Answer: a
Explanation: The computer understands only binary language. So, the micro-program should have instructions which are in the form of 0s and 1s. Each output line of the micro-program corresponds to one control signal.

15. A decoder is required in case of a _____
a) Vertical Microinstruction
b) Horizontal Microinstruction
c) Multilevel Microinstruction
d) All types of microinstructions

Answer: a
Explanation: There are two types of microinstructions: Horizontal and Vertical.
In a horizontal microinstruction, each bit represents a signal to be activated whereas, in case of vertical microinstruction bits are decoded and, the decoder then produces signals.

16. In boolean algebra, the OR operation is performed by which properties?
a) Associative properties
b) Commutative properties
c) Distributive properties
d) All of the Mentioned

Answer: d
Explanation: The expression for

Associative property is given by A+(B+C) = (A+B)+C & A*(B*C) = (A*B)*C.
The expression for Commutative property is given by A+B = B+A & A*B = B*A.
The expression for Distributive property is given by A+BC=(A+B)(A+C) & A(B+C) = AB+AC.

17. The expression for Absorption law is given by _____
a) A + AB = A
b) A + AB = B
c) AB + AA' = A
d) A + B = B + A

Answer: a
Explanation: The expression for Absorption Law is given by: A+AB = A.
Proof: A + AB = A(1+B) = A (Since 1 + B = 1 as per 1's Property).

18. According to boolean law: A + 1 = ?
a) 1
b) A
c) 0
d) A'

Answer: a
Explanation: A + 1 = 1, as per 1's Property.

19. The involution of A is equal to _____
a) A
b) A'
c) 1
d) 0

Answer: a
Explanation: The involution of A means double inversion of A (i.e. A") and is equal to A.
Proof: ((A)')' = A

20. A(A + B) = ?
a) AB
b) 1
c) (1 + AB)
d) A

Answer: d
Explanation: A(A + B) = AA + AB (By Distributive Property) = A + AB (A.A = A By Commutative Property) = A(1 + B) = A*1 (1 + B = 1 by 1's Property) = A.

21. DeMorgan's theorem states that _____
a) (AB)' = A' + B'
b) (A + B)' = A' * B
c) A' + B' = A'B'
d) (AB)' = A' + B

DIWAKAR EDUCATION HUB

Answer: a
Explanation: The DeMorgan's law states that (AB)' = A' + B' & (A + B)' = A' * B', as per the Dual Property.

22. (A + B)(A' * B') = ?
a) 1
b) 0
c) AB
d) AB'

Answer: b
Explanation: The DeMorgan's law states that (AB)' = A' + B' & (A + B)' = A' * B', as per the Dual Property.

23. Complement of the expression A'B + CD' is _____
a) (A' + B)(C' + D)
b) (A + B')(C' + D)
c) (A' + B)(C' + D)
d) (A + B')(C + D')

Answer: b
Explanation: (A'B + CD')' = (A'B)'(CD')' (By DeMorgan's Theorem) = (A" + B')(C' + D") (By DeMorgan's Theorem) = (A + B')(C' + D).

24. Simplify Y = AB' + (A' + B)C.
a) AB' + C
b) AB + AC
c) A'B + AC'
d) AB + A

Answer: a
Explanation: Y = AB' + (A' + B)C = AB' + (AB')'C = (AB' + C)(AB' + AB') = (AB' + C).1 = (AB' + C).

25. The boolean function A + BC is a reduced form of _____
a) AB + BC
b) (A + B)(A + C)
c) A'B + AB'C
d) (A + C)B

Answer: b
Explanation: (A + B)(A + C) = AA + AC + AB + BC = A + AC + AB + BC (By Commutative Property) = A(1 + C + B) + BC = A + BC (1 + B + C =1 By 1's Property).

26. What is an ambiguous condition in a NAND based S'-R' latch?
a) S'=0, R'=1
b) S'=1, R'=0
c) S'=1, R'=1
d) S'=0, R'=0

Answer: d
Explanation: In a NAND based S-R latch, If S'=0 & R'=0 then both the outputs (i.e. Q & Q') goes HIGH and this condition is called as ambiguous/forbidden state. This state is also known as an Invalid state

DIWAKAR EDUCATION HUB

as the system goes into an unexpected situation.

27. In a NAND based S'-R' latch, if S'=1 & R'=1 then the state of the latch is _____
a) No change
b) Set
c) Reset
d) Forbidden

Answer: a
Explanation: In a NAND based S'-R, latch if S'=1 & R'=1 then there is no any change in the state. It remains in its prior state. This state is used for the storage of data.

28. A NAND based S'-R' latch can be converted into S-R latch by placing _____
a) A D latch at each of its input
b) An inverter at each of its input
c) It can never be converted
d) Both a D latch and an inverter at its input

Answer: d
Explanation: A NAND based S'-R' latch can be converted into S-R latch by placing either a D latch or an inverter at its input as it's operations will be complementary.

29. One major difference between a NAND based S'-R' latch & a NOR based S-R latch is _____
a) The inputs of NOR latch are 0 but 1 for NAND latch
b) The inputs of NOR latch are 1 but 0 for NAND latch
c) The output of NAND latch becomes set if S'=0 & R'=1 and vice versa for NOR latch
d) The output of NOR latch is 1 but 0 for NAND latch

Answer: a
Explanation: Due to inverted input of NAND based S'-R' latch, the inputs of NOR latch are 0 but 1 for NAND latch.

30. The characteristic equation of S-R latch is _____
a) Q(n+1) = (S + Q(n))R'
b) Q(n+1) = SR + Q(n)R
c) Q(n+1) = S'R + Q(n)R
d) Q(n+1) = S'R + Q'(n)R

Answer: a
Explanation: A characteristic equation is needed when a specific gate requires a specific output in order to satisfy the truth table. The characteristic equation of S-R latch is Q(n+1) = (S + Q(n))R'.

DIWAKAR EDUCATION HUB

31. The difference between a flip-flop & latch is _____
a) Both are same
b) Flip-flop consist of an extra output
c) Latches has one input but flip-flop has two
d) Latch has two inputs but flip-flop has one

Answer: c
Explanation: Flip-flop is a modified version of latch. To determine the changes in states, an additional control input is provided to the latch.

32. How many types of flip-flops are?
a) 2
b) 3
c) 4
d) 5

Answer: c
Explanation: There are 4 types of flip-flops, viz., S-R, J-K, D, and T. D flip-flop is an advanced version of S-R flip-flop, while T flip-flop is an advanced version of J-K flip-flop.

33. The S-R flip flop consist of _____
a) 4 AND gates
b) Two additional AND gates
c) An additional clock input
d) 3 AND gates

Answer: b
Explanation: The S-R flip flop consist of two additional AND gates at the S and R inputs of S-R latch.

34. What is one disadvantage of an S-R flip-flop?
a) It has no Enable input
b) It has a RACE condition
c) It has no clock input
d) Invalid State

Answer: d
Explanation: The main drawback of s-r flip flop is invalid output when both the inputs are high, which is referred to as Invalid State.

35. One example of the use of an S-R flip-flop is as _____
a) Racer
b) Stable oscillator
c) Binary storage register
d) Transition pulse generator

Answer: c
Explanation: S-R refers to set-reset. So, it is used to store two values 0 and 1. Hence, it is referred to as binary storage element. It functions as memory storage during the No Change State.

DIWAKAR EDUCATION HUB

36. When is a flip-flop said to be transparent?
a) When the Q output is opposite the input
b) When the Q output follows the input
c) When you can see through the IC packaging
d) When the Q output is complementary of the input

Answer: b
Explanation: Flip-flop have the property of responding immediately to the changes in its inputs. This property is called transparency.

37. On a positive edge-triggered S-R flip-flop, the outputs reflect the input condition when _____
a) The clock pulse is LOW
b) The clock pulse is HIGH
c) The clock pulse transitions from LOW to HIGH
d) The clock pulse transitions from HIGH to LOW

Answer: c
Explanation: Edge triggered device will follow when there is transition. It is a positive edge triggered when transition takes place from low to high, while, it is negative edge triggered when the transition takes place from high to low.

38. A group of bits used to represent a symbol is called a _____
a) byte
b) memory
c) nibble
d) code

Answer: a
Explanation: In binary coding, every symbol that appears in data is represented by a group of bits, which are called bytes. Computer codes use binary coding schemes.

39. BCD uses 6 bits to represent a symbol.
a) True
b) False

Answer: a
Explanation: In a Binary Coded Decimal format, 64 characters i.e. 26 different characters can be represented. It is one of the early computer codes.

40. Which of the following is not a type of computer code?
a) EBCDIC
b) BCD

DIWAKAR EDUCATION HUB

c) ASCII
d) EDIC

Answer: d
Explanation: There is no coding scheme like EDIC. EBCDIC stands for Extended Binary Coded Decimal Interchange Code. BCD stands for Binary Coded Decimal. ASCII stands for American Standard Code for information interchange.

41. The BCD representation of $(34)_{10}$ is _____
a) 6
b) 7
c) 8
d) 5

Answer: b
Explanation: BCD numbers are represented as:
34 = $(0011\ 0100)_{BCD}$.
Each digit is individually taken and an equivalent standard 4 bit term is written for the respective digit.

42. Perform BCD addition of $(23)_{BCD}$ + $(20)_{BCD}$.
a) 00110100
b) 01000011
c) 10011
d) 11100

Answer: b
Explanation: To add any two BCD numbers :
Simply perform the addition :
23+20=43.
Then, write the equivalent BCD number = $(0100\ 0011)_{BCD}$.

43. The weights used in Binary coded decimal code are:
a) 4,2,1
b) 8,4,2,1
c) 6,4,2,1
d) 2,1

Answer: b
Explanation: BCD is a weighted code and it uses the weights 8,4,2,1 respectively. It is often called the 8421 code. Since, it uses 4 bits for the representation therefore the weights are assigned as : $2^3 = 8$, $2^2 = 4$, $2^1 = 2$, $2^0 = 1$.

44. Write the decimal equivalent for $(110001)_{BCD}$.
a) 31
b) 13
c) C1
d) 1C

Answer: a
Explanation: To obtain the decimal equivalent :

DIWAKAR EDUCATION HUB

We start from the rightmost bit and make groups of 4, then write the decimal equivalent accordingly.
0011 0001 = $(31)_{10}$.

45. The 9's complement of 45 is _____
a) 45
b) 54
c) 64
d) 46

Answer: b
Explanation: The 9's complement of a number is obtained by subtracting each digit from 9. Here, 99-45=54. Therefore, the 9's complement is 54.

46. The 10's complement of 455 is _____
a) 543
b) 544
c) 545
d) 546

Answer: c
Explanation: To obtain the 10's complement, we first obtain the 9's complement and then add 1 to it.
999-455=544 (9's)
544+1=545(10's).

47. The Excess-3 representation of $(0100)_{BCD}$ is _____
a) 0110
b) 1110
c) 0111
d) 1100

Answer: c
Explanation: The excess-3 code is obtained by adding 3 to the BCD code.
Here, 0100+0011=0111.
Also, 4+3=7.

48. What is the hold condition of a flip-flop?
a) Both S and R inputs activated
b) No active S or R input
c) Only S is active
d) Only R is active

Answer: b
Explanation: The hold condition in a flip-flop is obtained when both of the inputs are LOW. It is the No Change State or Memory Storage state if a flip-flop.

49. If an active-HIGH S-R latch has a 0 on the S input and a 1 on the R input and then the R input goes to 0, the latch will be _____
a) SET
b) RESET
c) Clear
d) Invalid

DIWAKAR EDUCATION HUB

Answer: b
Explanation: If S=0, R=1, the flip flop is at reset condition. Then at S=0, R=0, there is no change. So, it remains in reset. If S=1, R=0, the flip flop is at the set condition.

50. The circuit that is primarily responsible for certain flip-flops to be designated as edge-triggered is the _____
a) Edge-detection circuit
b) NOR latch
c) NAND latch
d) Pulse-steering circuit

Answer: a
Explanation: The circuit that is primarily responsible for certain flip-flops to be designated as edge-triggered is the edge-detection circuit.

51. The output of a logic gate is 1 when all the input are at logic 0 as shown below:

INPUT		OUTPUT
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

INPUT		OUTPUT
A	B	C
0	0	1
0	1	0
1	0	0
1	1	1

The gate is either _____
a) A NAND or an EX-OR
b) An OR or an EX-NOR
c) An AND or an EX-OR
d) A NOR or an EX-NOR

Answer: d
Explanation: The output of a logic

gate is 1 when all inputs are at logic 0. The gate is NOR. The output of a logic gate is 1 when all inputs are at logic 0 or all inputs are at logic 1, then it is EX-NOR. (The truth tables for NOR and EX-NOR Gates are shown in above table).

52. The code where all successive numbers differ from their preceding number by single bit is _____
a) Alphanumeric Code
b) BCD
c) Excess 3
d) Gray

Answer: d
Explanation: The code where all successive numbers differ from their preceding number by single bit is gray code. It is an unweighted code. The most important characteristic of this code is that only a single bit change occurs when going from one code number to next. BCD Code is one in which decimal digits are represented by a group of 4-bits each, whereas, in Excess-3 Code, the decimal numbers are incremented by 3 and then written in their BCD format.

53. The following switching functions are to be implemented using a decoder:
f1 = ∑m(1, 2, 4, 8, 10, 14) f2 = ∑m(2, 5, 9, 11) f3 = ∑m(2, 4, 5, 6, 7)
The minimum configuration of decoder will be _____
a) 2 to 4 line
b) 3 to 8 line
c) 4 to 16 line
d) 5 to 32 line

Answer: c
Explanation: 4 to 16 line decoder as the minterms are ranging from 1 to 14.

54. How many AND gates are required to realize Y = CD + EF + G?
a) 4
b) 5
c) 3
d) 2

Answer: d
Explanation: To realize Y = CD + EF + G, two AND gates are required and two OR gates are required.

55. The NOR gate output will be high if the two inputs are _____
a) 00
b) 01
c) 10
d) 11

DIWAKAR EDUCATION HUB

Answer: a
Explanation: In 01, 10 or 11 output is low if any of the I/P is high. So, the correct option will be 00.

56. How many two-input AND and OR gates are required to realize Y = CD+EF+G?
a) 2, 2
b) 2, 3
c) 3, 3
d) 3, 2

Answer: a
Explanation: Y = CD + EF + G
The number of two input AND gate = 2
The number of two input OR gate = 2.

57. A universal logic gate is one which can be used to generate any logic function. Which of the following is a universal logic gate?
a) OR
b) AND
c) XOR
d) NAND

Answer: d
Explanation: An Universal Logic Gate is one which can generate any logic function and also the three basic gates: AND, OR and NOT. Thus, NOR and NAND can generate any logic function and are thus Universal Logic Gates.

58. A full adder logic circuit will have _____
a) Two inputs and one output
b) Three inputs and three outputs
c) Two inputs and two outputs
d) Three inputs and two outputs

Answer: d
Explanation: A full adder circuit will add two bits and it will also accounts the carry input generated in the previous stage. Thus three inputs and two outputs (Sum and Carry) are there. In case of half adder circuit, there are only two inputs bits and two outputs (SUM and CARRY).

59. How many two input AND gates and two input OR gates are required to realize Y = BD + CE + AB?
a) 3, 2
b) 4, 2
c) 1, 1
d) 2, 3

Answer: a
Explanation: There are three product terms. So, three AND gates of two inputs are required. As only two input OR gates are available, so two OR

DIWAKAR EDUCATION HUB

gates are required to get the logical sum of three product terms.

60. Which of following are known as universal gates?
a) NAND & NOR
b) AND & OR
c) XOR & OR
d) EX-NOR & XOR

Answer: a
Explanation: The NAND & NOR gates are known as universal gates because any digital circuit can be realized completely by using either of these two gates, and also they can generate the 3 basic gates AND, OR and NOT.

61. The gates required to build a half adder are _____
a) EX-OR gate and NOR gate
b) EX-OR gate and OR gate
c) EX-OR gate and AND gate
d) EX-NOR gate and AND gate

Answer: c
Explanation: The gates required to build a half adder are EX-OR gate and AND gate. EX-OR outputs the SUM of the two input bits whereas AND outputs the CARRY of the two input bits.

