DIGITAL ELECTRONICS AND MICROPROCESSOR

[TH-03]

5TH SEM ELECTRICAL ENGG.

PREPARED BY :-**B.D. PANDA**

CHAPTER-1

Basics of Digital Electronics

Short questions

1.Convert the decimal no.$(1000)_{10}$ into binary.

2	1000	
2	500	0
2	250	0
2	125	0
2	62	1
2	31	0
2	15	1
2	7	1
2	3	1
	1	1

Ans.($1111101000)_2$

2.Convert (10110101) from binary to gray code.

1 0 1 1 0 1 0 1 (binary)

↓

1 1 1 0 1 1 1 1 (gray)

3. Perform 2's complement substraction of 1000011-1010111.

$(67)_{10} – (87)_{10} = (-20)_{10}$

Convert substrate part into 2's complement

1010111 —1's→ 0101000 —2's comp→

\+ 1

0101001

Then add both manuent and substrant

1000011

$$\begin{array}{r} + \quad 0101001 \\ \hline 1101100 \end{array}$$

Here there is no carry ,if there is no carry then substrate 1 from the result then recomplement it.

$$\begin{array}{r} 1101100 \\ - \quad 1 \\ \hline 1101011 \end{array}$$

Then recomplement of this $(0010100)_{10}$

4.What is the difference between weighted and non weighted binary code.

The weighted codes are those that obey the position weighting principle ,which states that the position of each number represent a specific weight .

The non-weighted codes are not positionally weighted .In other words codes that are not assigned with any weight to each digit position.

5.Convert the binary no.$(10110111.1101)_2$ to decimal.

$(10110111.1101)_2$

$1\times2^7+1\times2^5+1\times2^4+1\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-2}+1\times2^{-4}$

128+32+16+4+2+1+1/2+1/4+1/16

183.81

6.What are the application of gray code.

- Used as counter
- Used for asynchronous fifo's adress pointer
- Help to reduce the digital noise
- High speed decode circuit
- Also use in computers to address program memory .

7.What is parity bit.

It is an error detection scheme used to detect a change in the value of a bits (0 or 1) during transmission to detect error .

8.What is the base or radix of a number system.

The Radix of a number system is defined as the number of different digits which can occur in each position in the number system for example :Decimal number system has a base or radix of 10

9.Convert $(FADE)_{16}$ to binary and octal.

In the binary form

FADE is $(1111\ \ 1010\ \ 1101\ \ 1110)_2$

In octal form

001 111 101 011 011 110

1 7 5 3 3 6

Answer is $(175336)_8$

10.Convert $(7743)_d$ to binary and hexadecimal.

For binary

7 7 4 3

111 111 100 011

In the binary form the answer is $(111\ \ 111\ \ 100\ \ 011)_2$

For hexadecimal

First octal is converted to binary then to hexadecimal .

7 7 4 3

111 111 100 011

15 14 3

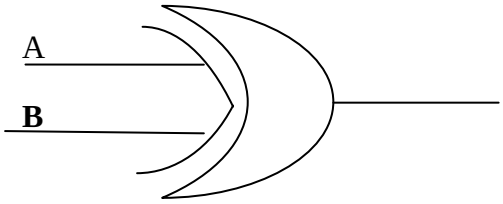
In the hexadecimal form the answer is $(FE3)_{16}$

11. Write down the truth table of 2 input ex-or gate.

A	B	C=**AB'+A'B**
0	0	0
0	1	1
1	0	1
1	1	0

12.Draw the symbol ,truth table and expression for ex-or and AND gates.

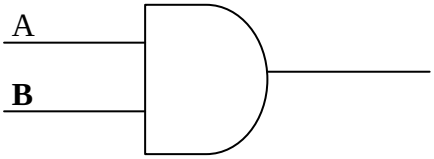
EX-OR gate



Truth table

A	**B**	**Y=AB'+A'B**
0	0	0
0	1	1
1	0	1
1	1	0

AND gate



Truth table

A	**B**	**Y=AB**
0	0	0
0	1	0
1	0	0
1	1	1

13.Define demorgan's law.

This law states that the complement of any expression can be obtained by replacing each variable and element with its complement and changing OR operators with AND operators and AND operators to OR operators .These can be expressed

A) $\overline{X + Y} = \overline{X} \cdot \overline{Y}$

B) $\overline{X \cdot Y} = \overline{X} + \overline{Y}$

5 marks questions

1.What are the difference between 1's complement and 2's complement ?Substrat $(39)_{10}$ and $(25)_{10}$ using 2's complement.

1's complement of a binary number can be find out by complementing 1 to 0 and 0 to 1 means (in place of 0 we put 1 and in place of 1 we put 0)

2's complement of the number can be findout by adding 1 to the 1's complement of the given number .

39-25=?

2	39	
2	19	1

2	25	
2	12	1

2	9	1
2	4	1
2	2	0
	1	0

$(100111)_2$

2	6	0
2	3	0
	1	1

$(11001)_2$

Convert equal bi39----------100111

25----------011001

Convert substrant into 2's complement

011001----------100110

\+ 1

100111

Then add both

100111

100111

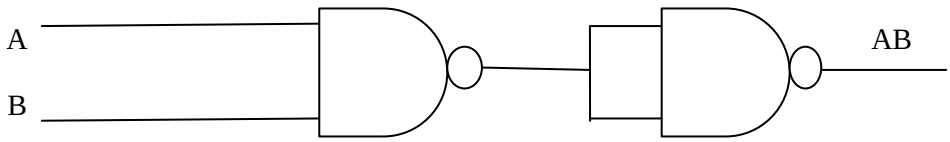
1001110

Ans. 001110

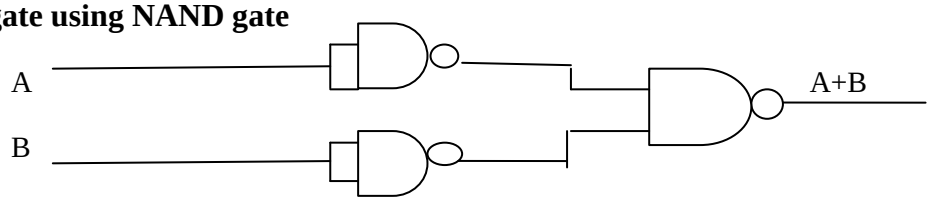
2. Which gates are referred to as universal gates and why ?How other gates can be realized using NAND gates.

NAND gate and NOR gate are known as universal logic gates .All logic gates are implemented using NAND and NOR gate only.

