

Name : Simran Jeet Gill.

8DP = 500122195

Roll no. : R214230847

BATCH 29.

### Assignment 2

a) Differentiate the following :-

#### COMBINATIONAL CIRCUIT

#### SEQUENTIAL CIRCUIT

1. It is a type of circuit ! it is a type of circuit that generates an output by relying on the input it requires at that instant.
2. It is very less complex.
3. Logic gates form the building block.
4. Requires no feedback.
5. Used for Boolean & Arithmetic operation.
2. It is very complex.
3. Flip flop form the building block.
4. It requires feedback.
5. Used for Storing data.

#### b) SYNCHRONOUS COUNTER

#### ASYNCHRONOUS COUNTER

1. All flip-flops are triggered by same clock pulse.
2. It is fast.
3. More Complex.
4. more reliable at higher speeds.
5. No Big Ripple Effect
1. Each flip-flop is triggered by the previous one.
2. It is slow.
3. less Complex.
4. Not reliable at higher speeds.
5. Ripple Effect.

## C) COUNTER

## REGISTER

- |                                |   |
|--------------------------------|---|
| 1. It can't hold data.         | 1. It can hold data.                      |
| 2. follows sequence of states. | 2. does not follow any specific sequence. |
| 3. does not have same clock.   | 3. has some clock.                        |
| 4. All counter are registers.  | 4. All registers are not counter.         |
| 5. Stage is predefined.        | 5. Stage is not predefined.               |
- 
- |  |                                      |
|--|--------------------------------------|
| a) SIPO                                      | PIPO                                 |
| 1. Has single data input                     | 1. Has Multiple inputs.              |
| 2. It is shifted through register in serial. | 2. It is transferred in parallel.    |
| 3. It is suitable for serial data.           | 3. It is suitable for parallel data. |
| 4. Commonly used in serial communication.    | 4. Used in parallel.                 |
| 5. Strict timing.                            | 5. Simpler timing.                   |

Q2. Explain the following flip-flop in detail:-

a. S-R flip flop:-

It is a flip flop with 2 inputs, one in S + one in R. S here stands for Set & R stands for Reset. Set initially indicates set the flip-flop, which means output 1 + Reset indicates resetting the flip-flop which means output is 0.

10) T flip-flop:-

T stands for Toggle. T flip flop used to toggle. It's output depending upon the input. It indicates that one bit will be flipped either from 1 to 0 or from 0 to 1.

Q3. Design a 4-bit synchronous down counter that counts through all states from 1111 down to 0000.

| $Q_p$ | $Q_c$ | $Q_B$ | $Q_A$ | $Q_p^+$ | $Q_c^+$ | $Q_B^+$ | $Q_A^+$ | $J_0 K_0$ | $J_1 K_1$ | $J_2 K_2$ | $J_3 K_3$ |
|-------|-------|-------|-------|---------|---------|---------|---------|-----------|-----------|-----------|-----------|
| 1     | 1     | 1     | 1     | 1       | 0       | 1       | 0       | d 0       | d 0       | d 0       | d 1       |
| 1     | 1     | 1     | 0     | 1       | 1       | 1       | 0       | d 0       | d 0       | d 1       | 1 d       |
| 1     | 1     | 0     | 1     | 1       | 1       | 0       | 0       | d 0       | d 0       | d 0       | d 1       |
| 1     | 1     | 0     | 0     | 1       | 0       | 1       | 0       | d 0       | d 1       | 1 d       | 1 d       |
| 1     | 0     | 1     | 1     | 0       | 1       | 0       | 1       | d 0       | d 1       | d 0       | d 1       |
| 1     | 0     | 1     | 0     | 1       | 0       | 0       | 1       | d 0       | d 1       | d 1       | 1 d       |
| 1     | 0     | 0     | 1     | 0       | 1       | 1       | 0       | d 0       | d 1       | 0 d       | d 1       |
| 1     | 0     | 0     | 0     | 0       | 1       | 1       | 1       | d 0       | d 0       | 1 d       | d 1       |
| 0     | 1     | 1     | 1     | 0       | 1       | 1       | 0       | d 1       | d 1       | d 0       | 1 d       |
| 0     | 1     | 1     | 0     | 0       | 1       | 0       | 1       | 0         | d 1       | d 1       | 1 d       |
| 0     | 1     | 0     | 1     | 0       | 0       | 1       | 0       | 0         | d 1       | 0 d       | d 1       |
| 0     | 1     | 0     | 0     | 0       | 0       | 0       | 0       | 0         | d 1       | 1 d       | 1 d       |
| 0     | 0     | 1     | 1     | 0       | 0       | 0       | 1       | 0         | 0         | d 0       | d 0       |
| 0     | 0     | 1     | 0     | 0       | 0       | 0       | 0       | 0         | d 0       | d 0       | d 1       |
| 0     | 0     | 0     | 1     | 0       | 0       | 0       | 0       | 0         | d 0       | d 0       | d 1       |
| 0     | 0     | 0     | 0     | 0       | 0       | 0       | 0       | 0         | d 0       | 1 d       | d 1       |
| 0     | 0     | 0     | 0     | 0       | 0       | 0       | 0       | 0         | d 1       | d 1       | d 1       |