62. The CISC stands for _____
a) Computer Instruction Set Compliment
b) Complete Instruction Set Compliment
c) Computer Indexed Set Components
d) Complex Instruction set computer

Answer: d
Explanation: CISC is a computer architecture where in the processor performs more complex operations in one step.

63. The computer architecture aimed at reducing the time of execution of instructions is _____
a) CISC
b) RISC
c) ISA
d) ANNA

Answer: b
Explanation: The RISC stands for Reduced Instruction Set Computer.

3. The Sun micro systems processors usually follow _____ architecture.
a) CISC
b) ISA
c) ULTRA SPARC
d) RISC

DIWAKAR EDUCATION HUB

Answer: d
Explanation: The Risc machine aims at reducing the instruction set of the computer.

64. The RISC processor has a more complicated design than CISC.
a) True
b) False

Answer: b
Explanation: The RISC processor design is more simpler than CISC and it consists of fewer transistors.

65. The iconic feature of the RISC machine among the following is _____
a) Reduced number of addressing modes
b) Increased memory size
c) Having a branch delay slot
d) All of the mentioned

Answer: c
Explanation: A branch delay slot is an instruction space immediately following a jump or branch.

66. Both the CISC and RISC architectures have been developed to reduce the _____
a) Cost
b) Time delay
c) Semantic gap
d) All of the mentioned

Answer: c
Explanation: The semantic gap is the gap between the high level language and the low level language.

67. Out of the following which is not a CISC machine.
a) IBM 370/168
b) VAX 11/780
c) Intel 80486
d) Motorola A567

Answer: d
Explanation: None.

68. Pipe-lining is a unique feature of _____
a) RISC
b) CISC
c) ISA
d) IANA

Answer: a
Explanation: The RISC machine architecture was the first to implement pipe-lining.

69. In CISC architecture most of the complex instructions are stored in _____
a) Register
b) Diodes

DIWAKAR EDUCATION HUB

c) CMOS
d) Transistors

Answer: d
Explanation: In CISC architecture more emphasis is given on the instruction set and the instructions take over a cycle to complete.

70. Which of the architecture is power efficient?
a) CISC
b) RISC
c) ISA
d) IANA

Answer: b
Explanation: Hence the RISC architecture is followed in the design of mobile devices.

71. A task carried out by the OS and hardware to accommodate multiple processes in main memory.
a) Memory control
b) Memory management
c) Memory sharing
d) Memory usage

Answer: b
Explanation: Memory management is carried out by the OS and hardware to accommodate multiple processes in main memory.

72. An HTML file is a text file containing small markup tags.
a) True
b) False

Answer: a
Explanation: The statement is true. HTML stands for Hyper Text Markup Language. It is a text file containing small markup tags.

73. Secondary memory is the long term store for programs and data while main memory holds program and data currently in use. What kind of an organization is this?
a) Physical
b) Logical
c) Structural
d) Simple

Answer: a
Explanation: The secondary memory is the long term store for programs and data while main memory holds program and data currently in use. This is a physical organization.

74. Memory organization in which users write programs in modules with different characteristics.
a) Physical
b) Logical

DIWAKAR EDUCATION HUB

c) Structural
d) Simple

Answer: b
Explanation: The answer is Logical. To handle user programs properly, the operating system and the hardware should support a basic form of module to provide protection and sharing.

75. An executing process must be loaded entirely in main memory. What kind of a memory organization is this?
a) Physical
b) Logical
c) Structural
d) Simple

Answer: d
Explanation: This is simple memory organisation. An executing process must be loaded entirely in main memory (if overlays are not used).

76. FTP stands for?
a) File Text Protocol
b) File Transfer Protocol
c) Firm Transfer Protocol
d) File Transplant Protocol

Answer: b
Explanation: FTP stands for File Transfer Protocol. It is a type of internet service use for the transmission of files.

77. A set of overlapping divisions in the main memory are called _____
a) Partitions
b) Divisions
c) Blocks
d) Modules

Answer: a
Explanation: Partition main memory into a set of non overlapping regions called partitions. Partitions can be of equal or unequal sizes.

78. Any program, no matter how small, occupies an entire partition. This is called _____
a) fragmentation
b) prior fragmentation
c) internal fragmentation
d) external fragmentation

Answer: c
Explanation: It is called as internal fragmentation. Main memory use is inefficient. Any program, no matter how small, occupies an entire partition. This is called internal fragmentation.

DIWAKAR EDUCATION HUB

79. _____ is used to shift processes so they are contiguous and all free memory is in one block.
a) Fragmentation
b) Compaction
c) External Fragmentation
d) Division

Answer: b
Explanation: Use compaction to shift processes so they are contiguous and all free memory is in one block.

80. _____ searches for smallest block. The fragment left behind is small as possible.
a) best fit
b) first fit
c) next fit
d) last fit

Answer: a
Explanation: Best fit searches for the smallest block. The fragment left behind is as small as possible.

81. What is the static charge that can be stored by your body as you walk across a carpet?
a) 300 volts
b) 3000 volts
c) 30000 volts
d) Over 30000 volts

Answer: d
Explanation: When a person walks across a carpeted or tile floor electric charge builds up in the body due to the friction between shoes and floor material. If the friction static is greater the voltage potential develop in the body will be greater. You start act as a capacitor. This is called Electrostatic discharge. The potential static charge that can develop from walking on tile floors is greater than 15000 volts while carpeted floors can generate in excess of 30000 volts.

82. What must be done to interface TTL to CMOS?
a) A dropping resistor must be used on the CMOS of 12 V supply to reduce it to 5 V for the TTL
b) As long as the CMOS supply voltage is 5 V they can be interfaced (however, the fan-out of the TTL is limited to five CMOS gates)
c) A 5 V zener diode must be placed across the inputs of the TTL gates in order to protect them from the higher output voltages of the CMOS gates
d) A pull-up resistor must be used between the TTL output-CMOS input node and Vcc; the value of RP will

DIWAKAR EDUCATION HUB

depend on the number of CMOS gates connected to the node

Answer: d
Explanation: To interface TTL to CMOS a pull-up resistor must be used between the TTL output-CMOS input node and Vcc. A pull-up resistor is used to avoid the floating state on the input node of the CMOS, thus using a small amount of current. The value of RP will depend on the number of CMOS gates connected to the node.

83. What causes low-power Schottky TTL to use less power than the 74XX series TTL?
a) The Schottky-clamped transistor
b) A larger value resistor
c) The Schottky-clamped MOSFET
d) A small value resistor

Answer: b
Explanation: A larger value resistor causes low power low-power Schottky TTL to use less power than the 74XX series TTL.

84. What are the major differences between the 5400 and 7400 series of ICs?
a) The 5400 series are military grade and require tighter supply voltages and temperatures
b) The 5400 series are military grade and allow for a wider range of supply voltages and temperatures
c) The 7400 series are an improvement over the original 5400s
d) The 7400 series was originally developed by Texas Instruments and the 5400 series was brought out by National Semiconductors after TI's patents expired as a second supply source

Answer: b
Explanation: The 5400 series are military grade and allow for a wider range of supply voltages and temperatures, these are the major differences between the 5400 and 7400 series of ICs. Also, the working temperature range of 5400 series is -50 to 125C while that for 7400 is 0 to 70C.

85. Which of the following statements apply to CMOS devices?
a) The devices should not be inserted into circuits with the power on
b) All tools, test equipment and metal workbenches should be tied to earth ground
c) The devices should be stored and shipped in antistatic tubes or

DIWAKAR EDUCATION HUB

conductive foam
d) All of the Mentioned

Answer: d
Explanation: For CMOS devices, all the mentioned statements are applicable. The devices should not be inserted into circuits with the power on. All tools, test equipment and metal workbenches should be tied to earth ground. Also, the devices should be stored and shipped in antistatic tubes or conductive foam.

86. Brain of computer is _____
a) Control unit
b) Arithmetic and Logic unit
c) Central Processing Unit
d) Memory

Answer: c
Explanation: The CPU is referred to as the brain of a computer.
It consists of a control unit and the arithmetic and logic unit. It is responsible for performing all the processes and operations.

87. Control Unit acts as the central nervous system of the computer.
a) True
b) False

Answer: a
Explanation: The control unit is referred to as the central nervous system because it selects and interprets the instructions and coordinates execution.

88. What does MBR stand for?
a) Main Buffer Register
b) Memory Buffer Routine
c) Main Buffer Routine
d) Memory Buffer Register

Answer: d
Explanation: The binary subtraction 0 – 1 gives the result 1.
A borrow of 1 is although generated and is removed from the next higher column.

89. In the instruction ADD A, B, the answer gets stored in _____
a) B
b) A
c) Buffer
d) C

Answer: b
Explanation: In any instruction of the form ADD A, B; the answer gets stored in the A register. The format is: ADD Destination, Source.

DIWAKAR EDUCATION HUB

90. What is the high speed memory between the main memory and the CPU called?
a) Register Memory
b) Cache Memory
c) Storage Memory
d) Virtual Memory

Answer: b
Explanation: It is called the Cache Memory. The cache memory is the high speed memory between the main memory and the CPU.

91. Cache Memory is implemented using the DRAM chips.
a) True
b) False

Answer: b
Explanation: The Cache memory is implemented using the SRAM chips and not the DRAM chips. SRAM stands for Static RAM. It is faster and is expensive.

92. Whenever the data is found in the cache memory it is called as _____
a) HIT
b) MISS
c) FOUND
d) ERROR

Answer: a
Explanation: Whenever the data is found in the cache memory, it is called as Cache HIT. CPU first checks in the cache memory since it is closest to the CPU.

93. LRU stands for _____
a) Low Rate Usage
b) Least Rate Usage
c) Least Recently Used
d) Low Required Usage

Answer: c
Explanation: LRU stands for Least Recently Used. LRU is a type of replacement policy used by the cache memory.

94. When the data at a location in cache is different from the data located in the main memory, the cache is called _____
a) Unique
b) Inconsistent
c) Variable
d) Fault

Answer: b
Explanation: The cache is said to be inconsistent. Inconsistency must be avoided as it leads to serious data bugs.

DIWAKAR EDUCATION HUB

95. Which of the following is not a write policy to avoid Cache Coherence?
a) Write through
b) Write within
c) Write back
d) Buffered write

Answer: b
Explanation: There is no policy which is called as the write within policy. The other three options are the write policies which are used to avoid cache coherence.

96. What does PC stand for?
a) Program Changer
b) Program Counter
c) Performance Counter
d) Performance Changer

Answer: b
Explanation: The Program counter contains the address of the next instruction which is to be fetched by the control unit.
All other options are invalid.

97. Which of the following holds the last instruction fetched?
a) PC
b) MAR
c) MBR
d) IR

Answer: d
Explanation: The IR which stands for the instruction register contains the last instruction fetched.
All the others options are registers which are used for the fetch operation.

98. The portion of the processor which contains the hardware required to fetch the operations is _____
a) Datapath
b) Processor
c) Control
d) Output unit

Answer: a
Explanation: The datapath contains the hardware required to fetch the operations. The control tells the data path what needs to be done.

99. Causing the CPU to step through a series of micro operations is called _____
a) Execution
b) Runtime
c) Sequencing
d) Pipelining

Answer: c
Explanation: Sequencing is the process of causing the CPU to step

DIWAKAR EDUCATION HUB

through a series of micro operations. Execution causes the performance of each micro operation.

100. The functions of execution and sequencing are performed by using _____

a) Input Signals
b) Output Signals
c) Control Signals
d) CPU

Answer: c
Explanation: Sequencing followed by the process of execution is performed by the Control signals. Sequencing is traversing each and every operation whereas execution causes the performance of each operation.

101. What does D in the D-flip flop stand for?
a) Digital
b) Direct
c) Delay
d) Durable

Answer: c
Explanation: In the hardwired control unit, the delay element method uses D-flip flop which causes a delay. Since, in the delay element method, there must be a finite time gap between the 2 steps.

102. All input of NOR as low produces result as _____
a) Low
b) Mid
c) High
d) Floating

Answer: c
Explanation: All input of NOR as low produces the result as high, whereas, rest all conditions produce output as low.

103. What is a multiplexer?
a) It is a type of decoder which decodes several inputs and gives one output
b) A multiplexer is a device which converts many signals into one
c) It takes one input and results into many output
d) It is a type of encoder which decodes several inputs and gives one output

Answer: b
Explanation: 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, depending on the active select lines.

DIWAKAR EDUCATION HUB

104. Which combinational circuit is renowned for selecting a single input from multiple inputs & directing the binary information to output line?
a) Data Selector
b) Data distributor
c) Both data selector and data distributor
d) DeMultiplexer

Answer: a
Explanation: Data Selector is another name of Multiplexer. 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, depending on the active select lines.

105. It is possible for an enable or strobe input to undergo an expansion of two or more MUX ICs to the digital multiplexer with the proficiency of large number of _____
a) Inputs
b) Outputs
c) Selection lines
d) Enable lines

Answer: a
Explanation: It is possible for an enable or strobe input to undergo an expansion of two or more MUX ICs to the digital multiplexer with the proficiency of large number of inputs.

106. Which is the major functioning responsibility of the multiplexing combinational circuit?
a) Decoding the binary information
b) Generation of all minterms in an output function with OR-gate
c) Generation of selected path between multiple sources and a single destination
d) Encoding of binary information

Answer: c
Explanation: The major functioning responsibility of the multiplexing combinational circuit is generation of selected path between multiple sources and a single destination because it makes the circuit too flexible. 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, depending on the active select lines.

107. What is the function of an enable input on a multiplexer chip?
a) To apply Vcc
b) To connect ground

DIWAKAR EDUCATION HUB

c) To active the entire chip
d) To active one half of the chip

Answer: c
Explanation: Enable input is used to active the chip, when enable is high the chip works (ACTIVE), when enable is low the chip does not work (MEMORY). However, Enable can be Active-High or Active-Low, indicating it is active either when it is connected to VCC or GND respectively.

108. The language made up of binary coded instructions.
a) Machine
b) C
c) BASIC
d) High level

Answer: a
Explanation: The language made up of binary coded instructions built into the hardware of a particular computer and used directly by the computer is machine language.

109. Binary code comprises of digits from 0 to 9.
a) True
b) False

Answer: b
Explanation: The statement is false. Binary as the word suggests contains only 2 digits : 0 and 1.
0 denotes false and 1 denotes a truth value.

110. The _____ contains the address of the next instruction to be executed.
a) IR
b) PC
c) Accumulator
d) System counter

Answer: b
Explanation: PC stands for program counter (It contains the address of the next instruction to be executed).

111. The memory unit is made up of _____ bytes.
a) 256
b) 124
c) 4096
d) 3096

Answer: c
Explanation: The memory unit is made up of 4,096 bytes. Memory unit is responsible for the storage of data. It is an important entity in the computer system.

112. A document that specifies how many times and with what data the

DIWAKAR EDUCATION HUB

program must be run in order to thoroughly test it.
a) addressing plan
b) test plan
c) validation plan
d) verification plan

Answer: b
Explanation: Test plan is the A document that specifies how many times and with what data the program must be run in order to thoroughly test it. It comes under testing.

113. An approach that designs test cases by looking at the allowable data values.
a) Maintenance
b) Evaluation
c) Data coverage
d) Validation

Answer: c
Explanation: Data coverage is the term used. It is responsible for designing the test cases.

114. The formal grammar rules governing the construction of valid instruction.
a) test case
b) syntax
c) program
d) semantics

Answer: b
Explanation: Syntax determines the grammatical rules in a code. Semantics give meaning to the instructions.

115. A program that reads each of the instructions in mnemonic form and translates it into the machine-language equivalent.
a) Machine language
b) Assembler
c) Interpreter
d) C program

Answer: b
Explanation: Assembler does this job. A language that uses mnemonic codes for the representation of machine-language instructions is called assembly language.

116. An approach that designs test cases by looking at the allowable data values.
a) Data coverage
b) Code Coverage
c) Debugging
d) Validation

DIWAKAR EDUCATION HUB

Answer: a
Explanation: Data coverage is an approach that designs test cases by looking at the allowable data values. Code coverage is an approach that designs test cases by looking at the code.

117. The rules that give meaning to the instructions.
a) Semantics
b) Syntax
c) Code
d) Cases

Answer: a
Explanation: The answer is semantics. They are the rules that give meaning to the instructions. The syntax is the formal rules that ensure validation of code.