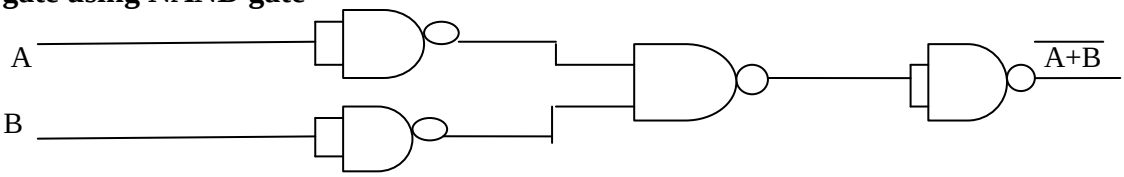
AND gate using NAND gate



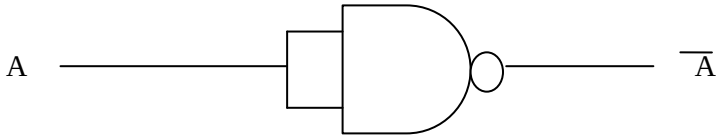
OR gate using NAND gate



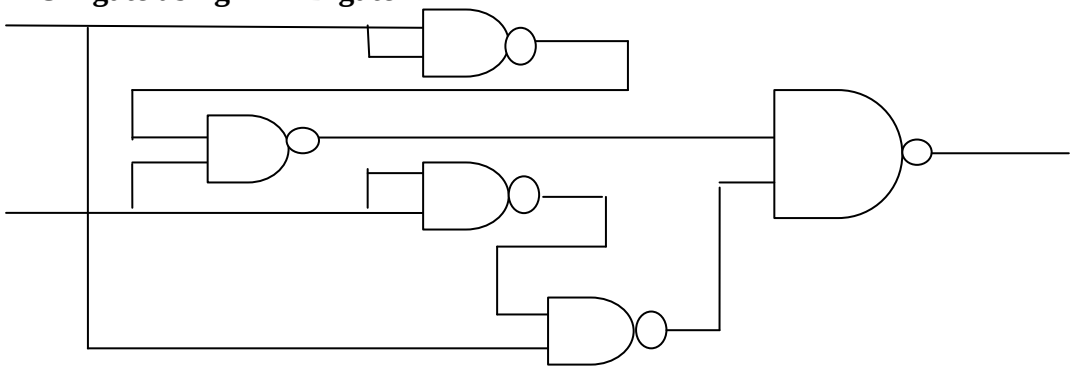
NOR gate using NAND gate



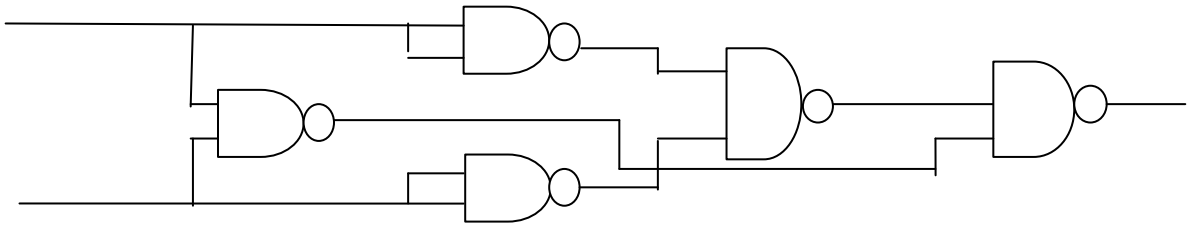
NOT gate using NAND gate



EX-OR gate using NAND gate



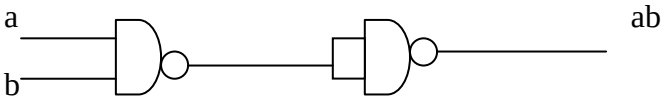
EX-NOR gate using NAND gate



3. Simplify the Boolean expression $a[b+c(\overline{ab+ac})]$

$A[b+c(\overline{ab+ac})] = a[b+c\overline{(ab}\times\overline{ac})] = a[b+c(\overline{a}+\overline{b})(\overline{a}+\overline{c}) = a[b+c(\overline{a}\times\overline{a} + \overline{a}\times\overline{b} + \overline{a}\times\overline{c} + \overline{b}\times\overline{c})]$

$A[b+c(\overline{a} + \overline{a}\ \overline{b} + \overline{a}\ \overline{c} + \overline{b}\ \overline{c})] = a[b + c\overline{a}(1 + \overline{b} + \overline{c}) + \overline{b}\ \overline{c}\ c] = a[b + \overline{a}\ c] = ab$



4. Simplify the expression.

$Y=\overline{(AB+\overline{C})(\overline{A+b}+C)}$

$\overline{(AB+\overline{C})} + \overline{(\overline{A+B}+C)}$ $=[\overline{AB} \times \overline{C}] + \{\overline{\overline{A+B}}\} \times \overline{C}$

$=[\overline{AB} \times C] + (A+B) \times \overline{C}$

$=[(\overline{A}+\overline{B})C] + (A+B) \times \overline{C}$

$= \overline{A}C + \overline{B}C + A\overline{C} + B\overline{C}$

5.Simplify the following expression using k-map.

$Y(A,B,C,D)=\sum(1,5,10,11,12,13,15)$

	1		
	1		
1	1	1	
		1	1

6. State and prove demorgan's theorem .Explain with relevant diagrams, Boolean expressions and functional tables.

This theorem states that the complement of any expression can be obtained by replacing each variable and the operators of or is changed to AND and via-versa.

$\overline{X + Y} = \overline{X} \cdot \overline{Y}$

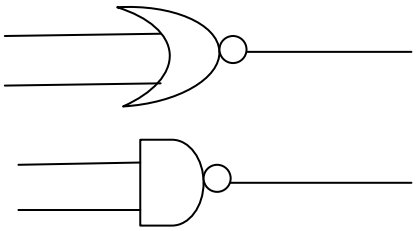
$\overline{X \cdot Y} = \overline{X} + \overline{Y}$

Verification or truth table for de morgans theorem

	0	1	2	3
X	0	0	1	1
Y	0	1	0	1
X	1	1	0	0
Y	1	1	0	0
XY	1	0	0	0
X+Y	0	1	1	1
X+Y	1	0	0	0

From truth table X+Y=X .Y

Drawing the logic diagram



X	Y	X+Y
0	0	1
0	1	0

1	0	0
1	1	0

2.Define SOP and POS term .Obtain the canonical SOP form of the function .

$Y(A,B,C)=A+B\bar{C}$ and draw the truth table.

Sop :- When the product terms goes for logical OR operation OR operation then the final expression is in sum of product expression or sop form .

Ex: $AB+A\bar{B}C+ABC$

Pos :- When two or more sum terms goes for logical AND operation then the output of the AND gate is in pos form .

Ex $(A+\bar{B})(\bar{A}+\bar{B}+C)$

$Y(A,B,C)=A+\bar{B}\bar{C}=A(B+\bar{B})(C+\bar{C})+\bar{B}\bar{C}(A+\bar{A})$

$=(AB+A\bar{B})(C+\bar{C})+A\bar{B}\bar{C}+\bar{A}\bar{B}\bar{C}$

$=AB(C+\bar{C})+A\bar{B}(C+\bar{C})+\bar{A}\bar{B}\bar{C}+\bar{A}\bar{B}\bar{C}$

$=ABC+AB\bar{C}+A\bar{B}C+A\bar{B}\bar{C}+\bar{A}\bar{B}\bar{C}+\bar{A}\bar{B}\bar{C}$

$=M_7+M_6+M_5+M_4+M_4+M_0$

$=M_0+M_4+M_5+M_6+M_7$

$=\sum_m(0,4,5,6,7)$

Truth table :-

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Chapter 2

COMBINATIONAL LOGIC CIRCUIT

2 marks

1.What is combinational logic circuit?

Ans:-it is a logic circuit whose output depands only on the present input.here clock pulse is not used.

Eax:-adder, subtractor, multiplexer, demultiplexer, encoder, decoder etc

1.What is an encoder and where it is used.

An encoder is a device whose inputs are decimal digits and outputs are coded representation of input .It has 2^n inputs and 1 output .It has n of control signal .

5 marks

1. Design a 4:2 encoder with neat circuit diagram.

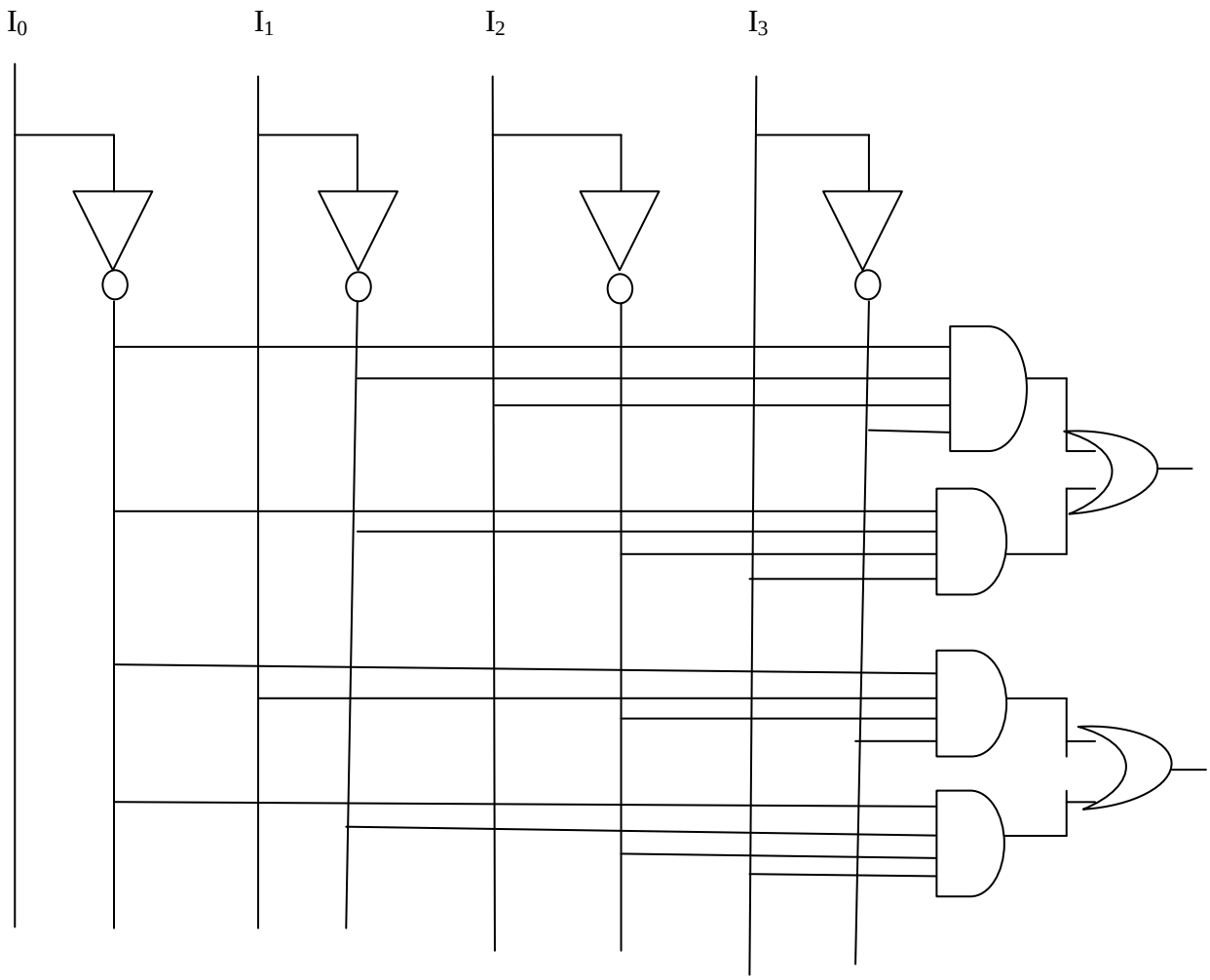
Input $=2^n=4=$input

N=output

Input----------I_0 ,I_1 ,I_2,I_3

Output--------D_0 ,D_1

Input				output	
I_0	I_1	I_2	I_3	D_0	D_1
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1



2.Describe the operation of full subtractors with the help of truth table and circuit diagram.