Q4. Design & implement J-K flip-flop using S-R flip flop.

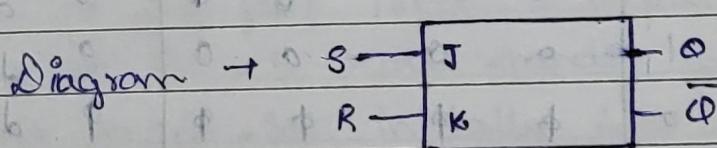
$s \quad R \quad Q_n \quad Q_{n+1} \quad J \quad K$

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | x |
| 0 | 0 | 1 | 1 | x | 0 |
| 0 | 1 | 0 | 0 | 0 | x |
| 0 | 1 | 1 | 1 | x | 1 |
| 1 | 0 | 0 | 0 | 1 | x |
| 1 | 0 | 1 | 1 | x | 0 |
| 1 | 1 | 0 | 0 | x | 1 |

$$J(1) = \sum_{\text{odd}}(4) + \alpha(1, 3, 5, 6, 7)$$

$$R = \sum_m (3) + \alpha (0, 2, 4, 6, 7)$$

| for k | s | 00 | 01 | 11 | 10 | 0 |
|-------|---|----|----|----|----|---|
| 0     | 0 | 1  | 0  | 0  | 0  | 0 |
| 1     | 1 | 0  | 0  | 0  | 0  | 1 |



Q5 Design a 4-bit self-correcting shift counter using either JK or D flip flop.

## State Table (RPS - NS)

$$q_A + q_B + q_C + q_D = Q_A + Q_B + Q_C + Q_D$$

|   |     |    |    |    |   |    |   |   |   |   |
|---|-----|----|----|----|---|----|---|---|---|---|
| b   | bob | o  | ob | bo | b | 1b | o | o | o | o |
| b   | bob | o  | o  | b1 | b | 1b | o | 1 | o | o |
| b   | bob | bo | b  | bo | b | ob | o | o | o | 1 |
| b   | bob | bo | b  | b1 | b | db | d | 1 | d | 1 |
| o   | 1   | o  | o  | o  | o | o  | o | 1 | o | o |
| (81, P1, S1, S1, 11, o1, P, 3, 3, d, d, d, d)     | o   | 1  | o  | 1  | o | 1  | o | 1 | o | o |
| (0, G, P1, S1, S1, 11, o1, P, 3, 3, d, d, d, d)   | o   | 1  | o  | 1  | o | 1  | o | 1 | o | o |
| (-o, P1, 11, S1, S1, 11, d1, 8, 5, 3, 2, d, P, d) | o   | 1  | o  | 1  | o | 1  | o | 1 | o | o |
| 81, 11, S1, 9, S1, o1, S1, P, P, d, 2, 31, 8, o)  | o   | 1  | o  | 1  | o | 1  | o | 1 | o | o |
| 1   | o   | o  | 1  | d  | d | d  | d | d | d | d |
| 1   | o   | 1  | o  | d  | d | d  | d | d | d | d |
| 1   | o   | 1  | d  | d  | d | d  | d | d | d | d |
| 1   | 1   | o  | o  | d  | d | d  | d | d | d | d |
| 1   | 1   | o  | 1  | d  | d | d  | d | d | d | d |
| 1   | 1   | o  | 1  | d  | d | d  | d | d | d | d |