118. One multiplexer can take the place of _____
a) Several SSI logic gates
b) Combinational logic circuits
c) Several Ex-NOR gates
d) Several SSI logic gates or combinational logic circuits

Answer: d
Explanation: 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, depending on the active select lines. Since many operational behaviour can be performed by using a multiplexer. Whereas, a combinational circuit is a combination of many logic gates which makes the circuit more complex.

119. A digital multiplexer is a combinational circuit that selects _____
a) One digital information from several sources and transmits the selected one
b) Many digital information and convert them into one
c) Many decimal inputs and transmits the selected information
d) Many decimal outputs and accepts the selected information

Answer: a
Explanation: A digital multiplexer is a combinational circuit that selects one digital information from several sources and transmits the selected information on a single output line depending on the status of the select lines. That is why it is also known as a data selector.

DIWAKAR EDUCATION HUB

120. In a multiplexer, the selection of a particular input line is controlled by _____

a) Data controller
b) Selected lines
c) Logic gates
d) Both data controller and selected lines

Answer: b
Explanation: The selection of a particular input line is controlled by a set of selected lines in a multiplexer, which helps to select a particular input from several sources.

121. If the number of n selected input lines is equal to 2^m then it requires _____ select lines.

a) 2
b) m
c) n
d) 2^n

Answer: b
Explanation: If the number of n selected input lines is equal to 2^m then it requires m select lines to select one of m select lines.

122. How many select lines would be required for an 8-line-to-1-line multiplexer?

a) 2
b) 4
c) 8
d) 3

Answer: d
Explanation: 2^n input lines, n control lines and 1 output line available for MUX. Here, 8 input lines mean 2^3 inputs. So, 3 control lines are possible. Depending on the status of the select lines, the input is selected and fed to the output.

123. A basic multiplexer principle can be demonstrated through the use of a _____

a) Single-pole relay
b) DPDT switch
c) Rotary switch
d) Linear stepper

Answer: c
Explanation: A basic multiplexer principle can be demonstrated through the use of a rotary switch. Since its behaviour is similar to the multiplexer. There are around 10 digits out of which one is selected one at a time and fed to the output.

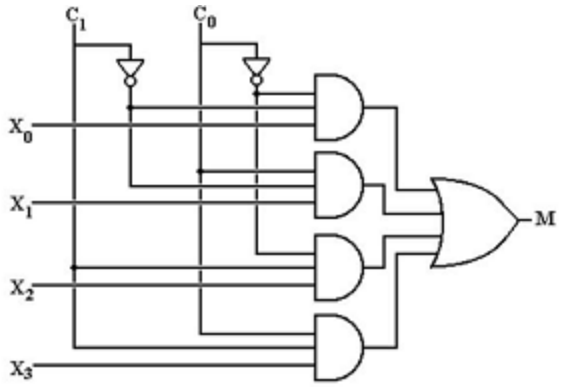
124. How many NOT gates are required for the construction of a 4-to-1 multiplexer?

a) 3

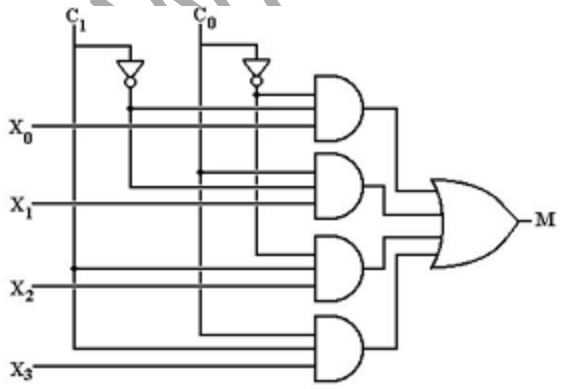
DIWAKAR EDUCATION HUB

b) 4
c) 2
d) 5

Answer: c
Explanation: There are two NOT gates required for the construction of 4-to-1 multiplexer. x0, x1, x2 and x3 are the inputs and C1 and C0 are the select lines and M is the output.
The diagram of a 4-to-1 multiplexer is shown
below:



125. In the given 4-to-1 multiplexer, if c1 = 0 and c0 = 1 then the output M is _____



a) X0
b) X1
c) X2
d) X3

Answer: b
Explanation: The output will be X1, because c1 = 0 and c0 = 1 results into 1 which further results as X1. And rest of the AND gates gives output as 0.

126. The enable input is also known as _____
a) Select input
b) Decoded input
c) Strobe
d) Sink

Answer: c
Explanation: The enable input is also known as strobe which is used to cascade two or more multiplexer ICs to construct a multiplexer with a larger number of inputs. Enable input activates the multiplexer to operate.

127. Execution of several activities at the same time.
a) processing
b) parallel processing
c) serial processing
d) multitasking

Answer: b
Explanation: Execution of several activities at the same time is referred to as parallel processing. Like, Two multiplications at the same time on 2 different processes.

128. Parallel processing has single execution flow.
a) True
b) False

Answer: b
Explanation: The statement is false. Sequential programming specifically has single execution flow.

129. A term for simultaneous access to a resource, physical or logical.
a) Multiprogramming
b) Multitasking
c) Threads
d) Concurrency

Answer: d
Explanation: Concurrency is the term used for the same. When several things are accessed simultaneously, the job is said to be concurrent.

130. _____ leads to concurrency.
a) Serialization
b) Parallelism
c) Serial processing
d) Distribution

Answer: b
Explanation: Parallelism leads naturally to Concurrency. For example, Several processes trying to print a file on a single printer.

131. A parallelism based on increasing processor word size.
a) Increasing
b) Count based
c) Bit based
d) Bit level

Answer: d
Explanation: Bit level parallelism is based on increasing processor word size. It focuses on hardware capabilities for structuring.

132. A type of parallelism that uses micro architectural techniques.
a) instructional
b) bit level
c) bit based
d) increasing

Answer: a
Explanation: Instructional level uses micro architectural techniques. It focuses on program instructions for structuring.

DIWAKAR EDUCATION HUB

133. MIPS stands for?
a) Mandatory Instructions/sec
b) Millions of Instructions/sec
c) Most of Instructions/sec
d) Many Instructions / sec

Answer: b
Explanation: MIPS stands for Millions of Instructions/sec. MIPS is a way to measure the cost of computing.

134. The measure of the "effort" needed to maintain efficiency while adding processors.
a) Maintainablity
b) Efficiency
c) Scalabilty
d) Effectiveness

Answer: c
Explanation: The measure of the "effort" needed to maintain efficiency while adding processors is called as scalabilty.

135. The rate at which the problem size need to be increased to maintain efficiency.
a) Isoeffciency
b) Efficiency
c) Scalabilty
d) Effectiveness

Answer: a
Explanation: Isoefficiency is the rate at which the problem size need to be increased to maintain efficiency.

136. Several instructions execution simultaneously in _____
a) processing
b) parallel processing
c) serial processing
d) multitasking

Answer: b
Explanation: In parallel processing, the several instructions are executed simultaneously.

137. How many inputs will a decimal-to-BCD encoder have?
a) 4
b) 8
c) 10
d) 16

Answer: c
Explanation: An encoder is a combinational circuit encoding the information of 2^n input lines to n output lines, thus producing the binary equivalent of the input. Thus, a Decimal-to-bcd converter has decimal values as inputs which range from 0-9. So, total 10 inputs are there in a decimal-to-BCD encoder.

DIWAKAR EDUCATION HUB

138. How many outputs will a decimal-to-BCD encoder have?
a) 4
b) 8
c) 12
d) 16

Answer: a
Explanation: An encoder is a combinational circuit encoding the information of 2^n input lines to n output lines, thus producing the binary equivalent of the input. Thus, a decimal to BCD encoder has 4 outputs.

139. How is an encoder different from a decoder?
a) The output of an encoder is a binary code for 1-of-N input
b) The output of a decoder is a binary code for 1-of-N input
c) The output of an encoder is a binary code for N-of-1 output
d) The output of a decoder is a binary code for N-of-1 output

Answer: a
Explanation: An encoder is a combinational circuit encoding the information of 2^n input lines to n output lines, thus producing the binary equivalent of the input. It performs the opposite operation of a decoder which results in 2^n outputs from n inputs. Thus, an encoder different from a decoder because of the output of an encoder is a binary code for 1-of-N input.

140. The full form of ROM is _____
a) Read Outside Memory
b) Read Out Memory
c) Read Only Memory
d) Read One Memory

Answer: c
Explanation: The full form of ROM is Read Only Memory.

141. ROM consist of _____
a) NOR and OR arrays
b) NAND and NOR arrays
c) NAND and OR arrays
d) NOR and AND arrays

Answer: c
Explanation: ROM consists of NAND and OR arrays which can be programmed by the user to implement combinational & sequential functions. Combinational Operations like that of adders and subtractors and Sequential Functions like that of storing in the memory.

DIWAKAR EDUCATION HUB

142. For reprogrammability, PLDs use _____

a) PROM
b) EPROM
c) CDROM
d) PLA

Answer: b
Explanation: For reprogrammability, PLDs use EPROM (i.e. Erasable PROM). It erases the previous program and starts uploading a new one. However, data is erased by exposing it to UV-light, which is a tedious and time-consuming process.

143. The full form of PROM is _____

a) Previous Read Only Memory
b) Programmable Read Out Memory
c) Programmable Read Only Memory
d) Previous Read Out Memory

Answer: c
Explanation: The full form of PROM is Programmable Read Only Memory, where the ROM can be programmed by the user.

144. The full form of EPROM is _____

a) Easy Programmable Read Only Memory
b) Erasable Programmable Read Only Memory
c) Eradicate Programmable Read Only Memory
d) Easy Programmable Read Out Memory

Answer: b
Explanation: The full form of EPROM is Erasable Programmable Read Only Memory, where the ROM can be erased and re-used by the user.

145. PLDs with programmable AND and fixed OR arrays are called _____

a) PAL
b) PLA
c) APL
d) PPL

Answer: a
Explanation: PLDs with programmable AND and fixed OR arrays are called PAL (i.e. Programmable Array Logic). However, PAL is less flexible but has higher speed.

146. When both the AND and OR are programmable, such PLDs are known as _____

a) PAL
b) PPL

DIWAKAR EDUCATION HUB

c) PLA
d) APL

Answer: c
Explanation: When both the AND and OR are programmable, such PLDs are known as PLA (i.e. Programmable Logic Array). However, PLA is more flexible but has less speed.

147. ASIC stands for _____
a) Application Special Integrated Circuits
b) Applied Special Integrated Circuits
c) Application Specific Integrated Circuits
d) Applied Specific Integrated Circuits
Answer: c
Explanation: In digital electronics, ASIC stands for Application Specific Integrated Circuits. It is a customized integrated circuit which is produced for a specific use and not for a common-purpose.

148. The programmability and high density of PLDs make them useful in the design of _____
a) ISAC
b) ASIC
c) SACC
d) CISF

Answer: b
Explanation: The programmability and high density of PLDs make them useful in the design of ASIC (i.e. Application Specific Integrated Circuits) where design changes can be more rapidly and inexpensively.

149. FPGA stands for _____
a) Full Programmable Gate Array
b) Full Programmable Genuine Array
c) First Programmable Gate Array
d) Field Programmable Gate Array

Answer: d
Explanation: In digital electronics, FPGA stands for Field Programmable Gate Array. This type of integrated circuit is for general-purpose which is configured by the user as per their requirement.

150. Which of the following is a reprogrammable gate array?
a) EPROM
b) FPGA
c) Both EPROM and FPGA
d) ROM

Answer: c
Explanation: Both FPGA and EPROM are reprogrammable gate array.

DIWAKAR EDUCATION HUB

151. The difference between FPGA and PLD is that _____
a) FPGA is slower than PLD
b) FPGA has high power dissipation
c) FPGA incorporates logic blocks
d) All of the Mentioned

Answer: c
Explanation: The differences between FPGA and PLD is that FPGA incorporates logic blocks instead of fixed AND-OR gates and is faster with low power dissipation. FPGAs are designed for having higher gate count whereas, PLDs are used for lesser gate counts.

152. If we record any music in any recorder, such types of process is called _____
a) Multiplexing
b) Encoding
c) Decoding
d) Demultiplexing

Answer: b
Explanation: If we record any music in any recorder, it means that we are giving data to a recorder. So, such process is called encoding. Getting back the music from the recorded data, is known as decoding.

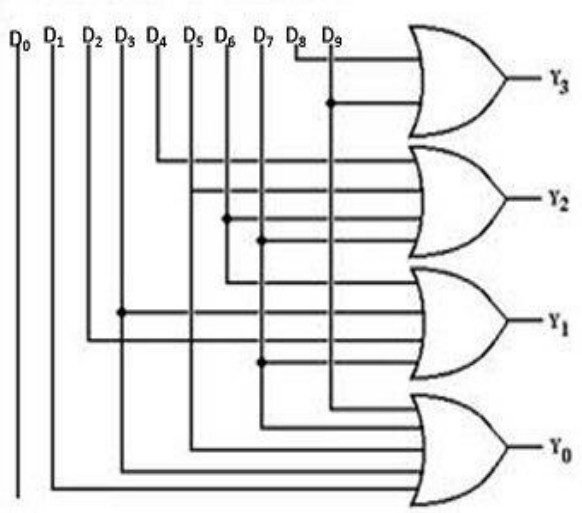
153. Can an encoder be a transducer?
a) Yes
b) No
c) May or may not be
d) Both are not even related slightly

Answer: a
Explanation: Of course, a transducer is a device which has the capability to emit data as well as to accept. Transducer converts signal from one form of energy to another.

154. How many OR gates are required for a Decimal-to-bcd encoder?
a) 2
b) 10
c) 3
d) 4

Answer: d
Explanation: An encoder is a combinational circuit encoding the information of 2^n input lines to n output lines, thus producing the binary equivalent of the input.
This is clear from the diagram that it requires 4 OR gates:

DIWAKAR EDUCATION HUB



.

155. Separation of user logical memory and physical memory is _____

a) Memory control
b) Memory management
c) Memory sharing
d) Virtual memory

Answer: d
Explanation: The separation of user logical memory and physical memory is called virtual memory. Only part of the program needs to be in memory for execution.

156. Logical Address space can be larger than physical address space.
a) True
b) False

Answer: a
Explanation: The statement is true. Since, a part of the program needs to be in memory for the process of execution, the logical space can therefore be much larger than the physical address space.

157. Virtual Memory can be implemented via _____
a) Demand Paging
b) Logical paging
c) Structural way
d) Simple division

Answer: a
Explanation: Demand paging can implement virtual memory. Another way is demand segmentation.

158. COW stands for?
a) Copy over write
b) Convert over write
c) Count over write
d) Copy over write

Answer: d
Explanation: COW stands for Copy over write. COW allows both parent and child processes to share the same pages initially.

159. LRU stands for?
a) Least Recently used
b) Less Recently used

DIWAKAR EDUCATION HUB

c) Least Recurrently used
d) Least Randomly used

Answer: a
Explanation: LRU stands for Least Recently used. LRU is least recently used. It replaces page with the smallest count.

160. An allocation that uses a proportional allocation scheme using priorities rather than size.
a) Priority allocation
b) File allocation
c) Preference allocation
d) Simple allocation

Answer: a
Explanation: Priority allocation uses a proportional allocation scheme using priorities rather than size.

161. A process selects a replacement frame from the set of all frames.
a) Local replacement
b) Global replacement
c) Block replacement
d) Module replacement

Answer: b
Explanation: Global replacement process selects a replacement frame from the set of all frames; one process can take a frame from another.

162. How many OR gates are required for an octal-to-binary encoder?
a) 3
b) 2
c) 8
d) 10

Answer: a
Explanation: An encoder is a combinational circuit encoding the information of 2^n input lines to n output lines, thus producing the binary equivalent of the input. Thus, in octal to binary encoder there are 8 ($=2^3$) inputs, thus 3 output lines.

163. For 8-bit input encoder how many combinations are possible?
a) 8
b) 2^8
c) 4
d) 2^4

Answer: b
Explanation: An encoder is a combinational circuit encoding the information of 2^n input lines to n output lines, thus producing the binary equivalent of the input. There are 2^8 combinations are possible for

DIWAKAR EDUCATION HUB

an 8-bit input encoder but out of which only 8 are used using 3 output lines. It is a disadvantage of encoder.