Ans:-A full subtractor is a circuit that subtracts two bits along with the borrow.the truth table and logic diagram is given below

Truth table

A	B	C	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0

1	1	0	0	0
1	1	1	1	1

K'map for the Difference is=$\sum_m$(1 ,2 ,4 ,7)

	1		1
1		1	

$X\bar{Y}\bar{Z}+\bar{X}\,\bar{Y}Z+XYZ+\bar{X}Y\bar{Z}$

$X\bar{Y}\bar{Z}+XYZ+\bar{X}\bar{Y}Z+\bar{X}YZ$

$X(\bar{Y}\bar{Z}+YZ)+\bar{X}(\bar{Y}Z+\bar{Y}Z)$

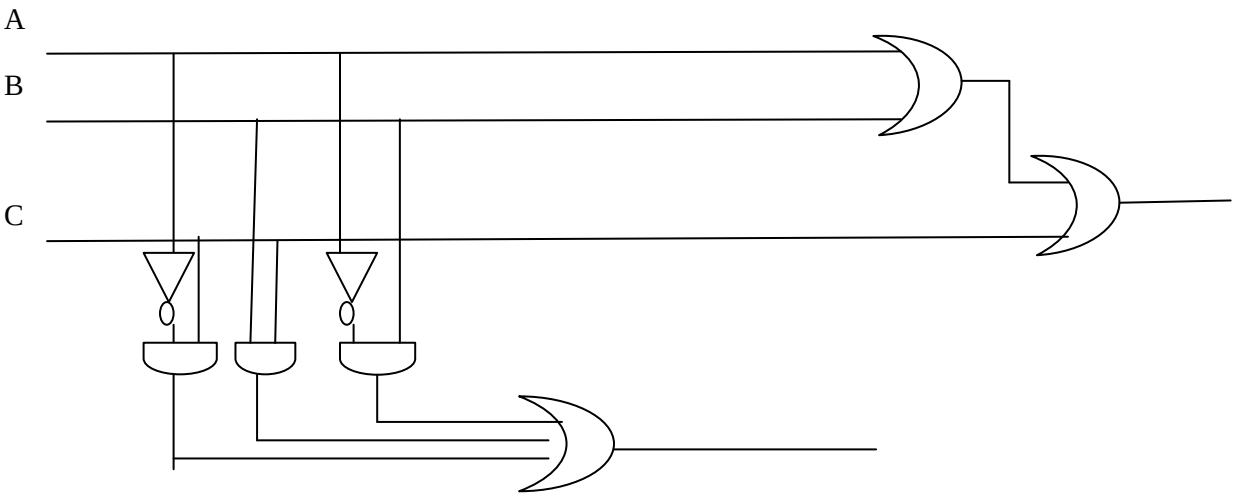
$X(\overline{y+z})+x(y+z)$

X+Y+Z

K'map for Borrow is=$\sum$m(1 ,2 ,3 ,7)

	1	1	1
		1	

XZ +YZ +XZ



1. With neat circuit diagram explain the function of 1:4 de-mux & 4:1 mux.

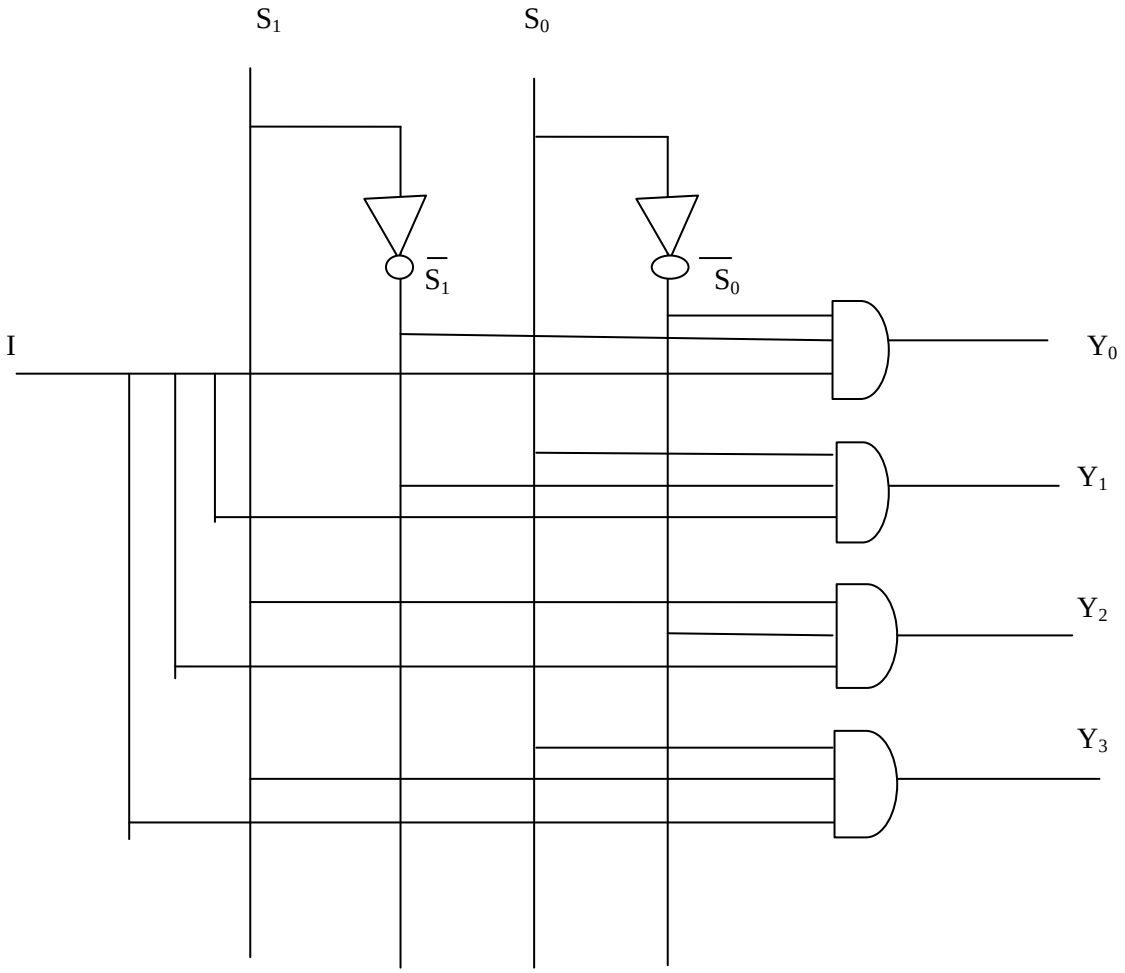
Ans:-1:4 DEMUX

No. of input =1 No. 0f selection line =n=2 No. of output $=2^2=4$

Truth table :-

Input	selection lines		outputs
	S_1	S_0	
I	0	0	0 0 0 1
I	0	1	0 0 1 0
I	1	0	0 1 0 0
I	1	1	1 0 0 0

$Y_0 = \overline{S_0}\,\overline{S_1}\,I$, $Y = S_0\,\overline{S_1}\,I$, $Y = \overline{S_0}\,S_1\,I$, $Y = S_0\,S_1\,I$



4:1 MUX

4:1 MUX = 2^n :1 =2^2 :1mux

No. of inputs =2 =4(D_0 D_1 D_2 D_3)