## Excitation table

| PS    |       |       |           | NS.   |       |       |       | Excitation. |       |       |       |
|-------|-------|-------|-----------|-------|-------|-------|-------|-------------|-------|-------|-------|
| $q_A$ | $q_B$ | $q_C$ | $q_{D11}$ | $q_A$ | $q_B$ | $q_C$ | $q_D$ | $D_A$       | $D_B$ | $D_C$ | $D_D$ |
| 0     | 0     | 0     | 0b        | 1     | 0     | 0     | 0     | 1           | 0     | 0     | 0     |
| 0     | 0     | 0     | 1b        | 1     | 0     | 0     | 100   | 1           | 0     | 0     | 0     |
| 0     | 0     | 1b    | 0b        | 0     | 0     | 0     | 11    | 0           | 0     | 0     | 1     |
| 0     | 0     | 1b    | 1b        | d     | d     | d     | d     | d           | d     | d     | d     |
| 0     | 1     | 0     | 0         | 0     | 0     | 1     | 0     | 0           | 0     | 1     | 0     |

$$D_B = \sum_m (0, 1, 3, 4, 5, 6, 9, 10, 11, 12, 13, 14, 15)$$

$$DB = \sum m (8, 3, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15)$$

$$P_C = \sum_m (3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15)$$

$$D_F = \{2, 3, 18, 6, 7, 9, 10, 11, 12, 13, 14, 15\}$$

we b dont b take for DA

don't care conditions for PA

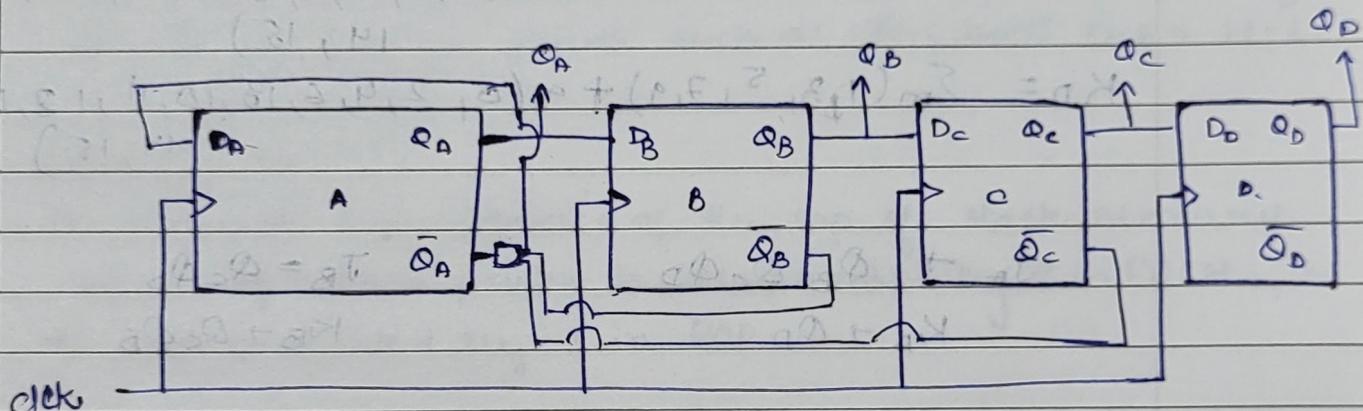
$$\begin{array}{cccc|ccccc}
 & b & b & b & b & 0 & 0 & 1 \\
 \rightarrow & b & b & b & b & 00 & 101 & 110 & 10 \\
 & b & b & b & b & 000 & C1 & b1 & d0 \\
 & b & b & b & b & 010 & 1d & d1 & d1 \\
 & & & & & 11 & d & d & d \\
 & & & & & 00 & 1 & d & d
 \end{array} = \bar{q}_A \bar{q}_B \bar{q}_C$$