164. The discrepancy of 0 output due to all inputs being 0 or D0, being 0 is resolved by using additional input known as _____
a) Enable
b) Disable
c) Strobe
d) Clock

Answer: a
Explanation: Such problems are resolved by using enable input, which behaves as active if it gets 0 as input since it is an active-low pin.

165. Can an encoder be called as multiplexer?
a) No
b) Yes
c) Sometimes
d) Never

Answer: b
Explanation: A multiplexer or MUX is a combination circuit that contains more than one input line, one output line and more than one selection line. Whereas, an encoder is also considered a type of multiplexer but without a single output line and without any selection lines.

166. If two inputs are active on a priority encoder, which will be coded on the output?
a) The higher value
b) The lower value
c) Neither of the inputs
d) Both of the inputs

Answer: a
Explanation: An encoder is a combinational circuit encoding the information of 2^n input lines to n output lines, thus producing the binary equivalent of the input. If two inputs are active on a priority encoder, the input of higher value will be coded in the output.

167. How many outputs are present in a BCD decoder?
a) 4
b) 5
c) 15
d) 10

Answer: d
Explanation: A binary decoder is a combinational logic circuit which decodes binary information from n-inputs to a maximum of 2^n outputs. A BCD to Decimal decoder has 10

DIWAKAR EDUCATION HUB

number of outputs because the decimal digit's range is from 0 to 9.

168. Which digital system translates coded characters into a more useful form?
a) Encoder
b) Display
c) Counter
d) Decoder

Answer: d
Explanation: A binary decoder is a combinational logic circuit which decodes binary information from n-inputs to a maximum of 2^n outputs. Decoder converts the coded characters into our required data form.

169. What control signals may be necessary to operate a 1-line-to-16 line decoder?
a) Flasher circuit control signal
b) A LOW on all gate enable inputs
c) Input from a hexadecimal counter
d) A HIGH on all gate enable circuits

Answer: b
Explanation: A LOW on all gate enable inputs is necessary to operate a 1-line-to-16 line decoder because enable pins are usually, active-low pins.

170. How many inputs are required for a 1-of-10 BCD decoder?
a) 4
b) 8
c) 10
d) 2

Answer: a
Explanation: A binary decoder is a combinational logic circuit which decodes binary information from n-inputs to a maximum of 2^n outputs. Therefore, for a BCD to decimal decoder, No. of inputs = 4 such that number of outputs is <= 2^n.

171. A BCD decoder will have how many rows in its truth table?
a) 10
b) 9
c) 8
d) 3

Answer: a
Explanation: A binary decoder is a combinational logic circuit which decodes binary information from n-inputs to a maximum of 2^n outputs. Thus, BCD decoder will have 10 rows as it's input ranges from 0 to 9.

172. How many possible outputs would a decoder have with a 6-bit binary input?

DIWAKAR EDUCATION HUB

a) 32
b) 64
c) 128
d) 16

Answer: c
Explanation: The possible outputs would be: 2^n = 64 (Since n = 6 here).

173. One way to convert BCD to binary using the hardware approach is:
a) By using MSI IC circuits
b) By using a keyboard encoder
c) By using an ALU
d) By using UART

Answer: a
Explanation: One way to convert BCD to binary using the hardware approach is MSI (medium scale integration) IC circuits.

174. How many inputs are required for a 1-of-16 decoder?
a) 2
b) 16
c) 8
d) 4

Answer: d
Explanation: A binary decoder is a combinational logic circuit which decodes binary information from n-inputs to a maximum of 2^n outputs. Here, number of outputs = 16.
$16 = 2^4 = 2^n$. Thus, number of inputs is 4.

175. A truth table with output columns numbered 0–15 may be for which type of decoder IC?
a) Hexadecimal 1-of-16
b) Dual octal outputs
c) Binary-to-hexadecimal
d) Hexadecimal-to-binary

Answer: a
Explanation: A binary decoder is a combinational logic circuit which decodes binary information from n-inputs to a maximum of 2^n outputs. A truth table with output columns numbered 0–15 may be for Hexadecimal 1-of-16. Because, hexadecimal occupies less space in a system.

176. How can the active condition (HIGH or LOW) or the decoder output be determined from the logic symbol?
a) A bubble indicates active-HIGH
b) A bubble indicates active-LOW
c) A triangle indicates active-HIGH
d) A triangle indicates active-LOW

DIWAKAR EDUCATION HUB

Answer: b
Explanation: A bubble indicates active-LOW in a decoder always. Enable pin of the decoder is usually active-LOW and is triggered on input being at 0.

177. A code converter is a logic circuit that _____
a) Inverts the given input
b) Converts into decimal number
c) Converts data of one type into another type
d) Converts to octal

Answer: c
Explanation: A code converter is a logic circuit that changes data presented in one type of binary code to another type of binary code.

178. Use the weighting factors to convert the following BCD numbers to binary _____

0101 0011 & 0010 0110 1000

a) 01010011 001001101000
b) 11010100 100001100000
c) 110101 100001100
d) 101011 001100001

Answer: c
Explanation: Firstly, convert every 4 sets of binary to decimal from the given: 0101=5, 0011=3. Then convert 53 to binary, which will give 110101. Again, do the same with the next 4 set of binary digits.

179. The primary use for Gray code is _____
a) Coded representation of a shaft's mechanical position
b) Turning on/off software switches
c) To represent the correct ASCII code to indicate the angular position of a shaft on rotating machinery
d) To convert the angular position of a shaft on rotating machinery into hexadecimal code

Answer: a
Explanation: Gray code is useful because only one bit changes at a time, which is implemented easily in Coded representation of a shaft's mechanical position. In Gray Code, every sequence of successive bits differs by 1 bit only.

180. Code is a symbolic representation of _____
a) Discrete information
b) Continuous information
c) Decimal information into binary
d) Binary information into decimal

DIWAKAR EDUCATION HUB

Answer: a
Explanation: Code is a symbolic representation of discrete information. Codes can be anything like numbers, letter or words, written in terms of group of symbols.

181. One way to convert BCD to binary using the hardware approach is _____
a) With MSI IC circuits
b) With a keyboard encoder
c) With an ALU
d) UART

Answer: a
Explanation: One way to convert BCD to binary using the hardware approach is MSI IC (i.e. medium scale integration) circuits.

182. Why is the Gray code more practical to use when coding the position of a rotating shaft?
a) All digits change between counts
b) Two digits change between counts
c) Only one digit changes between counts
d) Alternate digit changes between counts

Answer: c
Explanation: The Gray code is more practical to use when coding the position of a rotating shaft because only one digit changes between counts that is reflected to the next count.

183. Reflected binary code is also known as _____
a) BCD code
b) Binary code
c) ASCII code
d) Gray Code

Answer: d
Explanation: The reflected binary code is also known as gray code because one digit reflected to the next bit. In Gray Code, every sequence of successive bits differs by 1 bit only.

184. Why do we use gray codes?
a) To count the no of bits changes
b) To rotate a shaft
c) Error correction
d) Error Detetction

Answer: c
Explanation: Today, Gray codes are widely used to facilitate error correction in digital communications such as digital terrestrial television and some cable TV systems.

DIWAKAR EDUCATION HUB

185. Earlier, reflected binary codes were applied to _____
a) Binary addition
b) 2's complement
c) Mathematical puzzles
d) Binary multiplication

Answer: c
Explanation: The reflected binary code is also known as gray code because one digit reflected to the next bit. In Gray Code, every sequence of successive bits differs by 1 bit only. Reflected binary codes were applied to mathematical puzzles before they became known to engineers.

186. The binary representation of BCD number 00101001 (decimal 29) is _____
a) 0011101
b) 0110101
c) 1101001
d) 0101011

Answer: a
Explanation: The given BCD number 00101001 has three 1s. So, it can be rewritten as 0000001-1, 0001000-8, 0010100-20 and after addition, we get 0011101 as output.

187. Convert binary number into gray code: 100101.
a) 101101
b) 001110
c) 110111
d) 111001

Answer: c
Explanation: : Conversion from Binary To Gray Code:

1 (XOR) 0 (XOR) 0 (XOR) 1 (XOR) 0 (XOR) 1

↓ ↓ ↓ ↓ ↓

1 1 0 1 1 1

188. Each personal computer has a _____ that manages the computer's arithmetical, logical and control activities.
a) Microprocessor
b) Assembler
c) Microcontroller
d) Interpreter

Answer: a
Explanation: Microprocessor handles all these activities. Each family of processors has its own set of instructions for handling various operations like getting input from keyboard, displaying information on a

DIWAKAR EDUCATION HUB

screen and performing various other jobs.

189. Assembly Language requires less memory and execution time.
a) True
b) False

Answer: a
Explanation: The statement is true. Advantages of using assembly language are:
• It requires less memory and execution time.
• It allows hardware-specific complex jobs in an easier way.
• It is suitable for time-critical jobs.

190. The data size of a word is _____
a) 2-byte
b) 4-byte
c) 8-byte
d)16-byte

Answer: a
Explanation: The processor supports the following data sizes:
• Word: a 2-byte data item
• Double word: a 4-byte (32 bit) data item, etc.

191. A direct reference of specific location.
a) Segment Address
b) Absolute Address
c) Offset
d) Memory Address

Answer: b
Explanation: There are two kinds of memory addresses:
• An absolute address – a direct reference of specific location.
• The segment address (or offset) – starting address of a memory segment with the offset value.

192. A Borland Turbo Assembler.
a) nasm
b) tasm
c) gas
d) asm

Answer: b
Explanation: Tasm is the borland turbo assembler. Nasm is used with linux generally. Gas is the GNU assembler.

193. The instructions that tell the assembler what to do.
a) Executable instructions
b) Pseudo-ops
c) Logical instructions
d) Macros

DIWAKAR EDUCATION HUB

Answer: a
Explanation: The executable instructions or simple instructions tell the processor what to do. Each instruction consists of an operation code (opcode). Each executable instruction generates one machine language instruction.

194. The segment containing data values passed to functions and procedures within the program.
a) Code
b) Data
c) Stack
d) System

Answer: c
Explanation: The stack segment contains data values passed to functions and procedures within the program. The code segment defines an area in memory that stores the instruction codes.

195. To speed up the processor operations, the processor includes some internal memory storage locations, called _____
a) Drives
b) Memory
c) Units
d) Registers

Answer: d
Explanation: The processor has some internal memory storage locations, known as registers. The registers stores data elements for processing without having to access memory.

196. To locate the exact location of data in memory, we need the starting address of the segment, which is found in the DS register and an offset value. This offset value is also called?
a) Effective Address
b) Direct offset address
c) Memory address
d) General Address

Answer: a
Explanation: When operands are specified in memory addressing mode, direct access to main memory, usually to the data segment, is required. This way of addressing results in slower processing of data. To get the exact location of data in memory, we need segment start address, which is found in the DS register and an offset value. This offset value is called an effective address.

197. Each byte of character is stored as its ASCII value in _____

DIWAKAR EDUCATION HUB

a) Hexadecimal
b) Binary
c) Octal
d) Decimal

Answer: a
Explanation: Assembly language deals with hexadecimal values only. Each decimal value is automatically converted to its 16-bit binary equivalent and stored as a hexadecimal number.

198. A latch is an example of a _____
a) Monostable multivibrator
b) Astable multivibrator
c) Bistable multivibrator
d) 555 timer

Answer: c
Explanation: A latch is an example of a bistable multivibrator. A Bistable multivibrator is one in which the circuit is stable in either of two states. It can be flipped from one state to the other state and vice-versa.

199. Latch is a device with _____
a) One stable state
b) Two stable state
c) Three stable state
d) Infinite stable states

Answer: b
Explanation: Since, a latch works on the principal of bistable multivibrator. A Bistable multivibrator is one in which the circuit is stable in either of two states. It can be flipped from one state to the other state and vice-versa. So a latch has two stable states.

200. Why latches are called a memory devices?
a) It has capability to stare 8 bits of data
b) It has internal memory of 4 bit
c) It can store one bit of data
d) It can store infinite amount of data

Answer: c
Explanation: Latches can be memory devices, and can store one bit of data for as long as the device is powered. Once device is turned off, the memory gets refreshed.

201. Two stable states of latches are _____
a) Astable & Monostable
b) Low input & high output

DIWAKAR EDUCATION HUB

c) High output & low output
d) Low output & high input

Answer: c
Explanation: A latch has two stable states, following the principle of Bistable Multivibrator. There are two stable states of latches and these states are high-output and low-output.

202. How many types of latches are

_____

a) 4
b) 3
c) 2
d) 5

Answer: a
Explanation: There are four types of latches: SR latch, D latch, JK latch and T latch. D latch is a modified form of SR latch whereas, T latch is an advanced form of JK latch.

203. The full form of SR is

_____

a) System rated
b) Set reset
c) Set ready
d) Set Rated

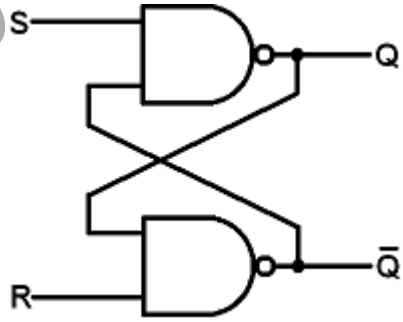
Answer: b
Explanation: The full form of SR is set/reset. It is a type of latch having two stable states.

204. The SR latch consists of

_____

a) 1 input
b) 2 inputs
c) 3 inputs
d) 4 inputs

Answer: b
Explanation: SR or Set-Reset latch is the simplest type of bistable multivibrator having two stable states.
The diagram of SR latch is shown below:



205. The outputs of SR latch are

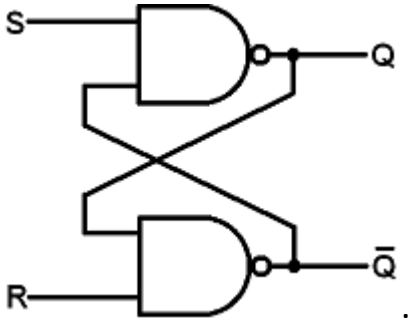
_____

a) x and y
b) a and b
c) s and r
d) q and q’

Answer: d
Explanation: SR or Set-Reset latch is

DIWAKAR EDUCATION HUB

the simplest type of bistable multivibrator having two stable states. The inputs of SR latch are s and r while outputs are q and q'. It is clear from the diagram:



206. The NAND latch works when both inputs are _____

a) 1
b) 0
c) Inverted
d) Don't cares

Answer: a
Explanation: The NAND latch works when both inputs are 1. Since, both of the inputs are inverted in a NAND latch.

207. The first step of analysis procedure of SR latch is to _____

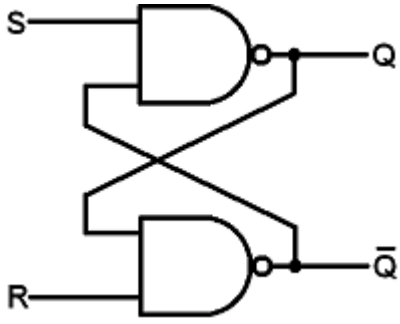
a) label inputs
b) label outputs
c) label states
d) label tables

Answer: b
Explanation: All flip flops have at least one output labeled Q (i.e. inverted). This is so because the flip flops have inverting gates inside them, hence in order to have both Q and Q complement available, we have atleast one output labelled.

208. The inputs of SR latch are _____

a) x and y
b) a and b
c) s and r
d) j and k

Answer: c
Explanation: SR or Set-Reset latch is the simplest type of bistable multivibrator having two stable states. The inputs of SR latch are s and r while outputs are q and q'. It is clear from the diagram:



209. When a high is applied to the Set line of an SR latch, then _____

DIWAKAR EDUCATION HUB

a) Q output goes high
b) Q' output goes high
c) Q output goes low
d) Both Q and Q' go high

Answer: a
Explanation: S input of a SR latch is directly connected to the output Q. So, when a high is applied Q output goes high and Q' low.

210. When both inputs of SR latches are low, the latch _____
a) Q output goes high
b) Q' output goes high
c) It remains in its previously set or reset state
d) it goes to its next set or reset state

Answer: c
Explanation: When both inputs of SR latches are low, the latch remains in it's present state. There is no change in the output.