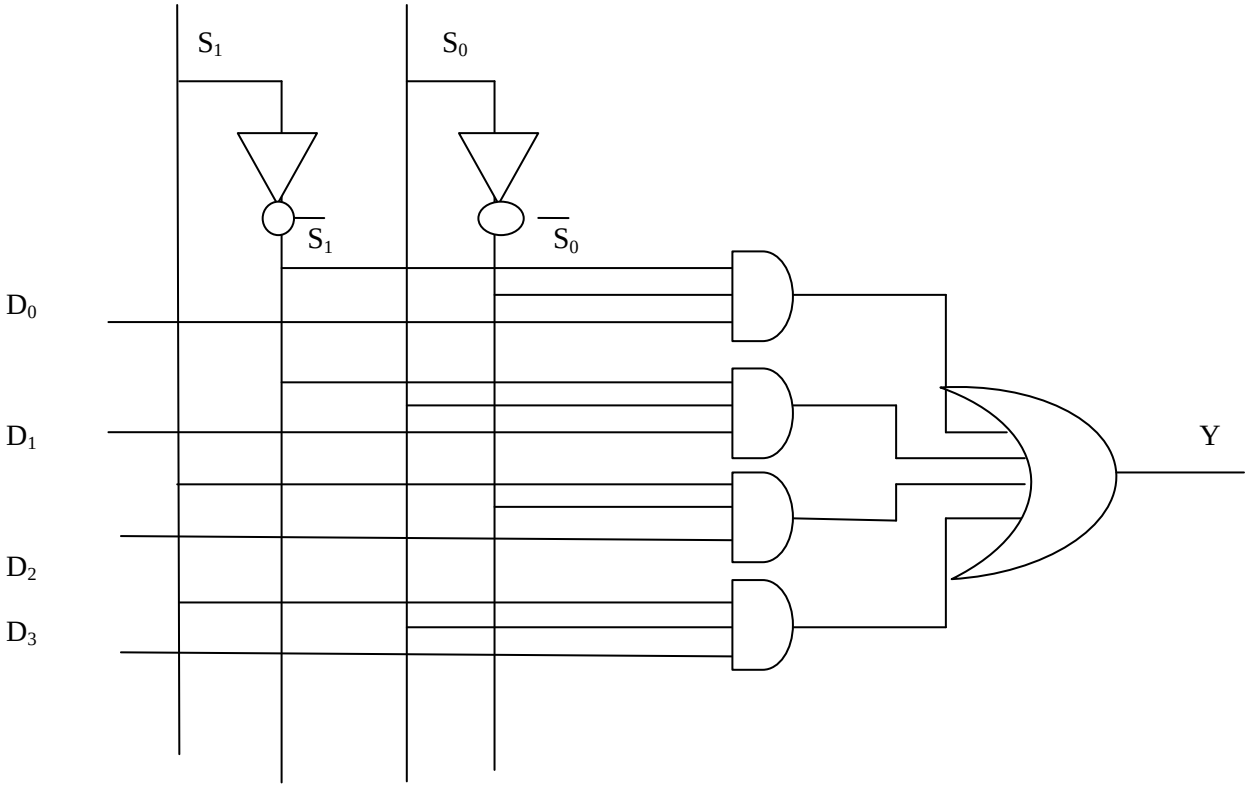
No. of selection line =n =2(S_1 , S_0)

No. of output =1 (Y)

Truth table :-

Data	select inputs	output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$Y= \overline{S}_1 \overline{S}_0 D_0 + \overline{S}_1 S_0 D_1 + S_1 \overline{S}_0 D_2 + S_1 S_0 D_3$



3. Design a full adder circuit and implement using NAND gate.

It will perform the addition operation of three binary bits

Truth table

X	Y	Z	SUM	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$Sum(x,y,z)=\sum_m(1,2,4,7)$

	1		1
1		1	

$\bar{X}\bar{Y}Z+\bar{X}Y\bar{Z}+XYZ+X\bar{Y}Z$

$=X\overline{YZ}+XYZ+\bar{X}\bar{Y}Z+\bar{X}YZ$

$=X(\bar{Y}\bar{Z}+YZ)+\bar{X}(\bar{Y}Z+Y\bar{Z})$

$=X(\overline{Y \oplus Z})+\bar{X}(Y \oplus Z)$

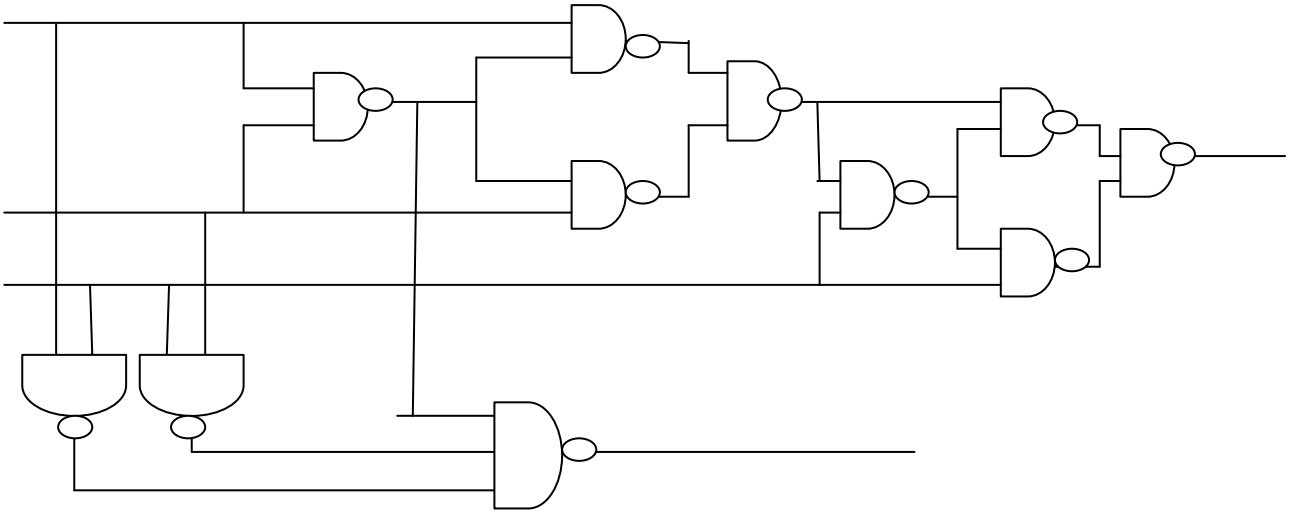
$=X \oplus Y \oplus Z$

$C(X,Y,Z)=\sum_M(3,5,6,7)$

		1	
	1	1	1

$XZ+YZ+XY$

Logic diagram



Chapter 3

SEQUENTIAL LOGIC CIRCUIT

2MARKS

1.What is sequential logic circuit.

Ans:-it is a logic circuit whose output not only depands on the present input but also the past output.For this perpose a memory element called flip flop is used**.**

2.what is the meaning of level triggering and edge triggering.

Ans:-Level triggering

It is a type of triggering that allows a circuit to become active when the clock pulse is on a particular level.i.e either ON or OFF level.

Edge triggering

It is a type of triggering that allows a circuit to become active at the positive edge or negative edge at the clock pulse

3. what is a flip flop.

Ans:-flip flop is a memory element that stores one digital bit.it can have only two states i.e either 1 or 0.flip flop can be obtained by using NAND or NOR gates.it is also called bistable multivibrator

4.Define modulous of a counter.

Modulous of a counter is defined as equal to the number of states through which the counter progress .The maximum possible number of states is equal to 2^n .Where n=number of flip flops in the counter.

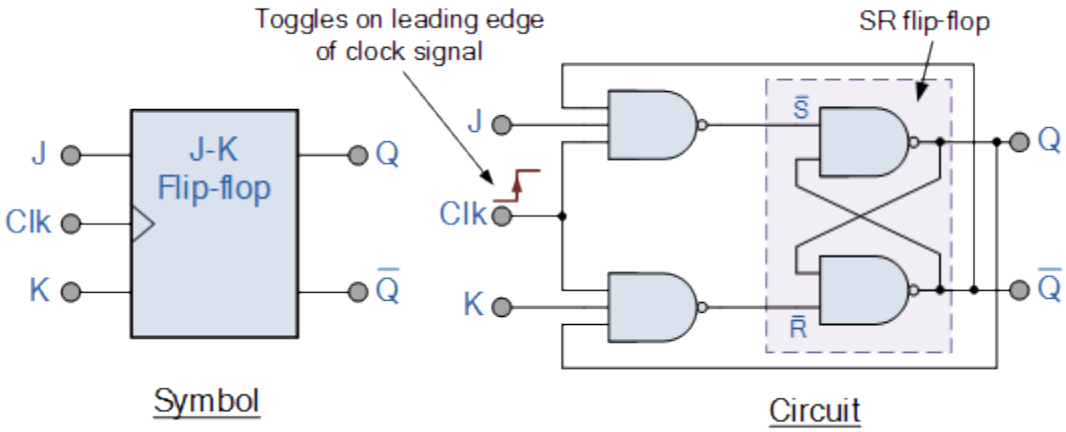
5 marks

1. Distinguish between combinational & sequential logic circuit.

Combinational logic	Sequential logic
-In which the output depend on the present input only is known as combinational logic .	-In this case the output depend on present input as well as past output .
-In this case no feedback path is there .	–In this case feedback path is there
--Combinational circuit has no memory	-Sequencial circuit has memory
- -Here we are not using clock pulse.	–In sequencial we use clock pulse.

2.Level clocked JK flip flop using S-R flip flop with truth table.

Ans:- The JK flip flop is basically a gated SR flip-flop with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level "1". Due to this additional clocked input, a JK flip-flop has four possible input combinations, "logic 1", "logic 0", "no change" and "toggle". The symbol for a JK flip flop is similar to that of an *SR* Bistable Latch



Both the S and the R inputs of the previous SR bistable have now been replaced by two inputs called the J and K inputs, respectively after its inventor Jack Kilby. Then this equates to: J = S and K = R.

The two 2-input AND gates of the gated SR bistable have now been replaced by two 3-input NAND gates with the third input of each gate connected to the outputs at Q and Q. This cross coupling of the SR flip-flop allows the previously invalid condition of S = "1" and R = "1" state to be used to produce a "toggle action" as the two inputs are now interlocked.