| for DB = qA |   | qB | 00 | 01 | 11 | 10 | 0P | 1P | 0P | 1P | 0P | 1P |
|-------------|---|----|----|----|----|----|----|----|----|----|----|----|
| 0           | 0 | 1  | 00 | 0  | 1  | d  | 0  | 0  | 0  | 0  | 0  | 0  |
| 0           | 0 | 1  | 01 | 0  | d  | d  | 1  | d  | 0  | 0  | 0  | 0  |
| 1           | 0 | 0  | 10 | 11 | d  | d  | d  | d  | 1  | 0  | 0  | 0  |
| b           | b | b  | b  | 10 | d  | d  | d  | 1  | d  | 0  | 0  | 0  |
| 1           | 0 | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1  | 0  |

|    | 00 | 01 | 11 | 10 | 0 | 1 |
|----|----|----|----|----|---|---|
| 00 | d  | d  | d  | d  | d | d |
| 01 | d  | d  | d  | d  | d | d |
| 11 | d  | d  | d  | d  | d | d |
| 10 | d  | d  | d  | d  | d | d |

$$\text{for } DD = \begin{pmatrix} q_A & q_B \\ 0 & 0 \end{pmatrix} \quad \begin{pmatrix} q_A & q_B \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

| QB | Qc | Qp | 00 | 01 | 11 | 10 | al               |
|----|----|----|----|----|----|----|------------------|
| 00 |    |    |    | d  | !  |    | ?                |
| 01 |    | d  |    | d  | d  |    | $\Leftarrow q_c$ |
| 11 | d  | d  |    | d  | d  |    | ?                |
| 10 |    | d  | d  | d  |    |    | ?                |



Q6 Design a synchronous BCD counter using either J-K or D flip flop.

| PS | NS. | $\oplus_A$ | $\oplus_B$ | $\oplus_C$ | $\oplus_D$ | J <sub>A</sub> K <sub>A</sub> | J <sub>B</sub> K <sub>B</sub> | J <sub>C</sub> K <sub>C</sub> | J <sub>D</sub> K <sub>D</sub> |
|----|-----|------------|------------|------------|------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| 0  | 0   | 0          | 0          | 0          | 1          | od                            | od                            | od                            | id                            |
| 0  | 0   | 0          | 1          | 0          | 0          | od                            | od                            | id                            | d1                            |
| 0  | 0   | 1          | 0          | 0          | 1          | od                            | od                            | -do                           | id                            |
| 0  | 0   | 1          | 1          | 0          | 1          | od                            | id                            | d1                            | d1                            |
| 0  | 1   | 0          | 0          | 0          | 1          | od                            | do                            | od                            | id                            |
| 0  | 1   | 0          | 1          | 0          | 1          | od                            | do                            | id                            | d1                            |
| 0  | 1   | 1          | 0          | 0          | 1          | od                            | do                            | do                            | cd                            |
| 0  | 1   | 1          | 1          | 1          | 0          | id                            | d1                            | d1                            | d1                            |

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

$$J_A = \sum_m(7) + \alpha(8, 9, 10, 11, 12, 13, 14, 15)$$

$$K_A = \sum_m(9) + \alpha(0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15)$$

$$J_B = \sum_m(8) + \alpha(4, 5, 6, 7, 10, 11, 12, 13, 14, 15)$$

$$K_B = \sum_m(7) + \alpha(0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$\Sigma = \sum_m(1, 5) + \alpha(2, 3, 6, 7, 10, 11, 12, 13, 14, 15)$$

$$K_C = \sum_m(3, 7) + \alpha(0, 1, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$J_D = \sum_m(0, 2, 4, 6, 8) + \alpha(0, 1, 2, 4, 6, 8, 10, 11, 12, 13, 14, 15)$$

$$K_D = \sum_m(1, 3, 5, 7, 9) + \alpha(0, 1, 2, 4, 6, 8, 10, 11, 12, 13, 14, 15)$$