211. When both inputs of SR latches are high, the latch goes _____
a) Unstable
b) Stable
c) Metastable
d) Bistable

Answer: c
Explanation: When both gates are identical and this is "metastable", and the device will be in an undefined state for an indefinite period.

212. Latches constructed with NOR and NAND gates tend to remain in the latched condition due to which configuration feature?
a) Low input voltages
b) Synchronous operation
c) Gate impedance
d) Cross coupling

Answer: d
Explanation: Latch is a type of bistable multivibrator having two stable states. Both inputs of a latch are directly connected to the other's output. Such types of structure is called cross coupling and due to which latches remain in the latched condition.

213. One example of the use of an S-R flip-flop is as _____
a) Transition pulse generator
b) Racer
c) Switch debouncer
d) Astable oscillator

Answer: c
Explanation: The SR flip-flop is very effective in removing the effects of switch bounce, which is the

DIWAKAR EDUCATION HUB

unwanted noise caused during the switching of electronic devices.

214. The truth table for an S-R flip-flop has how many VALID entries?
a) 1
b) 2
c) 3
d) 4

Answer: c
Explanation: The SR flip-flop actually has three inputs, Set, Reset and its current state. The Invalid or Undefined State occurs at both S and R being at 1.

215. When both inputs of a J-K flip-flop cycle, the output will _____
a) Be invalid
b) Change
c) Not change
d) Toggle

Answer: c
Explanation: After one cycle the value of each input comes to the same value. Eg: Assume J=0 and K=1. After 1 cycle, it becomes as J=0->1->0(1 cycle complete) and K=1->0->1(1 cycle complete). The J & K flip-flop has 4 stable states: Latch, Reset, Set and Toggle.

216. Which of the following is correct for a gated D-type flip-flop?
a) The Q output is either SET or RESET as soon as the D input goes HIGH or LOW
b) The output complement follows the input when enabled
c) Only one of the inputs can be HIGH at a time
d) The output toggles if one of the inputs is held HIGH

Answer: a
Explanation: In D flip flop, when the clock is high then the output depends on the input otherwise reminds previous output. In a state of clock high, when D is high the output Q also high, if D is '0' then output is also zero. Like SR flip-flop, the D-flip-flop also have an invalid state at both inputs being 1.

217. A basic S-R flip-flop can be constructed by cross-coupling of which basic logic gates?
a) AND or OR gates
b) XOR or XNOR gates
c) NOR or NAND gates
d) AND or NOR gates

Answer: c
Explanation: The basic S-R flip-flop

DIWAKAR EDUCATION HUB

can be constructed by cross coupling of NOR or NAND gates. Cross coupling means the output of second gate is fed to the input of first gate and vice-versa.

218. The logic circuits whose outputs at any instant of time depends only on the present input but also on the past outputs are called
a) Combinational circuits
b) Sequential circuits
c) Latches
d) Flip-flops

Answer: b
Explanation: In sequential circuits, the output signals are fed back to the input side. So, The circuits whose outputs at any instant of time depends only on the present input but also on the past outputs are called sequential circuits. Unlike sequential circuits, if output depends only on the present state, then it's known as combinational circuits.

219. Whose operations are more faster among the following?
a) Combinational circuits
b) Sequential circuits
c) Latches
d) Flip-flops

Answer: a
Explanation: Combinational circuits are often faster than sequential circuits. Since, the combinational circuits do not require memory elements whereas the sequential circuits need memory devices to perform their operations in sequence. Latches and Flip-flops come under sequential circuits.

220. How many types of sequential circuits are?
a) 2
b) 3
c) 4
d) 5

Answer: a
Explanation: There are two type of sequential circuits viz., (i) synchronous or clocked and (ii) asynchronous or unclocked. Synchronous Sequential Circuits are triggered in the presence of a clock signal, whereas, Asynchronous Sequential Circuits function in the absence of a clock signal.

221. The sequential circuit is also called _____
a) Flip-flop
b) Latch

DIWAKAR EDUCATION HUB

c) Strobe
d) Adder

Answer: b
Explanation: The sequential circuit is also called a latch because both are a memory cell, which are capable of storing one bit of information.

222. The basic latch consists of _____
a) Two inverters
b) Two comparators
c) Two amplifiers
d) Two adders

Answer: a
Explanation: The basic latch consists of two inverters. It is in the sense that if the output Q = 0 then the second output Q' = 1 and vice versa.

223. In S-R flip-flop, if Q = 0 the output is said to be _____
a) Set
b) Reset
c) Previous state
d) Current state

Answer: b
Explanation: In S-R flip-flop, if Q = 0 the output is said to be reset and set for Q = 1.

224. What is a trigger pulse?
a) A pulse that starts a cycle of operation
b) A pulse that reverses the cycle of operation
c) A pulse that prevents a cycle of operation
d) A pulse that enhances a cycle of operation

Answer: a
Explanation: Trigger pulse is defined as a pulse that starts a cycle of operation.

225. The circuits of NOR based S-R latch classified as asynchronous sequential circuits, why?
a) Because of inverted outputs
b) Because of triggering functionality
c) Because of cross-coupled connection
d) Because of inverted outputs & triggering functionality

Answer: c
Explanation: The cross-coupled connections from the output of one gate to the input of the other gate constitute a feedback path. For this reason, the circuits of NOR based S-R latch classified as asynchronous sequential circuits. Moreover, they

DIWAKAR EDUCATION HUB

are referred to as asynchronous because they function in the absence of a clock pulse.

226. In digital logic, a counter is a device which _____
a) Counts the number of outputs
b) Stores the number of times a particular event or process has occurred
c) Stores the number of times a clock pulse rises and falls
d) Counts the number of inputs

Answer: b
Explanation: In digital logic and computing, a counter is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal.

227. A counter circuit is usually constructed of _____
a) A number of latches connected in cascade form
b) A number of NAND gates connected in cascade form
c) A number of flip-flops connected in cascade
d) A number of NOR gates connected in cascade form

Answer: c
Explanation: A counter circuit is usually constructed of a number of flip-flops connected in cascade. Preferably, JK Flip-flops are used to construct counters and registers.

228. What is the maximum possible range of bit-count specifically in n-bit binary counter consisting of 'n' number of flip-flops?
a) 0 to 2^n
b) 0 to $2^n + 1$
c) 0 to $2^n - 1$
d) 0 to $2^{n+1/2}$

Answer: c
Explanation: The maximum possible range of bit-count specifically in n-bit binary counter consisting of 'n' number of flip-flops is 0 to 2^n-1. For say, there is a 2-bit counter, then it will count till 2^2-1 = 3. Thus, it will count from 0 to 3.

229. How many types of the counter are there?
a) 2
b) 3
c) 4
d) 5

Answer: b
Explanation: Counters are of 3 types,

DIWAKAR EDUCATION HUB

namely,
(i)asynchronous/synchronous,
(ii)single and multi-mode &
(iii)modulus counter. These further can be subdivided into Ring Counter, Johnson Counter, Cascade Counter, Up/Down Counter and such like.

230. A decimal counter has _____ states.
a) 5
b) 10
c) 15
d) 20

Answer: b
Explanation: Decimal counter is also known as 10 stage counter. So, it has 10 states. It is also known as Decade Counter counting from 0 to 9.

231. Ripple counters are also called _____
a) SSI counters
b) Asynchronous counters
c) Synchronous counters
d) VLSI counters

Answer: b
Explanation: Ripple counters are also called asynchronous counter. In Asynchronous counters, only the first flip-flop is connected to an external clock while the rest of the flip-flops have their preceding flip-flop output as clock to them.

232. Synchronous counter is a type of _____
a) SSI counters
b) LSI counters
c) MSI counters
d) VLSI counters

Answer: c
Explanation: Synchronous Counter is a Medium Scale Integrated (MSI). In Synchronous Counters, the clock pulse is supplied to all the flip-flops simultaneously.

233. Three decade counter would have _____
a) 2 BCD counters
b) 3 BCD counters
c) 4 BCD counters
d) 5 BCD counters

Answer: b
Explanation: Three decade counter has 30 states and a BCD counter has 10 states. So, it would require 3 BCD counters. Thus, a three decade counter will count from 0 to 29.

234. BCD counter is also known as _____
a) Parallel counter

DIWAKAR EDUCATION HUB

b) Decade counter
c) Synchronous counter
d) VLSI counter

Answer: b
Explanation: BCD counter is also known as decade counter because both have the same number of stages and both count from 0 to 9.

235. The parallel outputs of a counter circuit represent the _____
a) Parallel data word
b) Clock frequency
c) Counter modulus
d) Clock count

Answer: d
Explanation: The parallel outputs of a counter circuit represent the clock count. A counter counts the number of times an event takes place in accordance to the clock pulse.

236. A register is defined as _____
a) The group of latches for storing one bit of information
b) The group of latches for storing n-bit of information
c) The group of flip-flops suitable for storing one bit of information
d) The group of flip-flops suitable for storing binary information

Answer: d
Explanation: A register is defined as the group of flip-flops suitable for storing binary information. Each flip-flop is a binary cell capable of storing one bit of information. The data in a register can be transferred from one flip-flop to another.

237. The register is a type of _____
a) Sequential circuit
b) Combinational circuit
c) CPU
d) Latches

Answer: a
Explanation: Register's output depends on the past and present states of the inputs. The device which follows these properties is termed as a sequential circuit. Whereas, combinational circuits only depend on the present values of inputs.

238. How many types of registers are?
a) 2
b) 3
c) 4
d) 5

Answer: c
Explanation: There are 4 types of shift

DIWAKAR EDUCATION HUB

registers, viz., Serial-In/Serial-Out, Serial-In/Parallel-Out, Parallel-In/Serial-Out and Parallel-In/Parallel-Out.

239. The main difference between a register and a counter is

_____

a) A register has no specific sequence of states
b) A counter has no specific sequence of states
c) A register has capability to store one bit of information but counter has n-bit
d) A register counts data

Answer: a
Explanation: The main difference between a register and a counter is that a register has no specific sequence of states except in certain specialised applications.

240. In D register, 'D' stands for

_____

a) Delay
b) Decrement
c) Data
d) Decay

Answer: c
Explanation: D stands for "data" in case of flip-flops and not delay. Registers are made of a group of flip-flops.

241. Registers capable of shifting in one direction is _____

a) Universal shift register
b) Unidirectional shift register
c) Unipolar shift register
d) Unique shift register

Answer: b
Explanation: The register capable of shifting in one direction is unidirectional shift register. The register capable of shifting in both directions is known as a bidirectional shift register.

242. A register that is used to store binary information is called

_____

a) Data register
b) Binary register
c) Shift register
d) D – Register

Answer: b
Explanation: A register that is used to store binary information is called a binary register. A register in which data can be shifted is called shift register.

DIWAKAR EDUCATION HUB

243. A shift register is defined as _____

a) The register capable of shifting information to another register
b) The register capable of shifting information either to the right or to the left
c) The register capable of shifting information to the right only
d) The register capable of shifting information to the left only

Answer: b
Explanation: The register capable of shifting information either to the right or to the left is termed as shift register. A register in which data can be shifted only in one direction is called unidirectional shift register, while if data can shifted in both directions, it is known as a bidirectional shift register.

244. How many methods of shifting of data are available?
a) 2
b) 3
c) 4
d) 5

Answer: a
Explanation: There are two types of shifting of data are available and these are serial shifting & parallel shifting.

245. In serial shifting method, data shifting occurs _____
a) One bit at a time
b) simultaneously
c) Two bit at a time
d) Four bit at a time

Answer: a
Explanation: As the name suggests serial shifting, it means that data shifting will take place one bit at a time for each clock pulse in a serial fashion. While in parallel shifting, shifting will take place with all bits simultaneously for each clock pulse in a parallel fashion.

246. Memory is a/an _____
a) Device to collect data from other computer
b) Block of data to keep data separately
c) Indispensable part of computer
d) Device to connect through all over the world

Answer: c
Explanation: Memory is an indispensable unit of a computer and microprocessor based systems which stores permanent or temporary data.

DIWAKAR EDUCATION HUB

247. The instruction used in a program for executing them is stored in the _____
a) CPU
b) Control Unit
c) Memory
d) Microprocessor

Answer: c
Explanation: All of the program and the instructions are stored in the memory. The processor fetches it as and when required.

248. A flip flop stores _____
a) 10 bit of information
b) 1 bit of information
c) 2 bit of information
d) 3-bit information

Answer: b
Explanation: A flip-flop has capability to store 1 bit of information. It can be used further after erasing previous information.

249. A register is able to hold _____
a) Data
b) Word
c) Nibble
d) Both data and word

Answer: b
Explanation: Register is also a part of memory inside a computer. It stands there to hold a word. A word is a group of 16-bits or 2-bytes.

250. A register file holds _____
a) A large number of word of information
b) A small number of word of information
c) A large number of programs
d) A modest number of words of information

Answer: d
Explanation: A register file is different from a simple register because of capability to hold a modest number of words of information. A word is a group of 16-bits or 2-bytes.

251. The very first computer memory consisted of _____
a) A small display
b) A large memory storage equipment
c) An automatic keyboard input
d) An automatic mouse input

Answer: b
Explanation: The very first computer memory consisted of a minute magnetic toroid, which required

DIWAKAR EDUCATION HUB

large, bulky circuit boards stored in large cabinates.

252. A minute magnetic toroid is also called as _____
a) Large memory
b) Small memory
c) Core memory
d) Both small and large memory

Answer: c
Explanation: A minute magnetic toroid is also called as core memory which is made up of a semiconductor. A semiconductor is a device whose electrical conductivity lies between that of conductor and insulator.

253. Which one of the following has capability to store data in extremely high densities?
a) Register
b) Capacitor
c) Semiconductor
d) Flip-Flop

Answer: c
Explanation: Semiconductor has capability to store data in extremely high densities.

254. A large memory is compressed into a small one by using _____
a) LSI semiconductor
b) VLSI semiconductor
c) CDR semiconductor
d) SSI semiconductor

Answer: b
Explanation: VLSI (Very Large Scale Integration) semiconductor is used in modern computers to short the size of memory.

255. VLSI chip utilizes _____
a) NMOS
b) CMOS
c) BJT
d) All of the Mentioned

Answer: d
Explanation: VLSI (Very Large Scale Integration) is a memory chip which is made up of NMOS, CMOS, BJT, and BiCMOS. It can include 10,000 to 100,000 gates per IC.

256. CD-ROM refers to _____
a) Floppy disk
b) Compact Disk-Read Only Memory
c) Compressed Disk-Read Only Memory
d) Compressed Disk- Random Access Memory

Answer: b
Explanation: CD-ROM refers to Compact Disk-Read Only Memory.

DIWAKAR EDUCATION HUB

257. Data stored in an electronic memory cell can be accessed at random and on demand using _____

a) Memory addressing
b) Direct addressing
c) Indirect addressing
d) Control Unit

Answer: b
Explanation: Direct addressing eliminates the need to process a large stream of irrelevant data in order to the desired data word.

258. The full form of PLD is _____

a) Programmable Large Device
b) Programmable Long Device
c) Programmable Logic Device
d) Programmable Lengthy Device

Answer: c
Explanation: The full form of PLD is Programmable Logic Device.

259. The evolution of PLD began with _____

a) EROM
b) RAM
c) PROM
d) EEPROM

Answer: a
Explanation: The evolution of PLD (Programmable Logic Device) began with Programmable Read Only Memory (i.e. PROM). Here, the ROM can be externally programmed as per the user.