If the circuit is now "SET" the J input is inhibited by the "0" status of Q through the lower NAND gate. If the circuit is "RESET" the K input is inhibited by the "0" status of Q through the upper NAND gate. As Q and Q are always different we can use them to control the input. When both inputs J and K are equal to logic "1", the JK flip flop toggles as shown in the following truth table.

TRUTH TABLE:-

DATA INPUTS		PRESENT STATE		INPUT TO S-R FLIP FLOP		NEXT OUTPUT
J	**K**	**Qn**	**Q'n**	**S**	**R**	**Qn+1**
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	1	0
1	0	0	1	1	0	1
1	0	1	0	0	0	1
1	1	0	1	1	0	1
1	1	1	0	0	1	0

WHEN (J=K=0)

When both are low and the present output state is reset , there is no effect of clk pulse.so the next outputs remains the last state.

Again When both are low and the present output state is set .so the next outputs remains the last state.

When(J=0,K=1)

When J=0 and K=1 and the preset out put is in reset codition,then the input to the SR flip flop becomes 0'0'.so the next output remains unchanged.but when the preset output is in set codition the input to SR flip flop becomes 0 & 1.so the next output goes reset.

When(J=1,K=0)

When J=1and K=0 and the preset output is in reset condition ,then the input to the SR flip flop becomes 1&0.so the next output goes set.but when the preset output is in set codition the input to SR flip flop becomes 0 & 0.so the next output remains unchange.

When(J=K=1)

When J=1and K=1 and the preset output is in reset condition ,then the input to the SR flip flop becomes 1&0.so the next output goes set.but when the preset output is in set codition the input to SR flip flop becomes 0 & 1.so the next output goes reset.

3.Explain the working of 4 bit ripple counter with truth table and timing diagram.

The counters in which the output of one flipflop drives the another are called ripple counters or asynchronous counters .

In this case flipflop A has to change state before it can trigger the B flipflop and B has to change for triggering C and so on .

Now we discuss how the counting is carried out by the circuit .

Let us consider that all the flipflop are initially reset to output 0 .Thus the output condition to start with will be DCBA=0000

Now on the arrival of the 1^{st} clock pulse the first flipflop A changes state on occurance of –ve edge of the clock pulse .Hence after the end of first pulse output condition will be DCBA =0001

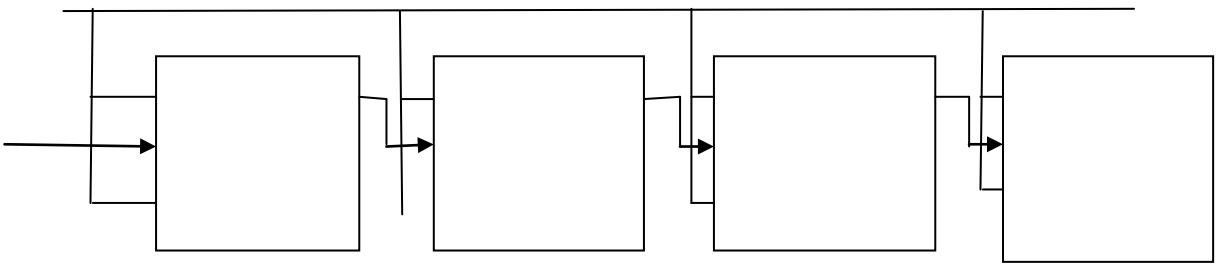
The change of A output from 0 to 1 i.e it is a +ve change and when this +ve change is fed to the clock input of B flipflop ,it cause no change because the change will takeplace only when a –ve edge trigger is applied .

When a 2^{nd} clock pulse arrives at the clock pulse of A flipflop it again change its state i.e it goes from 1 to 0 ,i.e this time at –ve edge so the state of B flipflop is now changes 0 to 1 .This change og 0 to 1 is +ve .So ,after that there is no change now output condition is 0010

Again when we apply clock pulse to A flip flop changes its state 0 to 1 and this is +ve so ,after that there is no change in flip flop output DCBA=0011

In next step DCBA will 0100 ,then 0101 in this way it continue upto 0000.

Because here we take 4 flipflop and we require 2^n input pulse i.e 2^4=16 step .



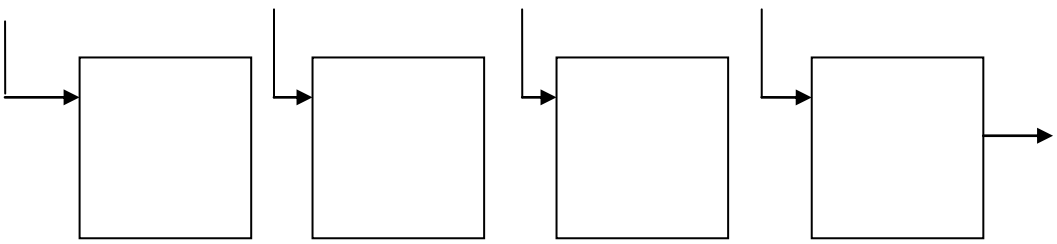
Truth table :-

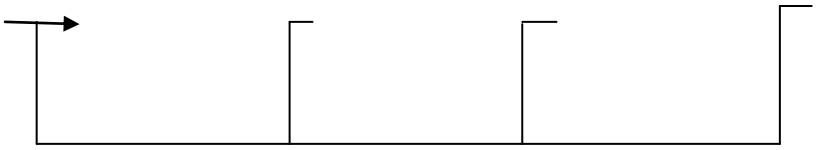
Clock pulse	Output conditions of flip flop			
	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

Q.With a neat diagram explain the operation of PISO register.

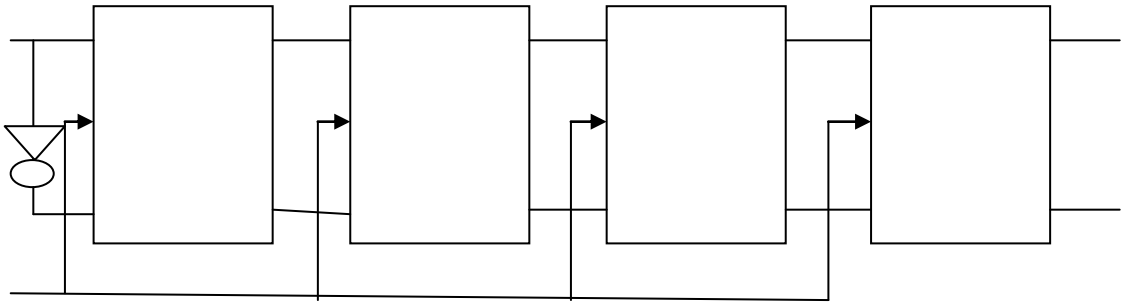
PISO (parallel in serial out):

In such resisters where we have parallel input ,the data bits are entered simultaneously into their respective stages on parallel lines & not on a bit by bit basic on one line as well the case with serial in registers .Serial outputs of these resisters are taken out the same way as we explained for (SISO).After the data bits are properly stored once in the resister the data bits are taken out one by one as was the case with SISO resister .





Q.With neat sketch explain the working of 4 bit SISO register with a neat diagram and timing diagram.



In serial in serial out shift register the binary data is accepted serially i.e one bit at a time is accepted on a line .The stored information is also produced at the output in serial form

As this is a 4 bit register its storage capacity will be 4 bit & will require 4 flipflops to construct it .

Let us consider that the number to be entered be 1010 .Intially counter is in reset on all 0 state .The data line feeds the data word one by one bit starting from the rightmost bit .

First 0 is put on the data input line .This makes S_1 =0 & R_1=1 as the complement of S_1 is fed as R_1 on the occurrence of first pulse under (S_1 =0 & R_1 =1)the 1st flipflop resets & 0 is stored now next bit of data word which is a 1 input on input dataline which makes S_1 =1 & R_1=0 and forces first flipflop to change its state from 0 to 1 on the occurrence of 2nd pulse .Flipflop doesnot change its state .

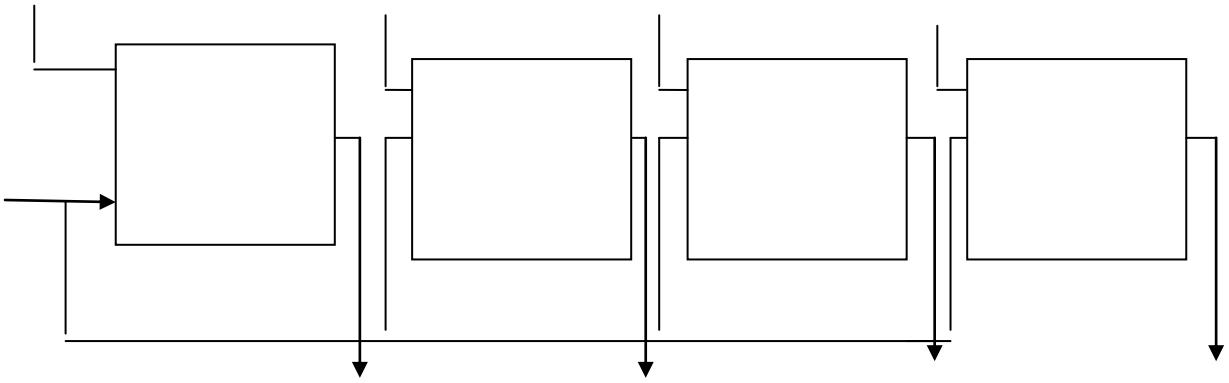
Now the 3 rd bit of data word 0 is put on data input line .This makes S_1 =0 & R_1=1 for first flipflops and on the occurance of 3rd clock pulse the output state of 1st flipflop is changed to 0 from 1 while 2nd flipflop changes from 0 to 1 .The thired flipflop does not change its state .