$$J_A + Q_B \cdot Q_C \cdot Q_D$$

$$K_D \rightarrow Q_D$$

$$J_B = Q_C \cdot Q_D$$

$$K_B \rightarrow Q_C \cdot Q_D$$

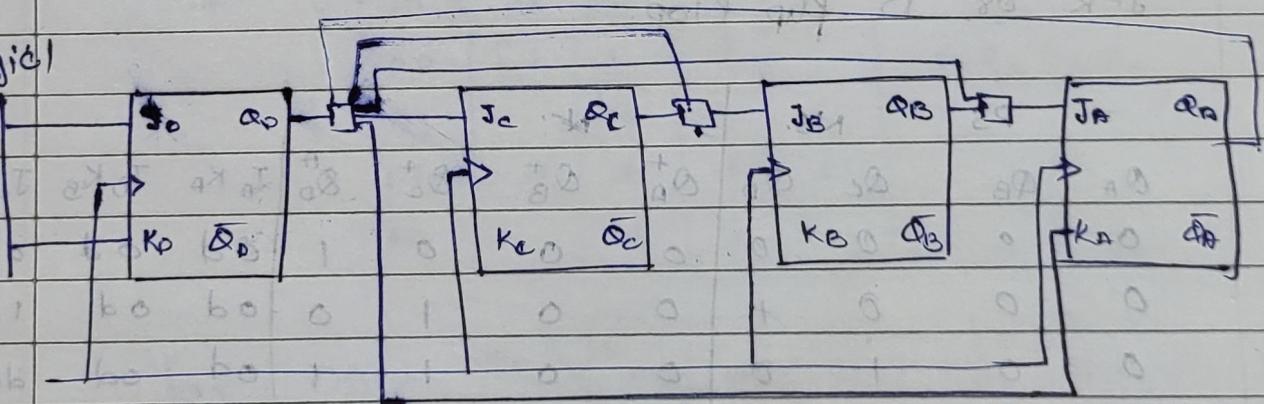
$$J_C + \bar{Q}_A \cdot Q_D$$

$$K_C \rightarrow \bar{Q}_D$$

$$J_D = 1$$

$$K_D = 1$$

logic



|    |    |    |    |   |   |   |   |   |   |   |   |
|----|----|----|----|---|---|---|---|---|---|---|---|
| 1b | 1b | b1 | b0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| b1 | b0 | 0b | b0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1b | b1 | 0b | b0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| b1 | 0b | 0b | b0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1b | 1b | 1b | b1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |

Q7. Discuss the significance of following registers of 8085 microprocessor.

- Accumulator: It is register A. It is used for arithmetic & logical operations as well as fast input / output operations. It is also used as a temporary storage location for data.
- General Purpose Registers: B, C, D, E, H, L. Each register can hold 8-bit data. It can be also used to work in pairs to hold 16-bit data. They can work in pairs such as B-C, C-D, D-E & H-L.
- Stack Pointer: It stores the address of the top of stack memory. Value of stack pointer is decremented by 1 in PUSH & incremented by 1 in POP.
- Program Counter: It is used to point the memory address from which the next byte is to be fetched.
- Instruction Register: It is used to receive the 8-bit data opcode portion of an instruction.
- Temporary Register: It stores the information internally. It can read and write more than once in a single instruction.
- Status Flags: It generally reflects the status of arithmetic & logic.

operations \* S (sign flag): Sign flag is set to 1 if result of  $07$  is one otherwise it reset.

Q8. what do you understand by timing diagram?

Draw & explain the timing diagram of

\* fetch opcode:

It is the graphical representation of input & output signals as function of time.

a) fetch opcode:

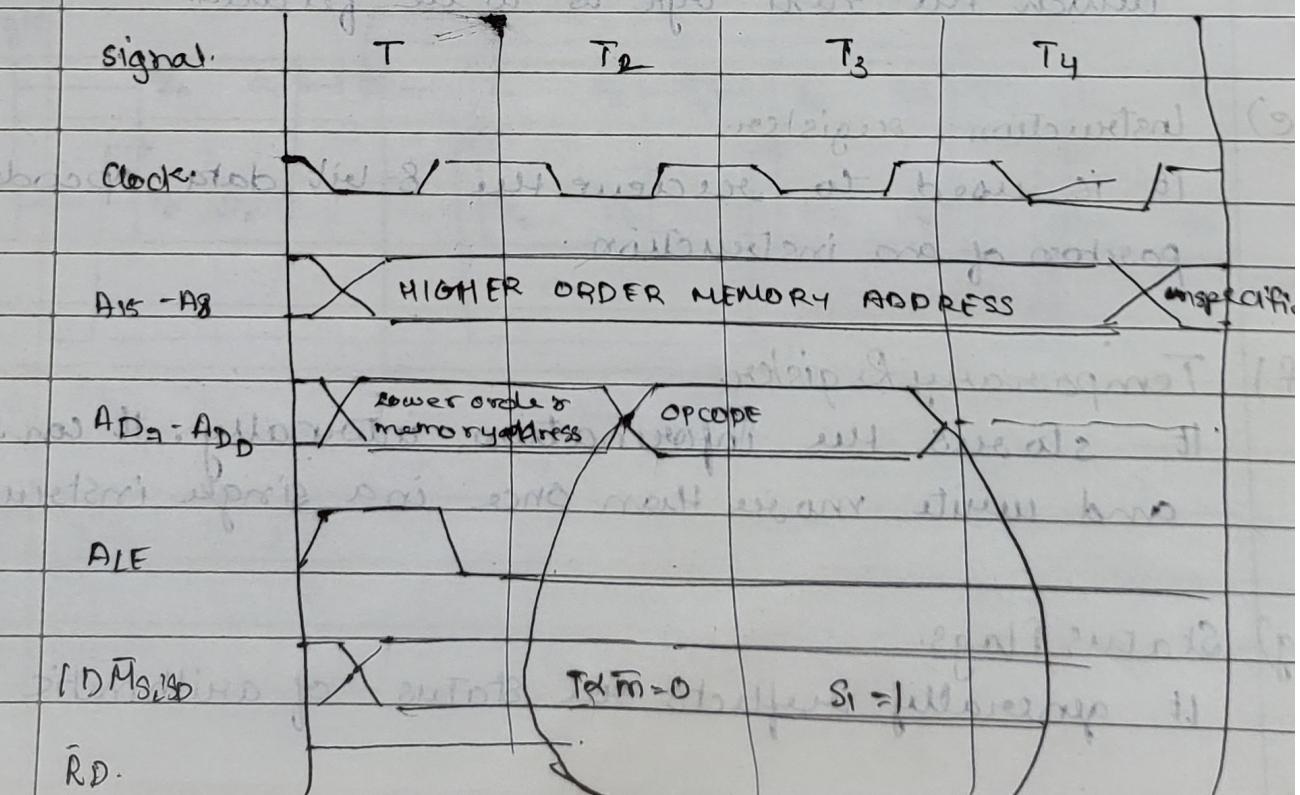
a process in computer architecture where processor retrieves data from memory. Memory is the main storage for data & instruction

working in a system.

Hardwiring is nothing but basic logic

b) Memory Write Operation:

It transfers data the address of desired word to address lines transfers data bits to be stored in memory to data input lines.



Q9. What do you understand by opcode + operand in an assembly language program. Explain the following types of instructions with the help of examples.

a) One Byte Instruction.

A binary or a hexadecimal number that represents a specific instruction for a processor to perform.

**OPERAND** → A value on which the instruction named by mnemonic operates. The operand may be processor register, memory address, a literal constant, or a label.

a) One Byte Instruction :-

It includes opcode and an operand in the same byte. Operands are internal registers & are in the instruction in form of codes. e.g. MOV C ,A, RAL, ADDB.

b) Two Byte Instruction:-

Source operand is a data byte & immediately following the opcode if an 8 bit one is present.

c) Three Byte instruction:-

If the instruction is of three bytes then it requires 3 machine cycles.

Q10. Explain

a) Binary weighted D/A converter.

It is a type of digital to analog converter that converts digital to equivalent analog output signal by using a network of binary weighted resistors.

b)

Successive Approximation A/D converter:

It is a type of analog to digital that converts a continuous analog wave form into a discrete digital representation using a binary staircase to search through all possible quantization levels before finally converging upon a digital output for each conversion.