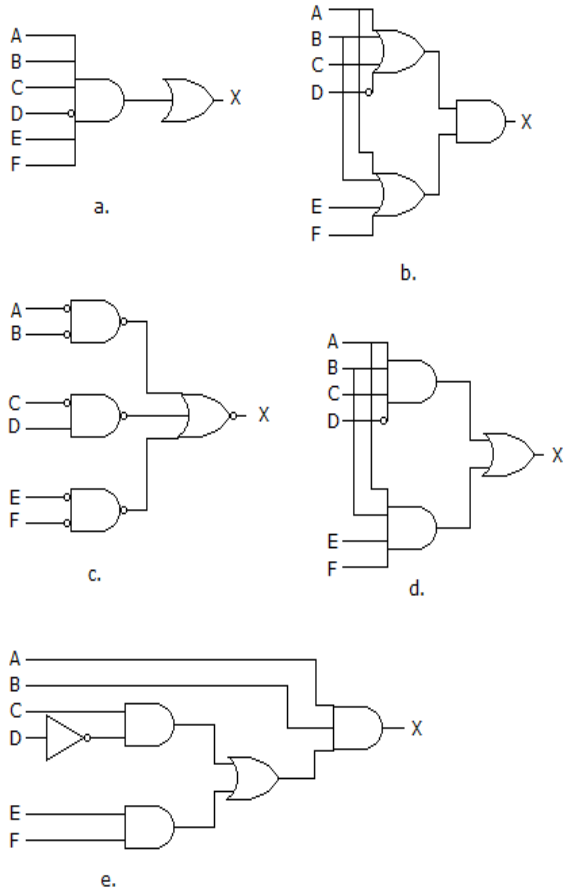
260. A ROM is defined as _____

a) Read Out Memory
b) Read Once Memory
c) Read Only Memory
d) Read One Memory
View Answer

Answer: c
Explanation: A ROM is defined as Read Only Memory which can read the instruction stored in a computer.

261. Which of the circuits in figure (a to d) is the sum-of-products implementation of figure (e)?

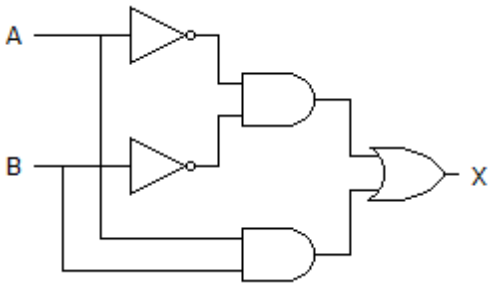
DIWAKAR EDUCATION HUB



a) a
b) b
c) c
d) d

Answer: d
Explanation: SOP means Sum Of Products form which represents the sum of product terms having variables in complemented as well as in uncomplemented form. Here, the diagram of d contains the OR gate followed by the AND gates, so it is in SOP form.

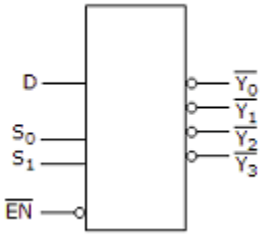
262. Which of the following logic expressions represents the logic diagram shown?



a) X=AB'+A'B
b) X=(AB)'+AB
c) X=(AB)'+A'B'
d) X=A'B'+AB

Answer: d
Explanation: 1st output of AND gate is = A'B'
2nd AND gate's output is = AB and, OR gate's output is = (A'B')+(AB) = AB + A'B'.

263. The device shown here is most likely a _____

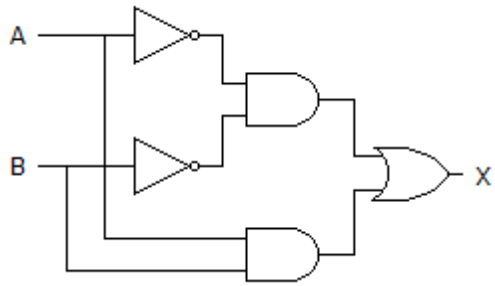


a) Comparator
b) Multiplexer
c) Inverter
d) Demultiplexer

DIWAKAR EDUCATION

Answer: d
Explanation: The given diagram is demultiplexer, because it takes single input & gives many outputs. A demultiplexer is a combinational circuit that takes a single output and latches it to multiple outputs depending on the select lines.

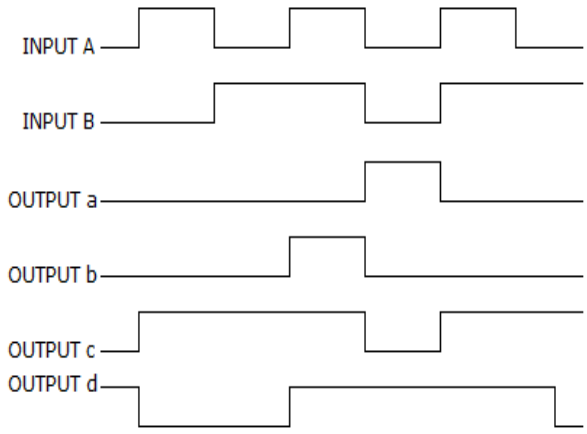
264. What type of logic circuit is represented by the figure shown below?



a) XOR
b) XNOR
c) AND
d) XAND

Answer: b
Explanation: After solving the circuit we get (A'B')+AB as output, which is XNOR operation. Thus, it will produce 1 when inputs are even number of 1s or all 0s, and produce 0 when input is odd number of 1s.

265. For a two-input XNOR gate, with the input waveforms as shown below, which output waveform is correct?



a) d
b) a
c) c
d) b

Answer: a
Explanation: When both inputs are same then the o/p is high for a XNOR gate.
i.e., A B O/P
0 0 1
0 1 0
1 0 0
1 1 1.
Thus, it will produce 1 when inputs are even number of 1s or all 0s, and produce 0 when input is odd number of 1s.

266. Which of the following combinations of logic gates can

DIWAKAR EDUCATION

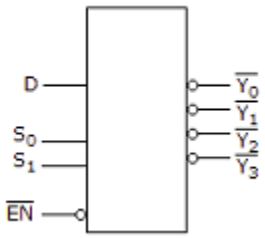
decode binary 1101?
a) One 4-input AND gate
b) One 4-input AND gate, one inverter
c) One 4-input AND gate, one OR gate
d) One 4-input NAND gate, one inverter

Answer: b
Explanation: For decoding any number output must be high for that code and this is possible in One 4-input NAND gate, one inverter option only. A decoder is a combinational circuit that converts binary data to n-coded data upto 2^n outputs.

267. What is the indication of a short to ground in the output of a driving gate?
a) Only the output of the defective gate is affected
b) There is a signal loss to all load gates
c) The node may be stuck in either the HIGH or the LOW state
d) The affected node will be stuck in the HIGH state

Answer: b
Explanation: Short to ground in the output of a driving gate indicates of a signal loss to all load gates. This results in information being disrupted and loss of data.

268. For the device shown here, assume the D input is LOW, both S inputs are LOW and the input is LOW. What is the status of the Y' outputs?



a) All are HIGH
b) All are LOW
c) All but Y0 are LOW
d) All but Y0 are HIGH

Answer: d
Explanation: In the given diagram, S0 and S1 are selection bits. So,
I/P S0 S1 O/P
D = 0 0 0 Y0
D = 0 0 1 Y1
D = 0 1 0 Y2
D = 0 1 1 Y3
Hence, inputs are S0 and S1 are Low means 0, so output is Y0 and rest all are HIGH.

269. The carry propagation can be expressed as _____
a) Cp = AB
b) Cp = A + B

DIWAKAR EDUCATION HUB

c) All but Y0 are LOW
d) All but Y0 are HIGH

Answer: b
Explanation: This happens in parallel adders (where we try to add numbers in parallel via more than one adders). A carry propagation occurs when carry from one adder needs to be forwarded to other adder and that second adder is holding the computation (addition) because carry from first adder has not come yet. So, there is a slight delay for second adder and this is known as carry propagation.

270. 3 bits full adder contains _____
a) 3 combinational inputs
b) 4 combinational inputs
c) 6 combinational inputs
d) 8 combinational inputs

Answer: d
Explanation: Full Adder is a combinational circuit with 3 input bits and 2 output bits CARRY and SUM. Three bits full adder requires $2^3 = 8$ combinational circuits.

271. The 'heart' of the processor which performs many different operations _____
a) Arithmetic and logic unit
b) Motherboard
c) Control Unit
d) Memory

Answer: a
Explanation: The Arithmetic and logic unit performs all the basic operations of the computer system. It performs all the arithmetic(+,-,*,/,etc) as well as the logical operations(AND, OR, NOT, etc.).

272. ALU is the place where the actual executions of instructions take place during the processing operation.
a) True
b) False

Answer: a
Explanation: ALU is a combinational electronic circuit which basically performs all the logical or the bitwise operations and the arithmetic operations. Therefore, it is the place where the actual executions of instructions take place.

273. Which of the following is not a bitwise operator?
a) |
b) ^

DIWAKAR EDUCATION HUB

c) .
d) <<

Answer: c
Explanation: All except the dot(.) operator are bitwise operators.
| : Bitwise OR
^ : Bitwise XOR
<< : Shift Left

274. The sign magnitude representation of -1 is _____
a) 0001
b) 1110
c) 1000
d) 1001

Answer: d
Explanation: The first leftmost bit i.e. the most significant bit in the sign magnitude represents if the number is positive or negative. If the MSB is 1, the number is negative else if it is 0, the number is positive. Here, +1=0001 and for -1=1001.

275. IEEE stands for _____
a) Instantaneous Electrical Engineering
b) Institute of Emerging Electrical Engineers
c) Institute of Emerging Electronic Engineers
d) Institute of Electrical and electronics engineers

Answer: d
Explanation: The IEEE is an organization of professionals in the field of electronics and electrical engineering. IEEE has given certain standards of its own which are followed in the field of computer science and electrical engineering.

276. The ALU gives the output of the operations and the output is stored in the _____
a) Memory Devices
b) Registers
c) Flags
d) Output Unit

Answer: b
Explanation: Any output generated by the ALU gets stored in the registers. The registers are the temporary memory locations within the processor that are connected by signal paths to the CPU.

278. The process of division on memory spaces is called _____
a) Paging
b) Segmentation

DIWAKAR EDUCATION HUB

c) Bifurcation
d) Dynamic Division

Answer: b
Explanation: The memory space is divided into segments of dynamic size. The programmer is aware of the segmentation and can reallocate the segments accordingly.

279. Number of bits in ALU is

_____

a) 4
b) 8
c) 16
d) 2

Answer: c
Explanation: Arithmetic and Logic Unit consists of 16bits. They perform certain Arithmetic and bitwise operations (add, subtract, AND, OR, XOR, Increment, decrement, shift).

280. Which flag indicates the number of 1 bit that results from an operation?
a) Zero
b) Parity
c) Auxiliary
d) Carry

Answer: b
Explanation: The parity flag indicates the number of 1 bit in any operation. The resultant bit is called the parity bit. The main aim of the parity bit is to check for errors.

281. The bitwise complement of 0 is

_____

a) 00000001
b) 10000000
c) 11111111
d) 11111110

Answer: c
Explanation: Bitwise complement is basically used to convert all the 0 digits to 1 and the 1s to 0s.
So, for 0 = 00000000(in 8-bits) ::: 11111111(1s complement). The bitwise complement is often referred to as the 1s complement.

282. Unicode provides a consistent way of encoding multilingual plain text.
a) True
b) False

Answer: a
Explanation: Unicode defines codes for characters used in all major languages of the world.
It is a coding system which supports almost all the languages. It defines

DIWAKAR EDUCATION HUB

special codes for different characters, symbols, diacritics, etc.

283. Which of the following is not a type of numeric value in zoned format?
a) Positive
b) Negative
c) Double
d) Unsigned

Answer: c
Explanation: The zoned format can represent numeric values of type Positive, negative and unsigned numbers. A sign indicator is used in the zone position of the rightmost digit.

284. The sign indicator of unsigned numbers is _____
a) C
b) D
c) F
d) X

Answer: c
Explanation: A sign indicator is used in the zone position of the rightmost digit. A sign indicator C is used for positive, D for negative and F is used for negative numbers.

285. The EBCDIC value of the number 345 in zoned format is _____
a) F3F4F5
b) E3E4E5
c) F3F4C5
d) F3F4D5

Answer: a
Explanation: F is used for the representation of unsigned numbers therefore, F3F4F5 represents 345. F3F4C5 represents +345 . F3F4D5 represents -345.

286. Which of the following logic families has the highest maximum clock frequency?
a) S-TTL
b) AS-TTL
c) HS-TTL
d) HCMOS

Answer: b
Explanation: AS-TTL (Advanced Schottky) has a maximum clock frequency of 105 MHz. S-TTL (Schottky High Speed TTL) has 100 MHz. Found nothing as HS-TTL. There are H and S separate TTL. HCMOS has 50 MHz clock frequency.

287. Why is the fan-out of CMOS gates frequency dependent?
a) Each CMOS input gate has a

DIWAKAR EDUCATION HUB

specific propagation time and this limits the number of different gates that can be connected to the output of a CMOS gate
b) When the frequency reaches the critical value the gate will only be capable of delivering 70% of the normal output voltage and consequently the output power will be one-half of normal and this defines the upper operating frequency
c) The higher number of gates attached to the output the more frequently they will have to be serviced thus reducing the frequency at which each will be serviced with an input signal
d) The input gates of the FETs are predominantly capacitive and as the signal frequency increases the capacitive loading also increases thereby limiting the number of loads that may be attached to the output of the driving gate

Answer: d
Explanation: Fan out is the measure of maximum number of inputs that a single logic gate output can drive. Actually power dissipation in CMOS circuits depends on clock frequency. As the frequency increases Pd also increases so fan-out depends on frequency.

288. Logic circuits that are designated as buffers, drivers or buffers/drivers are designed to have:
a) A greater current/voltage capability than an ordinary logic circuit
b) Greater input current/voltage capability than an ordinary logic circuit
c) A smaller output current/voltage capability than an ordinary logic
d) Greater the input and output current/voltage capability than an ordinary logic circuit

Answer: a
Explanation: Buffer circuits are usually incorporated to isolate the input from the output. Logic circuits that are designated as buffers, drivers or buffer/drivers are designed to have a greater current/voltage capability than an ordinary logic circuit.

289. Which of the following will not normally be found on a data sheet?
a) Minimum HIGH level output voltage

DIWAKAR EDUCATION HUB

b) Maximum LOW level output voltage
c) Minimum LOW level output voltage
d) Maximum HIGH level input current

Answer: c
Explanation: Minimum LOW level output voltage will not normally be found on a data sheet.

290. Which of the following logic families has the shortest propagation delay?
a) S-TTL
b) AS-TTL
c) HS-TTL
d) HCMOS

Answer: b
Explanation: AS-TTL (Advanced Schottky) has a maximum clock frequency that is 105 MHz. So, the propagation delay will be given by 1/105 sec which is the lowest one. It is followed by S-TTL and HCMOS in terms of increasing propagation delay.

291. ARM stands for _____
a) Advanced Rate Machines
b) Advanced RISC Machines
c) Artificial Running Machines
d) Aviary Running Machines

Answer: b
Explanation: ARM is a type of system architecture.

292. The main importance of ARM micro-processors is providing operation with _____
a) Low cost and low power consumption
b) Higher degree of multi-tasking
c) Lower error or glitches
d) Efficient memory management

Answer: a
Explanation: The Stand alone feature of the ARM processors is that they're economically viable.

293. ARM processors where basically designed for _____
a) Main frame systems
b) Distributed systems
c) Mobile systems
d) Super computers

Answer: c
Explanation: These ARM processors are designed for handheld devices.

294. The ARM processors don't support Byte addressability.
a) True
b) False

DIWAKAR EDUCATION HUB

Answer: b
Explanation: The ability to store data in the form of consecutive bytes.

295. The address space in ARM is _____
a) 2^{24}
b) 2^{64}
c) 2^{16}
d) 2^{32}

Answer: d
Explanation: None.

296. _____ have been developed specifically for pipelined systems.
a) Utility software
b) Speed up utilities
c) Optimizing compilers
d) None of the mentioned

Answer: c
Explanation: The compilers which are designed to remove redundant parts of the code are called as optimizing compilers.

297. The pipelining process is also called as _____
a) Superscalar operation
b) Assembly line operation
c) Von Neumann cycle
d) None of the mentioned

Answer: b
Explanation: It is called so because it performs its operation at the assembly level.

298. The fetch and execution cycles are interleaved with the help of _____
a) Modification in processor architecture
b) Clock
c) Special unit
d) Control unit

Answer: b
Explanation: The time cycle of the clock is adjusted to perform the interleaving.

299. Each stage in pipelining should be completed within _____ cycle.
a) 1
b) 2
c) 3
d) 4

Answer: a
Explanation: The stages in the pipelining should get completed within one cycle to increase the speed of performance.

DIWAKAR EDUCATION HUB

300. In pipelining the task which requires the least time is performed first.
a) True
b) False

Answer: b
Explanation: This is done to avoid starvation of the longer task.