Now we put 4th bit of data 1 on the data input line which makes S_1 =1 & R_1=0 .Hence when 4th clock pulse occurs flipflop 1 ,2 ,3 change their state while 4th flipflop doesnot change its output state forcing Q_1 =1 ,Q_2 =0 ,Q_3=1 ,Q_4=0 ,So the data word 1010 is shifted into the resister .

Q.With neat sketch explain the working of PIPO and SIPO registor.

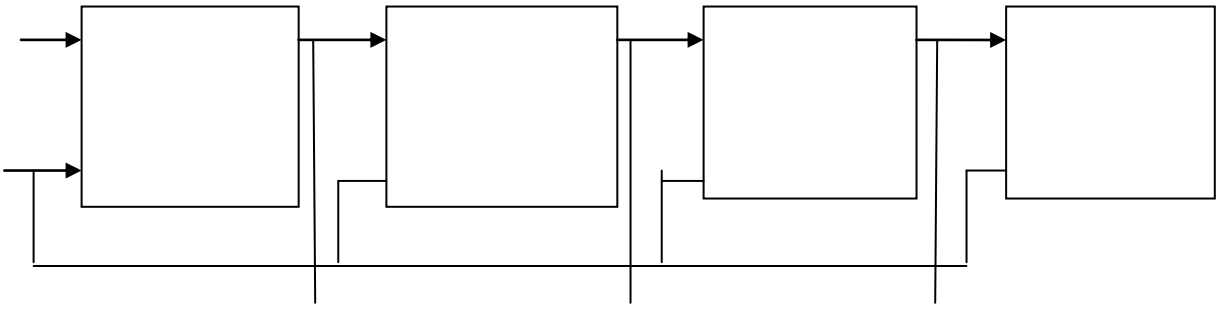
PIPO :(parallel in parallel out)

Here data is given simultaneously to all flipflops and also data is collected from all the flipflops simultaneously .



SIPO(Serial in parallel out)

It is the shift register that accepts data seriously and gives data parallel .Here output is taken from all flipflop simultaneously.



CHAPTER-04

8085 MICROPROCESSOR

SHORT QUESTION

1. **What is microprocessor?**

- Microprocessor is a semiconductor device which takes digital data and processes according to the instruction and the gives the result in digital form within small interval of time.
- It is called brain of the cpu of a computer.

2. **Define microcomputer?**

- Microcomputer is a computer which has only one microprocessor in its CPU.
- Ex- Laptop, PC etc….

3. **What is meant by 8 bit microprocessor?**

- The microprocessor which has word length 8 is known as 8 bit microprocessor. It means that the microprocessor can phase 8 bit of data parallely from the memory.
- It is able to handle$2^8 = 256$ bit.

4. What are the application of microprocessor?

- It is used as CPU of the computer.
- It is used in traffic light computer.
- It also used in industry for automatic voltage controller.
- It is used in vehicle & military system for automatic control.

5. What is an interrupt, write the name of two interrupt of 8085 microprocessor?

- Interrupts are request signals when it goes high , the microprocessor suspends its normal operation and attends the interrupts signal after completing the instruction in hand.
- There are five interrupts in 8085 microprocessor i.e. TRAP, RTS- 7.5, RST 6.5, RST 5.5 & INTR.

6. Write down the flag registers of 8085 microprocessor.

- There are five flag registers in 8085 microprocessor
  a) Carry flag(CS)
  b) Parity flag(P)
  c) Auxilary Flag(AC)
  d) Zero flag(Z)
  e) Sign flag(S)

7. Give one example of 1 byte 2 byte 3 byte instruction.

- 1 byte instruction – MOV A,B
- 2 byte instruction - MVI r,data
- 3 byte instruction - LDA addr

8. Define execution cycle.

- The necessary steps which are carried out to get data, if any from the memory and to perform the specific operation specified in the instruction is known as execution cycle.

9. What is an instruction cycle.

- The necessary steps which are carried out to fetch an instruction from a memory location and to execute is known as instruction cycle.
- It is the summation of phase cycle & execution cycle. That is IC = FC + EC

10. What is machine cycle?.

- The necessary steps which are carried out to perform any operation like Fetch, Read,& write is known as machine cycle.
- The number of T state required may be 3 or 4.

11. Define opcode and operand ?

Ans:- <u>Opcode</u>

Opcode means operational code

When a mnemonics is converted into code then that code is called opcode.

It is necessary to convert mnemonics into code because machine does not understand assembly language

Example:- opcode of ‘LXI H’ is ‘21’

Operand

Operand is a part of an instruction on which operation is carried out.

Example:-MOV A,B

Here A,B is the operand.

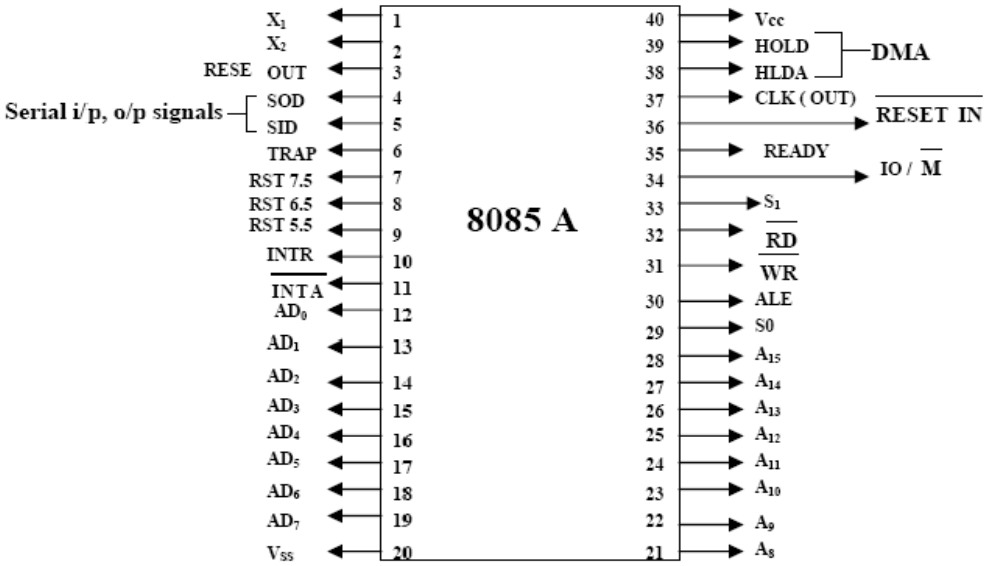
12. What is a counter?

Counter is a sequential circuit. A digital circuit which is used for counting clock pulses is known counter. Counter is the widest application of flip-flops. It is a group of flip-flops with a clock signal applied. Counters are of two types.i.e. synchronous and asynchronous counter

LONG TYPE

1. Draw the pin diagram of 8085 microprocessor and explain the function of each pin.

Pin diagram of 8085 microprocessor is given below:



Description of each pin :

Address and Data Buses:

- A8 – A15 (output, 3-state): Most significant eight bits of memory addresses and the eight bits of the I/O addresses. These lines enter into tri-state high impedance state during HOLD and HALT modes.
- AD0 – AD7 (input/output, 3-state): Lower significant bits of memory addresses and the eight bits of the I/O addresses during first clock cycle. Behaves as data bus during third and fourth clock cycle. These lines enter into tri-state high impedance state during HOLD and HALT modes.

Control & Status Signals:

- ALE: Address latch enable
- $\overline{RD}$: Read control signal.
- $\overline{WR}$: Write control signal.
- IO/$\overline{M}$, S1 and S0 : Status signals.

Power Supply & Clock Frequency:

- Vcc: +5 V power supply
- Vss: Ground reference
- X1, X2: A crystal having frequency of 6 MHz is connected at these two pins
- CLK: Clock output

Externally Initiated and Interrupt Signals:

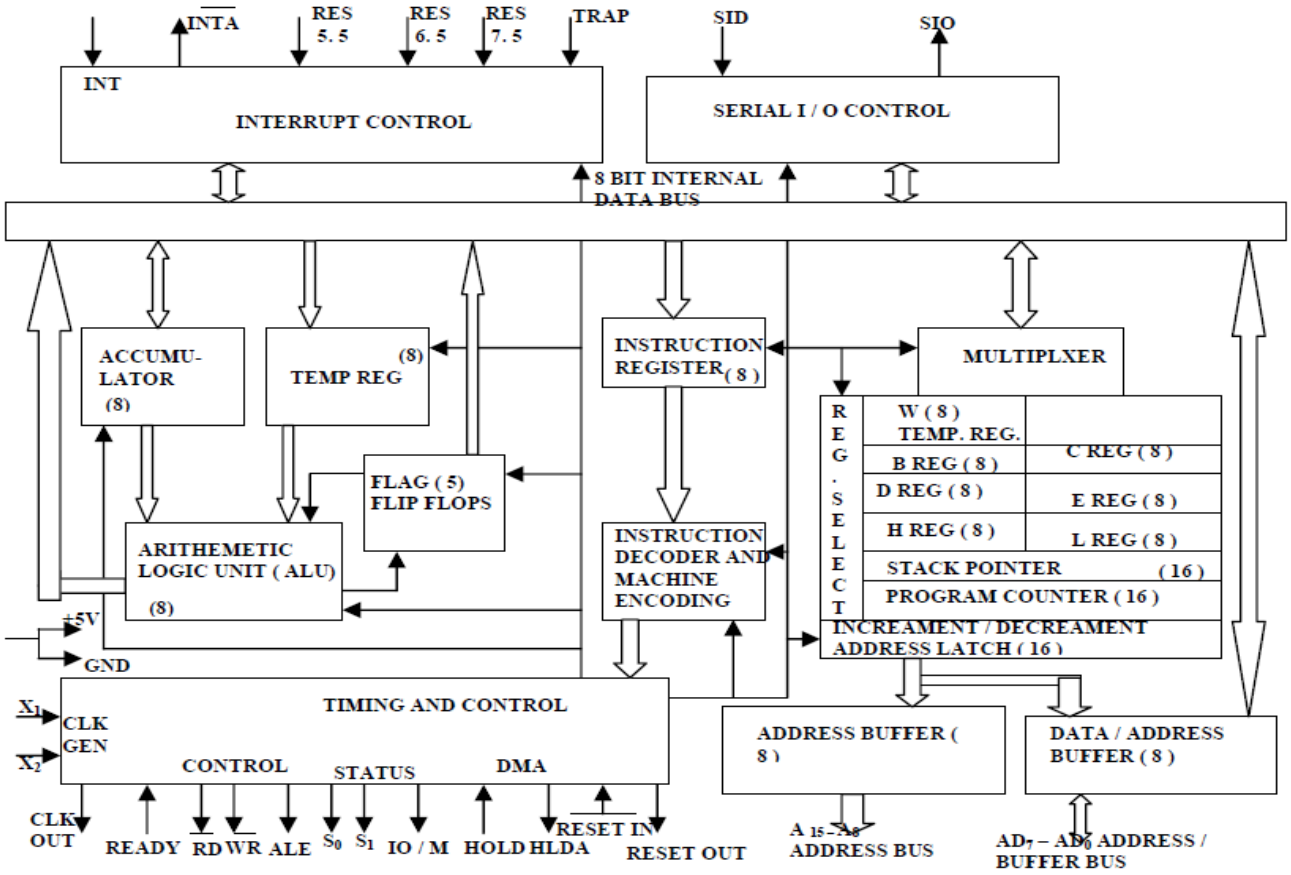
- $\overline{RESET\ IN}$: it reset the microprocessor, that is it reset program counter & temp. register
- RESET OUT: This signal indicates that the processor is being reset. The signal can be used to reset other devices
- READY: When this signal is low, the processor waits for an integral number of clock cycles until it goes high.
- HOLD: This signal indicates that a peripheral like DMA (direct memory access) Controller is requesting the use of address and data bus.
- HLDA: This signal acknowledges the HOLD request.
- INTR: Interrupt request is a general-purpose interrupt.
- $\overline{INTA}$: This is used to acknowledge an interrupt.
- RST 7.5, RST 6.5, RST 5.5 – restart interrupt: These are vectored interrupts and have highest priority than INTR interrupt.
- TRAP: This is a non-mask able interrupt and has the highest priority.

Serial I/O Signals:

- SID: Serial input signal. Bit on this line is loaded to D7 bit of register A using RIM Instruction.
- SOD: Serial output signal. Output SOD is set or reset by using SIM instruction

2. Describe the internal architecture of intel 8085 microprocessor with neat diagram.

The 8085 microprocessor is an 8-bit processor available as a 40-pin IC package and uses +5 V for power. It can run at a maximum frequency of 3 MHz. Its data bus width is 8-bit and address bus width is 16-bit.



Arithmetic and Logic Unit

The ALU performs the actual numerical and logical operations such as Addition (ADD), Subtraction (SUB), AND, OR etc. It uses data from memory and from Accumulator to perform operations. The results of the arithmetic and logical operations are stored in the accumulator.

Registers

The 8085 includes six registers, one accumulator and one flag register, as shown in Fig. In addition, it has two 16-bit registers: stack pointer and program counter. The 8085 has six general-purpose registers to store 8-bit data; these are identified as B, C, D, E, H and L. they can be combined as register pairs - BC, DE and HL to perform some 16-bit operations.

Accumulator (A)

The accumulator is an 8-bit register that is a part of ALU. This register is used to store 8-bit data and to perform arithmetic and logical operations. The result of an operation is stored in the accumulator.

Flag register

The ALU includes five flip-flops, which are set or reset after an operation according to data condition of the result in the accumulator and other registers. They are called Zero (Z), Carry (CY), Sign (S), Parity (P) and Auxiliary Carry (AC) flags. Their bit positions in the flag register are shown in Fig. 4. The microprocessor uses these flags to test data conditions.

Program Counter (PC)

It is a 16 bit special purpose register. It is used to store the address of the next instruction to be executed.

Stack Pointer (SP)

It is also 16 bit register. It is used to store the address of the top element of the stack.

Instruction Register/Decoder

It is an 8-bit register that temporarily stores the opcode of the current instruction of a program. Decoder then takes instruction and decodes or interprets the instruction.

Timing & Control unit

It generates clock pulse & controls the data flow between microprocessor & peripherals though the following buses.

- *Data Bus*: Data bus carries data in binary form between microprocessor and other external units such as memory. It is a 8 bit bidirectional bus.
- *Address Bus*: The address bus carries addresses and is one way bus from microprocessor to the memory or other devices. It is a 16 bit unidirectional bus.
- *Control Bus*: Control bus are various lines which have specific functions for coordinating and controlling microprocessor operations.

3. Explain different addressing modes of a 8085 microprocessor with example.

Addressing Modes

The process of specifying the data to be operated on by the instruction is called addressing. The various formats for specifying operands are called addressing modes. The 8085 has the following five types of addressing:

I. Immediate addressing
II. Direct addressing
III. Register addressing
IV. Register indirect addressing
V. Implicit addressing

- Immediate Addressing:

In this mode, the operand is specified in the instruction itself.

Ex: MVI A, 9AH
Here the data is moved to the accumulator.

- Direct Addressing:

In this mode of addressing the address of the operand is given in the instruction.

Ex: LDA 2500 H

This instruction is used to load the content of memory address 2500 H in the accumulator

- Register Addressing:

Register addressing mode transfers a copy of data from source register to destination register.

Ex: MOV B, C

It copies the content of register C to register B.

- Register Indirect Addressing:

Indirect addressing mode transfers a data between a register and a memory location.

Ex: MOV A, M

Here the data is in the memory location pointed by the content of H-L pair. The data is moved to the accumulator.

- Implicit Addressing

In this addressing mode the data itself specifies the data to be operated upon.

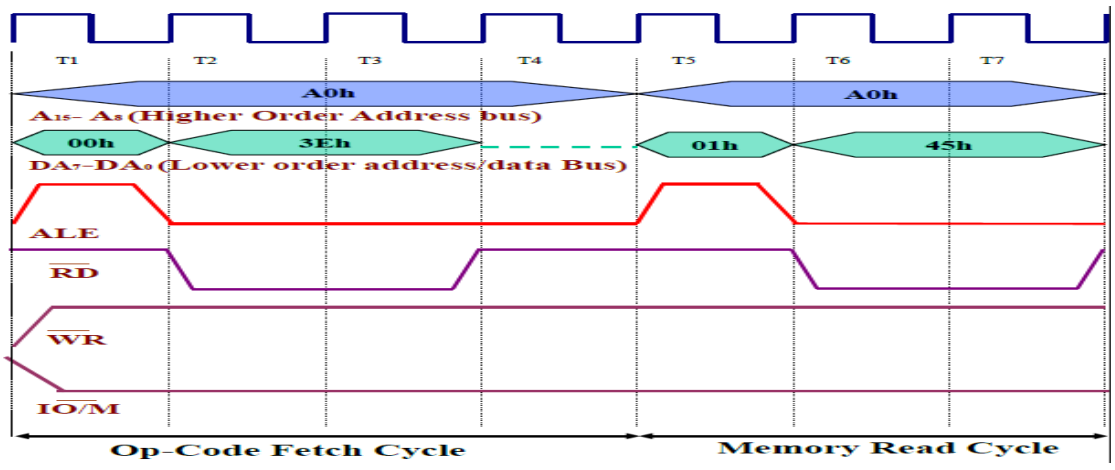
Ex: CMA

The instruction complements the content of the accumulator. No specific data or operand is mentioned in the instruction.

4. What is machine cycle and T state? Draw the timing diagram of MVI B,05 instruction.

Machine cycle – The time required to complete one operation like fetch, Read & write is called machine cycle.