301. If a unit completes its task before the allotted time period, then _____
a) It'll perform some other task in the remaining time
b) Its time gets reallocated to a different task
c) It'll remain idle for the remaining time
d) None of the mentioned

Answer: c
Explanation: None.

302. To increase the speed of memory access in pipelining, we make use of _____
a) Special memory locations
b) Special purpose registers
c) Cache
d) Buffers

Answer: c
Explanation: By using the cache we can reduce the speed of memory access by a factor of 10.

303. The periods of time when the unit is idle is called as _____
a) Stalls
b) Bubbles
c) Hazards
d) Both Stalls and Bubbles

Answer: d
Explanation: The stalls are a type of hazards that affect a pipelined system.

304. The contention for the usage of a hardware device is called _____
a) Structural hazard
b) Stalk
c) Deadlock
d) None of the mentioned

Answer: a
Explanation: None.

305. The situation wherein the data of operands are not available is called _____
a) Data hazard
b) Stock
c) Deadlock
d) Structural hazard

Answer: a
Explanation: Data hazards are

DIWAKAR EDUCATION HUB

generally caused when the data is not ready on the destination side.

306. The decimal numbers represented in the computer are called as floating point numbers, as the decimal point floats through the number.
a) True
b) False

Answer: a
Explanation: By doing this the computer is capable of accommodating the large float numbers also.

307. The numbers written to the power of 10 in the representation of decimal numbers are called as _____
a) Height factors
b) Size factors
c) Scale factors
d) None of the mentioned

Answer: c
Explanation: These are called as scale factors cause they're responsible in determining the degree of specification of a number.

308. If the decimal point is placed to the right of the first significant digit, then the number is called _____
a) Orthogonal
b) Normalized
c) Determinate
d) None of the mentioned

Answer: b
Explanation: None.

309. _____ constitute the representation of the floating number.
a) Sign
b) Significant digits
c) Scale factor
d) All of the mentioned

Answer: d
Explanation: The following factors are responsible for the representation of the number.

310. The sign followed by the string of digits is called as _____
a) Significant
b) Determinant
c) Mantissa
d) Exponent

Answer: c
Explanation: The mantissa also consists of the decimal point.

311. In IEEE 32-bit representations, the mantissa of the fraction is said to occupy _____ bits.

DIWAKAR EDUCATION HUB

a) 24
b) 23
c) 20
d) 16

Answer: b
Explanation: The mantissa is made to occupy 23 bits, with 8 bit exponent.

312. The normalized representation of $0.0010110 * 2_9$ is _____
a) 0 10001000 0010110
b) 0 10000101 0110
c) 0 10101010 1110
d) 0 11110100 11100

Answer: b
Explanation: Normalized representation is done by shifting the decimal point.

313. The 32 bit representation of the decimal number is called as _____
a) Double-precision
b) Single-precision
c) Extended format
d) None of the mentioned

Answer: b
Explanation: None.

314. In 32 bit representation the scale factor as a range of _____
a) -128 to 127
b) -256 to 255
c) 0 to 255
d) None of the mentioned

Answer: a
Explanation: Since the exponent field has only 8 bits to store the value.

315. In double precision format, the size of the mantissa is _____
a) 32 bit
b) 52 bit
c) 64 bit
d) 72 bit

Answer: b
Explanation: The double precision format is also called as 64 bit representation.

316. The DMA differs from the interrupt mode by _____
a) The involvement of the processor for the operation
b) The method of accessing the I/O devices
c) The amount of data transfer possible
d) None of the mentioned

Answer: d
Explanation: DMA is an approach of performing data transfers in bulk between memory and the external

DIWAKAR EDUCATION HUB

device without the intervention of the processor.

317. The DMA transfers are performed by a control circuit called as _____
a) Device interface
b) DMA controller
c) Data controller
d) Overlooker

Answer: b
Explanation: The Controller performs the functions that would normally be carried out by the processor.

318. In DMA transfers, the required signals and addresses are given by the _____
a) Processor
b) Device drivers
c) DMA controllers
d) The program itself

Answer: c
Explanation: The DMA controller acts as a processor for DMA transfers and overlooks the entire process.

319. After the completion of the DMA transfer, the processor is notified by _____
a) Acknowledge signal
b) Interrupt signal
c) WMFC signal
d) None of the mentioned

Answer: b
Explanation: The controller raises an interrupt signal to notify the processor that the transfer was complete.

320. The DMA controller has _____ registers.
a) 4
b) 2
c) 3
d) 1

Answer: c
Explanation: The Controller uses the registers to store the starting address, word count and the status of the operation.

330. When the R/W bit of the status register of the DMA controller is set to 1.
a) Read operation is performed
b) Write operation is performed
c) Read & Write operation is performed
d) None of the mentioned

Answer: a
Explanation: None.

DIWAKAR EDUCATION HUB

331. The controller is connected to the ____
a) Processor BUS
b) System BUS
c) External BUS
d) None of the mentioned

Answer: b
Explanation: The controller is directly connected to the system BUS to provide faster transfer of data.

332. Can a single DMA controller perform operations on two different disks simultaneously?
a) True
b) False

Answer: a
Explanation: The DMA controller can perform operations on two different disks if the appropriate details are known.

333. The technique whereby the DMA controller steals the access cycles of the processor to operate is called _____
a) Fast conning
b) Memory Con
c) Cycle stealing
d) Memory stealing

Answer: c
Explanation: The controller takes over the processor's access cycles and performs memory operations.

334. The technique where the controller is given complete access to main memory is _____
a) Cycle stealing
b) Memory stealing
c) Memory Con
d) Burst mode

Answer: d
Explanation: The controller is given full control of the memory access cycles and can transfer blocks at a faster rate.

335. The side of the interface circuits, that has the data path and the control signals to transfer data between interface and device is _____
a) BUS side
b) Port side
c) Hardwell side
d) Software side

Answer: b
Explanation: This side connects the device to the motherboard.

DIWAKAR EDUCATION HUB

336. What is the interface circuit?
a) Helps in installing of the software driver for the device
b) Houses the buffer that helps in data transfer
c) Helps in the decoding of the address on the address BUs
d) None of the mentioned

Answer: c
Explanation: Once the address is put on the BUS the interface circuit decodes the address and uses the buffer space to transfer data.

337. The conversion from parallel to serial data transmission and vice versa takes place inside the interface circuits.
a) True
b) False

Answer: a
Explanation: By doing this the interface circuits provide a better interconnection between devices.

338. The interrupt-request line is a part of the _____
a) Data line
b) Control line
c) Address line
d) None of the mentioned

Answer: b
Explanation: The Interrupt-request line is a control line along which the device is allowed to send the interrupt signal.

339. The return address from the interrupt-service routine is stored on the _____
a) System heap
b) Processor register
c) Processor stack
d) Memory

Answer: c
Explanation: The Processor after servicing the interrupts as to load the address of the previous process and this address is stored in the stack.

340. The signal sent to the device from the processor to the device after receiving an interrupt is _____
a) Interrupt-acknowledge
b) Return signal
c) Service signal
d) Permission signal

Answer: a
Explanation: The Processor upon receiving the interrupt should let the device know that its request is received.

DIWAKAR EDUCATION HUB

341. When the process is returned after an interrupt service _____ should be loaded again.
i) Register contents
ii) Condition codes
iii) Stack contents
iv) Return addresses
a) i, iv
b) ii, iii and iv
c) iii, iv
d) i, ii

Answer: d
Explanation: None.

342. The time between the receiver of an interrupt and its service is _____
a) Interrupt delay
b) Interrupt latency
c) Cycle time
d) Switching time

Answer: b
Explanation: The delay in servicing of an interrupt happens due to the time is taken for contact switch to take place.

343. Interrupts form an important part of _____ systems.
a) Batch processing
b) Multitasking
c) Real-time processing
d) Multi-user

Answer: c
Explanation: This forms an important part of the Real time system since if a process arrives with greater priority then it raises an interrupt and the other process is stopped and the interrupt will be serviced.

344. A single Interrupt line can be used to service n different devices.
a) True
b) False

Answer: a
Explanation: None.

345. *@Ac# is a type of _____ data.
a) Symbolic
b) Alphanumeric
c) Alphabetic
d) Numeric

Answer: b
Explanation: Alphanumeric data consists of symbols. Alphanumeric data may be a letter, either in uppercase or lowercase or some special symbols like #,^,*,(, etc.)

346. Which of the following is not a valid representation in bits?
a) 8-bit

DIWAKAR EDUCATION HUB

b) 24-bit
c) 32-bit
d) 64-bit

Answer: b
Explanation: There are no criteria like the 24-bit representation of numbers. Numbers can be written in 8-bit, 16-bit, 32-bit and 64-bit as per the IEEE format.

347. What are the entities whose values can be changed called?
a) Constants
b) Variables
c) Modules
d) Tokens

Answer: b
Explanation: Variables are the data entities whose values can be changed. Constants have a fixed value. Tokens are the words which are easily identified by the compiler.

348. Which of the following is not a basic data type in C language?
a) float
b) int
c) real
d) char

Answer: c
Explanation: There are 5 basic data types in C language: int, char, float, double, void.

Int is for the representation of integers, char is for strings and characters, float and double are for floating point numbers whereas void is a valueless special data type.

349. BOOLEAN is a type of data type which basically gives a tautology or fallacy.
a) True
b) False

Answer: a
Explanation: A Boolean representation is for giving logical values. It returns either true or false. If a result gives a truth value, it is called tautology whereas if it returns a false term, it is referred to as fallacy.

350. What does FORTRAN stands for?
a) Formula Transfer
b) Formula Transformation
c) Formula Translation
d) Format Transformation

Answer: c
Explanation: FORTRAN is a type of computer language. It was developed for solving mathematical and scientific problems. It is very commonly used among the scientific community.

351. The program written by the programmer in high level language is called _____
a) Object Program
b) Source Program
c) Assembled Program
d) Compiled Program

Answer: b
Explanation: The program written by the programmer is called a source program. The program generated by the compiler after compilation is called an object program. The object program is in machine language.

352. A standardized language used for commercial applications.
a) C
b) Java
c) COBOL
d) FORTRAN

Answer: c
Explanation: COBOL is a language used in business and commercial applications. It stands for Common Business Oriented Language. It is imperative, procedural as well as object oriented language.

353. _____ define how the locations can be used.
a) Data types
b) Attributes
c) Links
d) Data Objects

Answer: b
Explanation: Attributes can determine how any location can be used. Attributes can be type, name, component, etc. Data objects are the variables and constants in a program.

354. The decimal equivalent of the binary number (1011.011)2 is _____
a) (11.375)10
b) (10.123)10
c) (11.175)10
d) (9.23)10

Answer: a
Explanation: Binary to Decimal conversion is obtained by multiplying 2 to the power of base index along with the value at that index position.
$1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1*2^0 + 0 * 2^{-1} + 1 * 2^{-2} + 1 * 2^{-3}$ = (11.375)10
Hence, (1011.011)2 = (11.375)10

355. An important drawback of binary system is _____
a) It requires very large string of 1's and 0's to represent a decimal number
b) It requires sparingly small string of 1's and 0's to represent a decimal number
c) It requires large string of 1's and small string of 0's to represent a decimal number

d) It requires small string of 1's and large string of 0's to represent a decimal number

Answer: a
Explanation: The most vital drawback of binary system is that it requires very large string of 1's and 0's to represent a decimal number. Hence, Hexadecimal systems are used by processors for calculation purposes as it compresses the long binary strings into small parts.

356. The decimal equivalent of the octal number $(645)_8$ is _____
a) $(450)_{10}$
b) $(451)_{10}$
c) $(421)_{10}$
d) $(501)_{10}$

Answer: c
Explanation: Octal to Decimal conversion is obtained by multiplying 8 to the power of base index along with the value at that index position. The decimal equivalent of the octal number $(645)_8$ is $6 * 8^2 + 4 * 8^1 + 5 * 8^0 = 6 * 64 + 4 * 8 + 5 = 384 + 32 + 5 = (421)_{10}$.

357. The largest two digit hexadecimal number is _____
a) (FE)16
b) (FD)16
c) (FF)16
d) (EF)16

Answer: c
Explanation: (FE)16 is 254 in decimal system, while (FD)16 is 253. (EF)16 is 239 in decimal system. And, (FF)16 is 255. Thus, The largest two-digit hexadecimal number is (FF)16.

358. Representation of hexadecimal number (6DE)H in decimal:
a) $6 * 16^2 + 13 * 16^1 + 14 * 16^0$
b) $6 * 16^2 + 12 * 16^1 + 13 * 16^0$
c) $6 * 16^2 + 11 * 16^1 + 14 * 16^0$
d) $6 * 16^2 + 14 * 16^1 + 15 * 16^0$

Answer: a
Explanation: Hexadecimal to Decimal conversion is obtained by multiplying 16 to the power of base index along with the value at that index position. In hexadecimal number D & E represents 13 & 14 respectively. So, $6DE = 6 * 16^2 + 13 * 16^1 + 14 * 16^0$.

359. The quantity of double word is _____
a) 16 bits
b) 32 bits
c) 4 bits
d) 8 bits

Answer: b
Explanation: One word means 16 bits, Thus, the quantity of double word is 32 bits.

360. What is the addition of the binary numbers 11011011010 and 010100101?
a) 0111001000
b) 1100110110
c) 11101111111
d) 10011010011

Answer: c
Explanation: The rules for Binary Addition are :
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 (Carry 1)

```
    1

 1 1 0 1 1 0 1 1 0 1 0

+ 0 0 0 1 0 1 0 0 1 0 1
 _____________________
 1 1 1 0 1 1 1 1 1 1 1
 _____________________
```

361. Perform binary addition: 101101 + 011011 = ?
a) 011010
b) 1010100
c) 101110
d) 1001000

Answer: d
Explanation:The rules for Binary Addition are :
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 (Carry 1)

```
 1 1 1 1 1 1
   1 0 1 1 0 1
+ 0 1 1 0 1 1
 _______________
 1 0 0 1 0 0 0
 _______________
```

Therefore, the addition of 101101 + 011011 = 1001000.

362. Perform binary subtraction: 101111 – 010101 = ?
a) 100100
b) 010101
c) 011010
d) 011001

Answer: c
Explanation: The rules for Binary Subtraction are :
0 – 0 = 0
0 – 1 = 1 (Borrow 1)
1 – 0 = 1
1 – 1 = 0

```
  1 0 1 1 1 1
- 0 1 0 1 0 1
 _____________
  0 1 1 0 1 0
 _____________
```

Therefore, The subtraction of 101111 – 010101 = 011010.

363. Binary subtraction of 100101 – 011110 is
a) 000111

b) 111000
c) 010101
d) 101010

Answer: a
Explanation: The rules for Binary Subtraction are :
0 – 0 = 0
0 – 1 = 1 (Borrow 1)
1 – 0 = 1
1 – 1 = 0

```
 1 0 0 1 0 1
- 0 1 1 1 1 0
 ___________
 0 0 0 1 1 1
 ___________
```

Therefore, The subtraction of 100101 – 011110 = 000111.

364. Perform multiplication of the binary numbers: 01001 × 01011 = ?
a) 001100011
b) 110011100
c) 010100110
d) 101010111

Answer: a
Explanation: The rules for binary multiplication are:
0 * 0 = 0
0 * 1 = 0
1 * 0 = 0
1 * 1 = 1

```
        0 1 0 0 1
      x 0 1 0 1 1
      ____________
```

```
        0 1 0 0 1
      0 1 0 0 1 0
    0 0 0 0 0 0 0
  0 1 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0
_____________________
0 0 1 1 0 0 0 1 1
_____________________
```

Therefore, 01001 × 01011 = 001100011.

365. 100101 × 0110 = ?
a) 1011001111
b) 0100110011
c) 101111110
d) 0110100101

Answer: c
Explanation: The rules for binary multiplication are:
0 * 0 = 0
0 * 1 = 0
1 * 0 = 0
1 * 1 = 1

```
          1 0 0 1 0 1
        x     0 1 1 0
        ______________
        0 0 0 0 0 0
      1 0 0 1 0 1 0
    1 0 0 1 0 1 0 0
  0 0 0 0 0 0 0 0 0 0
_____________________
0 1 1 0 1 1 1 1 0
_____________________
```

Therefore, 100101 x 0110 = 011011110.