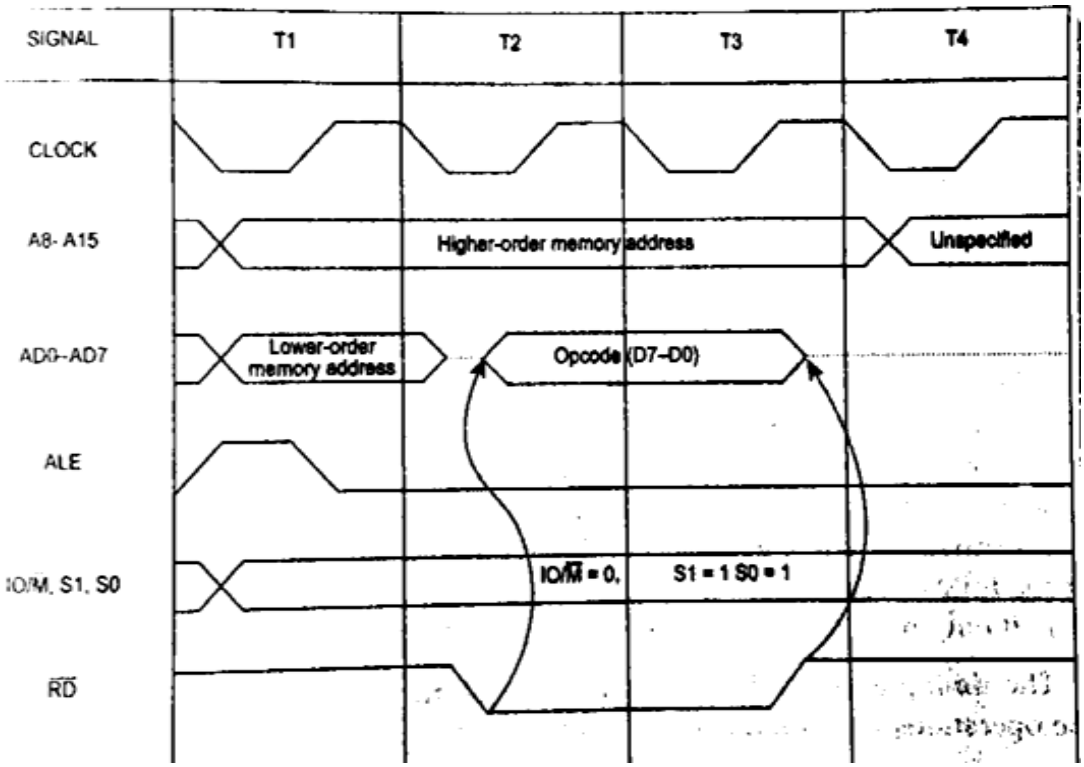
T-State: Time corresponding to one clock period. It is the basic unit to calculate execution of instructions or programs in a processor.



1. Draw the timing diagram of MOV B, A (opcode 47 H). Let the program memory location 4080H.

Or

Draw the timing diagram of opcode fetch?



5. Explain the different group of instruction of 8085 microprocessor with example.

The instructions used in 8085 microprocessor are grouped according to their working principle. These are classified into 5 groups, which are given below

- Data transfer group: Instruction that move data between registers, between register and memory location and I/O transfer.
  - Ex- MOV , MVI, LDA , LXI ,STA etc…
- Arithmetic group: Instructions that add, subtract, increment or decrement data in the register.
  - Ex – ADD, SUB, INR, DCR, INX, DCX etc…
- Logic group: Instruction that carryout logic operation, such an AND, OR, EX-OR compare between and data in the accumulator & a register compliment and rotate data in the accumulator.
  - Ex- ANA, XRA, ORA, CMP, RAL, RAR etc….

- Branch group: Instruction that changes the execution sequence of a program, such as conditional and unconditional jump instruction and subroutine call and return instruction.
  - Ex. – JMP, JC , JZ , CALL , RST etc..
- Stack machine control group: Instruction for maintaining the stack and internal control flags.
  - Ex- IN , OUT , PUSH , POP , HLT etc…

6. Define stack, stack top and stack pointer, why it is essential.

Stack: - Stack is a sequence of memory location set aside by the programmer to store the content of some register or memory location or accumulator so that that registers can be used in the program again. If without storing the content of these registers, it is used again then its content will get deleted. It is about 10% to 20% of the memory. It works in LIFO principle.

Stack top: - The last memory location of the occupied person of the stack is called stack top. The stack pointer holds the address of the stack top.

Stack pointer: - Stack pointer is a 16 bit register and is used to store the address of the stack top. The stack pointer indicates how much data is there in their stack. By the help of stack pointer we can push a data into the stack and POP a data from the stack.

These are essential while a program is shifts from the main program to the sub-outline. Because before shifting the program counter address is stored in the stack and when it is written from the sub-routine to the main program.

CHAPTER-05

INTERFACING AND SUPPORTING CHIPS

1. Why interfacing is needed for I/O device.

- Interfacing is required to connect external peripherals devices with microprocessor. Without interfacing microprocessor will not able to communicate with the peripheral device.

2. Explain what is memory mapped I/O scheme and I/O mapped I/O scheme.

Memory Mapped I/O

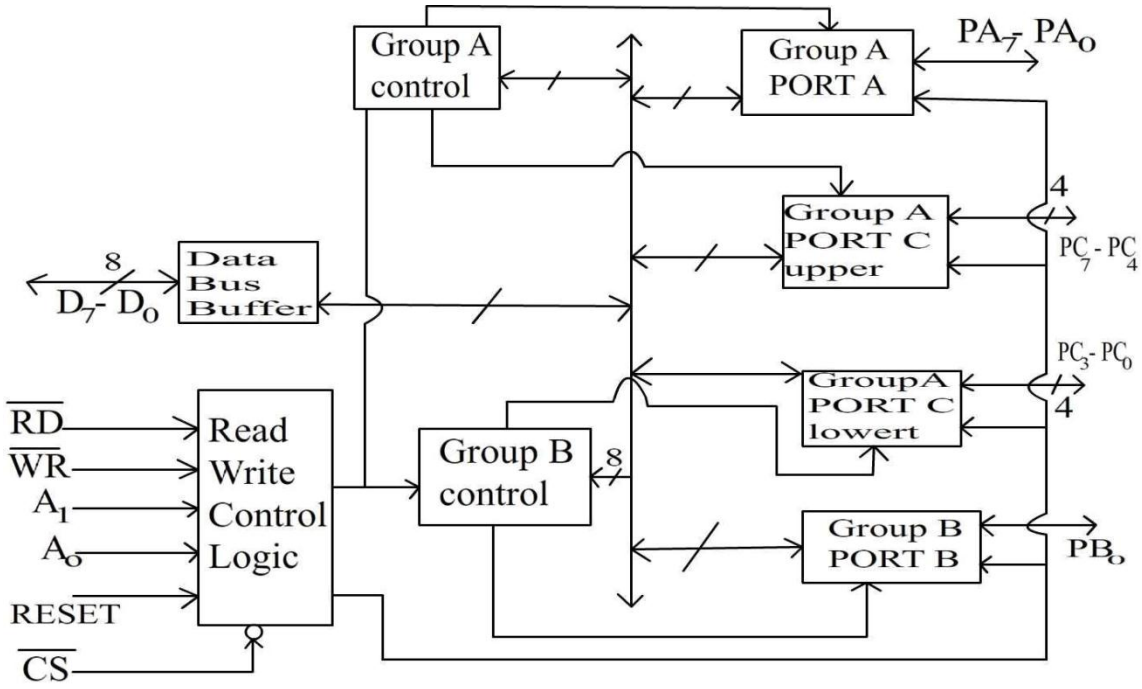
- 16-bit device address
- Data transfer between any general-purpose register and I/O port
- The memory map (64K) is shared between I/O device and system memory.
- More hardware is required to decode 16-bit address
- Arithmetic or logic operation can be directly performed with I/O data

Peripheral Mapped I/O

- 8-bit device address
- Data is transfer only between accumulator and I.O port
- The I/O map is independent of the memory map; 256 input device and 256. output device can be connected
- Less hardware is required to decode 8-bit address

Arithmetic or logical operation cannot be directly performed with I/O data

3. Draw the functional block diagram of 8255 and explain each blaok .



Functional Description:

This support chip is a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. It is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer:

It is a tri-state 8-bit buffer used to interface the chip to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU.

Read/Write and logic control:

The function of this block is to control the internal operation of the device and to control the transfer of data and control or status words through 5 pins like $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, A0, A1 & RESET

PORTs A, B and C:

The 8255A contains three 8-bit ports (A, B and C). All can be configured in a variety of functional characteristic by the system software.

PORTA:

One 8-bit data output latch/buffer and one 8-bit data input latch.

PORT B:

One 8-bit data output latch/buffer and one 8-bit data input buffer.

PORT C:

One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. That is

Port C upper & Port C lower.

Group A & Group B control:

The functional configuration of each port is programmed by the system software. The control words outputted by the CPU configure the associated ports of the each of the two groups. Each control block accepts command from Read/Write content logic receives control words from the internal data bus and issues proper commands to its associated ports.

Control Group A – Port A & Port C upper

Control Group B – Port B & Port C lower