366. On multiplication of (10.10) and (01.01), we get
a) 101.0010
b) 0010.101
c) 011.0010
d) 110.0011

Answer: c
Explanation: The rules for binary multiplication are:
0 * 0 = 0
0 * 1 = 0
1 * 0 = 0
1 * 1 = 1

```
      1 0.1 0
  x   0 1.0 1
    ___________
      1 0 1 0
    0 0 0 0 0
  1 0 1 0 0 0
0 0 0 0 0 0 0
_______________
0 1 1.0 0 1 0
_______________
```

Therefore, 10.10 x 01.01 = 011.0010.

367. Divide the binary numbers: 111101 ÷ 1001 and find the remainder
a) 0010
b) 1010
c) 1100
d) 0011

Answer: d
Explanation: Binary Division is accomplished using long division method.

```
1 0 0 1 ) 1 1 1 1 0 1 ( 1 1
        1 0 0 1
        ___________
        0 1 1 0 0
          1 0 0 1
          ___________
          0 1 1 1
```

Therefore, the remainder of 111101 ÷ 1001 = 0111.

368. CPU has built-in ability to execute a particular set of machine instructions, called as _____
a) Instruction Set
b) Registers
c) Sequence Set
d) User instructions

Answer: a
Explanation: An instruction is any task which is to be performed by the processor. Instructions are stored in the register. Instruction set is the set of machine instructions.

369. Opcode indicates the operations to be performed.
a) True
b) False

Answer: a
Explanation: Every instruction has an opcode. Additionally, it may have one

or more operands and the op code indicates the operation to be performed.

370. The length of a register is called _____
a) word limit
b) word size
c) register limit
d) register size

Answer: b
Explanation: The length of a register is called word size. It tells the number of bits a register can store.
Registers are a part of the CPU.

371. The _____ holds the contents of the accessed memory word.
a) MAR
b) MBR
c) PC
d) IR

Answer: b
Explanation: The MBR holds the contents of the accessed (read/written) memory word.
MBR stands for Memory Buffer Register.

372. Which of the following is not a visible register?
a) General Purpose Registers
b) Address Register
c) Status Register
d) MAR

Answer: d
Explanation: MAR or the memory address register is not a visible register. This register is user inaccessible. It contains the address of the memory block to be read or written to.

373. Which of the following is a data transfer instruction?
a) STA 16-bit address
b) ADD A, B
c) MUL C, D
d) RET

Answer: a
Explanation: The instruction STA 16-bit address is a data transfer instruction.
STA means Store in Accumulator.

374. What is correct instruction if you want the control to go to the location 2000h?
a) MOV 2000h
b) MOV A, 2000h
c) JMP 2000h
d) RET 2000h

DIWAKAR EDUCATION HUB

Answer: c
Explanation: The JMP instruction is used to move to a particular location. In 8085 microprocessor, JMP statement tells the processor to go to location 2000h (here).

375. What kind of a flag is the sign flag?
a) General Purpose
b) Status
c) Address
d) Instruction

Answer: b
Explanation: Sign flag is a type of status register or the flag register. It is used to indicate the sign of certain bits.

376. The number of sign bits in a 32-bit IEEE format _____
a) 1
b) 11
c) 9
d) 23

Answer: a
Explanation: There is only 1 sign bit in all the standards. In a 32-bit format, there is 1 sign bit, 8 bits for the exponent and 23 bits for the mantissa.

377. New CPU whose instruction set includes the instruction set of its predecessor CPU is said to be _____ with its predecessor.
a) fully compatible
b) forward compatible
c) compatible
d) backward compatible

Answer: d
Explanation: The CPU is called backward compatible since it contains the instruction set of its predecessor. Manufacturers tend to group their CPUs into families having similar instruction set.

378. What does ASCII stand for?
a) American Standard Code for Information Interchange
b) American Scientific Code for Information Interchange
c) American Scientific Code for Interchanging Information
d) American Standard Code for Interchanging Information

Answer: a
Explanation: The ASCII codes are used to represent the bits into symbols and vice versa. ASCII is the American Standard Code which is used to exchange information.

DIWAKAR EDUCATION HUB

379. The decimal representation for the character '!' in ASCII is

_____

a) 31
b) 32
c) 33
d) 34

Answer: c
Explanation: The decimal representation of a few basic characters are:
33 : !
34 : ”
35: #
36 :$.

380. The two types of ASCII are _____ and _____

a) ASCII-4 and ASCII-8
b) ASCII-8 and ASCII-16
c) ASCII-7 and ASCII-8
d) ASCII-4 and ASCII-16

Answer: c
Explanation: The two types of ASCII are ASCII-7 and ASCII-8. ASCII-7 uses 7 bits for the representation of numbers and ASCII-8 uses 8-bits.

381. Any set of digits or alphabets are generally referred as

_____

a) Characters
b) Symbols
c) Bits
d) Bytes

Answer: a
Explanation: We refer to the digits and alphabets generally as characters. A character is generally a unit of information in computers.

382. The first 128 characters are the same in both the types of ASCII i.e. ASCII-7 and ASCII-8.

a) True
b) False

Answer: a
Explanation: There are two types of ASCII codes: ASCII-7 and ASCII-8. ASCII-7 uses 7 bits to represent a number whereas ASCII-8 uses 8-bits to represent a number.

383. The number of characters that can be represented in ASCII-8 are

_____

a) 128
b) 256
c) 32
d) 64

Answer: b
Explanation: ASCII-8 can represent 256 different characters. ASCII-8 uses

DIWAKAR EDUCATION HUB

8-bits for the representation of numbers i.e. it can represent $2^8 = 256$ different characters.

384. The zone of alphabetic characters from A to O in ASCII is

_____

a) 1000
b) 0100
c) 0010
d) 0001

Answer: b
Explanation: The zone used by ASCII for alphabets is 0100. For e.g. A is represented as 0100(zone)0001(digit). The hex equivalent is 41 for A. The zone used by numbers is 0011.

385. The representation of the number 8 in binary in ASCII-8 format

_____

a) 00111000
b) 01001000
c) 1000
d) 00011000

Answer: a
Explanation: The ASCII-8 format will have 8 bits. The zone for the character 8 is 0011 and the digit is 1000. Therefore, its representation is 00111000.

386. Binary Coding for the letter X is

_____

a) 01011000
b) 00111000
c) 10001000
d) 00010100

Answer: a
Explanation: The binary coding for the letter X is 01011000. Here, 0101 is the zone whereas 1000 is the digit. The alphabets from P to Z have the zone 0101.

387. Express the ASCII equivalent of the signed binary number $(00110010)_2$.

a) 2
b) 1
c) A
d) ,

Answer: a
Explanation: The ASCII characters for the remaining options are:
1 : 00110001
A : 01000001
, : 00101100.

388. Computer has a built-in system clock that emits millions of regularly spaced electric pulses per _____ called clock cycles.

a) second

DIWAKAR EDUCATION HUB

b) millisecond
c) microsecond
d) minute

Answer: a
Explanation: The regularly spaced electric pulses per second are referred to as the clock cycles. All the jobs performed by the processor are on the basis of clock cycles.

389. It takes one clock cycle to perform a basic operation.
a) True
b) False

Answer: a
Explanation: It takes exactly one clock cycle to perform a basic operation, such as moving a byte of memory from a location to another location in the computer.

390. The operation that does not involves clock cycles is _____
a) Installation of a device
b) Execute
c) Fetch
d) Decode

Answer: a
Explanation: Normally, several clock cycles are required to fetch, execute and decode a particular program. Installation of a device is done by the system on its own.

391. The number of clock cycles per second is referred as _____
a) Clock speed
b) Clock frequency
c) Clock rate
d) Clock timing

Answer: a
Explanation: The number of clock cycles per second is the clock speed. It is generally measured in gigahertz(10^9 cycles/sec) or megahertz (10^6 cycles/sec).

392. CISC stands for _____
a) Complex Information Sensed CPU
b) Complex Instruction Set Computer
c) Complex Intelligence Sensed CPU
d) Complex Instruction Set CPU

Answer: b
Explanation: CISC is a large instruction set computer. It has variable length instructions. It also has variety of addressing modes.

393. Which of the following processor has a fixed length of instructions?
a) CISC
b) RISC

DIWAKAR EDUCATION HUB

c) EPIC
d) Multi-core

Answer: b
Explanation: The RISC which stands for Reduced Instruction set computer has a fixed length of instructions. It has a small instruction set. Also has reduced references to memory to retrieve operands.

394. Processor which is complex and expensive to produce is _____
a) RISC
b) EPIC
c) CISC
d) Multi-core

Answer: c
Explanation: CISC stands for complex instruction set computer. It is mostly used in personal computers. It has a large instruction set and a variable length of instructions.

395. The architecture that uses a tighter coupling between the compiler and the processor is _____
a) EPIC
b) Multi-core
c) RISC
d) CISC

Answer: a
Explanation: EPIC stands for Explicitly parallel instruction computing. It has a tighter coupling between the compiler and the processor. It enables the compiler to extract maximum parallelism in the original code.

396. MAR stands for _____
a) Memory address register
b) Main address register
c) Main accessible register
d) Memory accessible register

Answer: a
Explanation: The MAR stands for memory address register. It holds the address of the active memory location.

397. A circuitry that processes that responds to and processes the basic instructions that are required to drive a computer system is _____
a) Memory
b) ALU
c) CU
d) Processor

Answer: d
Explanation: The processor is responsible for processing the basic instructions in order to drive a

DIWAKAR EDUCATION HUB

computer. The primary functions of a processor are fetch, decode and execute.

398. The value of base in a decimal number system is _____
a) 8
b) 2
c) 10
d) 16

Answer: c
Explanation: A decimal number system consists of 10 digits from 0 to 9.
The definition of base describes it as a quantity to represent the number of digits present in that particular number system.
Therefore, here, the base is 10.

399. Convert : $(110)_2$ = (__)$_{10}$.
a) 4
b) 5
c) 6
d) 9

Answer: c
Explanation: The base 2 represents that the number is binary ,whereas, the base 10 represents that it is to be converted to the decimal format.
Conversion: $2^2 * 1 + 2^1 * 1 + 2^0 * 0 = 6$.

400. The 2's complement of 15 is _____
a) 0000
b) 0001
c) 0010
d) 0100

Answer: b
Explanation: 2's complement is obtained by adding 1 to the 1's complement of the number.
Here, Binary of 15 = 1111
1's complement of 15= 0000
2's complement of 15= 0000+1=0001.

401. Another name for base is _____
a) root
b) radix
c) entity
d) median

Answer: b
Explanation: Another name for base is radix. Base refers to the number of digits that a particular number system consists of.
The base of decimal number system is 10, binary is 2 and so on.

402. The decimal equivalent of $(0.101)_2$ will be _____
a) 0.5
b) 0.625

DIWAKAR EDUCATION HUB

c) 0.25
d) 0.875

Answer: b
Explanation: Since the base is 2 , it could be easily guessed that the number is binary. Conversion: $2^{-1} * 1 + 2^{-2} * 0 + 2^{-3} * 1 = 0.625$.

403. Which of the following is not a positional number system?
a) Roman Number System
b) Octal Number System
c) Binary Number System
d) Hexadecimal Number System

Answer: a
Explanation: The Roman number system isn't a positional number system since it uses symbols to represent numbers.
The octal number system uses digits from 0-7, the binary number system uses digits from 0-1 whereas, the hexadecimal number system uses digits from 0-15.

404. The value of radix in binary number system is _____
a) 2
b) 8
c) 10
d) 1

Answer: a
Explanation: In a binary number system, the value of base or radix is 2. The binary system uses only two digits for the representation of numbers, therefore its base id has chosen to be 2.

405. The binary equivalent of the decimal number 10 is _____
a) 0010
b) 10
c) 1010
d) 010

Answer: c
Explanation: To get the binary equivalent of any number, we need to divide the number by 2 and obtain the remainders as :

```
10/2=5 rem=0 ↑
5/2=2  rem=1 |
2/2=1  rem=0 |
1/2=0  rem=1 |
```

We then write the remainders in the reverse order as 1010 .

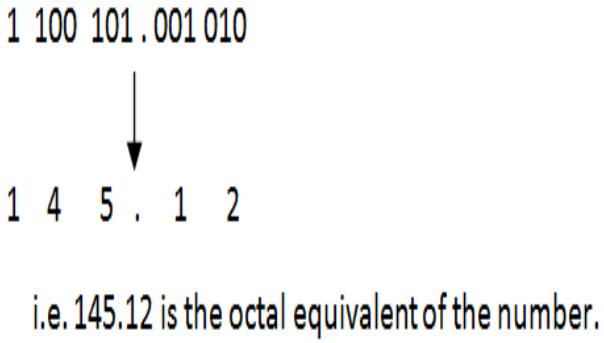
406. A computer language that is written in binary codes only is _____
a) machine language
b) C
c) C#
d) pascal

DIWAKAR EDUCATION HUB

Answer: a
Explanation: Machine Language is written in binary codes only. It can be easily understood by the computer and is very difficult for us to understand. A machine language, unlike other languages, requires no translators or interpreters.

407. The octal equivalent of 1100101.001010 is _____
a) 624.12
b) 145.12
c) 154.12
d) 145.21

Answer: b
Explanation: The octal equivalent is obtained by grouping the numbers into three, from right to left before decimal and from right to left after the decimal place.
Here,



i.e. 145.12 is the octal equivalent of the number.

408. The input hexadecimal representation of 1110 is
_____
a) 0111
b) E
c) 15
d) 14

Answer: b
Explanation: In hexadecimal number system, 1110 = 15, which is represented by the alphabet E.
Some representations are:
A 10
B 11
C 12
D 13
E 14
F 15.

409. What could be the maximum value of a single digit in an octal number system?
a) 8
b) 7
c) 6
d) 5

Answer: b
Explanation: The maximum value in any number system is one less than

KAR EDUCATION HUB

the value of the base. The base in an octal number system is 8, therefore, the maximum value of the single digit is 7. It takes digits from 0 to 7.

410. In a number system, each position of a digit represents a specific power of the base.
a) True
b) False

Answer: a
Explanation: In a number system, every digit is denoted by a specific power of base. Like in an octal system, consider the number 113, it will be represented as :
$8^2 * 1 + 8^1 * 1 + 8^0 * 3$.

411. The maximum number of bits sufficient to represent an octal number in binary is _____
a) 4
b) 3
c) 7
d) 8

Answer: b
Explanation: The octal number system comprises of only 8 digits. Hence, three bits (23 = 8) are sufficient to represent any octal number in the binary format.

412. The binary number 111 in octal format is _____
a) 6
b) 7
c) 8
d) 5

Answer: b
Explanation: Certain binary to octal representations are :
000=0
001=1
010=2
011=3
100=4
101=5
110=6
111=7.

413. Convert $(22)_8$ into its corresponding decimal number.
a) 28
b) 18
c) 81
d) 82

Answer: b
Explanation: To convert an octal number to decimal number:
$8^1 * 2 + 8^0 * 2 = 16 + 2 = 18$.
Hence, the decimal equivalent is 18.

414. What does the symbol D represent in a hexadecimal number

DIWAKAR EDUCATION HUB

system?
a) 8
b) 16
c) 13
d) 14

Answer: c
Explanation: The symbols A, B, C, D, E and F represent 10, 11, 12, 13, 14 and 15 respectively in a hexadecimal system. This system comprises of 15 numbers in total: digits from 0-9 and symbols from A to F.

415. ABC is a valid hexadecimal number.
a) True
b) False

Answer: a
Explanation: In a hexadecimal number system, alphabets are used for the representation of numbers from 10 to 15. Here, A represents 10, B represents 11 and C represents 12. Therefore, it is a valid hexadecimal number.

416. The maximum number of bits sufficient to represent a hexadecimal number in binary:
a) 4
b) 3
c) 7
d) 8

Answer: a
Explanation: The hexadecimal number system comprises of only 15 symbols: 10 digits and 5 symbols. Hence, three bits ($2^4 = 16$) are sufficient to represent any hexadecimal number in the binary format.

417. The binary number 1110 in hexadecimal format is
_____
a) 6
b) E
c) 14
d) 15

Answer: b
Explanation: Certain binary to hexadecimal representations are :
1010=A
1011=B
1100=C
1101=D
1110=E
1111=F.

DIWAKAR EDUCATION HUB