**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**MINIMUM STUDYMATERIAL**

Subject Code: **EC8392**

Subject Name: **Digital Electronics**

SEM / Year: **III/II**

**Regulation: 2017**

**EC8392** **DIGITAL ELECTRONICS** **L T C 3 0 0 3**

## UNIT I -MINIMIZATION TECHNIQUES AND LOGIC GATES 9

**Minimization Techniques**: Boolean postulates and laws – De-Morgan‟s Theorem - Principle of Duality - Boolean expression - Minimization of Boolean expressions — Minterm – Maxterm - Sum of Products (SOP) – Product of Sums (POS) – Karnaugh map Minimization – Don‟t care conditions – Quine - Mc Cluskey method of minimization.

**Logic Gates:** AND, OR, NOT, NAND, NOR, Exclusive–OR and Exclusive–NOR Implementations of Logic Functions using gates, NAND–NOR implementations – Multi level gate implementations- Multi output gate implementations. TTL and CMOS Logic and their characteristics – Tristate gates

## UNIT II- COMBINATIONAL CIRCUITS 9

Design procedure – Half adder – Full Adder – Half subtractor – Full subtractor – Parallel binary adder, parallel binary Subtractor – Fast Adder - Carry Look Ahead adder – Serial Adder/Subtractor - BCD adder – Binary Multiplier – Binary Divider - Multiplexer/ Demultiplexer – decoder - encoder – parity checker – parity generators – code converters - Magnitude Comparator.

## UNIT III- SEQUENTIAL CIRCUITS 9

Latches, Flip-flops - SR, JK, D, T, and Master-Slave – Characteristic table and equation – Application table – Edge triggering – Level Triggering – Realization of one flip flop using other flip flops – serial adder/subtractor- Asynchronous Ripple or serial counter – Asynchronous Up/Down counter - Synchronous counters – Synchronous Up/Down counters – Programmable counters – Design of Synchronous counters: state diagram- State table –State minimization –State assignment - Excitation table and maps-Circuit implementation - Modulo–n counter, Registers – shift registers - Universal shift registers – Shift register counters – Ring counter – Shift counters - Sequence generators.

## UNIT IV- MEMORY DEVICES 9

Classification of memories – ROM - ROM organization - PROM – EPROM – EEPROM –EAPROM, RAM – RAM organization – Write operation – Read operation – Memory cycle - Timing wave forms – Memory decoding – memory expansion – Static RAM Cell-

Bipolar RAM cell – MOSFET RAM cell – Dynamic RAM cell –Programmable Logic Devices – Programmable Logic Array (PLA) - Programmable Array Logic (PAL) – Field Programmable Gate Arrays (FPGA) - Implementation of combinational logic circuits using ROM, PLA, PAL

## UNIT V- SYNCHRONOUS AND ASYNCHRONOUS SEQUENTIAL CIRCUITS

**9**

**Synchronous Sequential Circuits:** General Model – Classification – Design – Use of Algorithmic State Machine – Analysis of Synchronous Sequential Circuits **Asynchronous Sequential Circuits:** Design of fundamental mode and pulse mode circuits – Incompletely specified State Machines – Problems in Asynchronous Circuits – Design of Hazard Free Switching circuits. Design of Combinational and Sequential circuits using VERILOG.

**TOTAL: 45 PERIODS**

## TEXT BOOK:

1. M. Morris Mano, “Digital Design”, 4th Edition, Prentice Hall of India Pvt. Ltd., 2008 / Pearson Education (Singapore) Pvt. Ltd., New Delhi, 2003.

## REFERENCES:

1. John F.Wakerly, “Digital Design”, Fourth Edition, Pearson/PHI, 2008

2. John.M Yarbrough, “Digital Logic Applications and Design”, Thomson Learning, 2006.

3. Charles H.Roth. “Fundamentals of Logic Design”, 6th Edition, Thomson Learning, 2013.

4. Donald P.Leach and Albert Paul Malvino, “Digital Principles and Applications”, 6th Edition, TMH, 2006.

5. Thomas L. Floyd, “Digital Fundamentals”, 10th Edition, Pearson Education Inc, 2011

6. Donald D.Givone, “Digital Principles and Design”, TMH, 2003.

# EC8392- DIGITAL ELECTRONICS

## 1. Aim and Objective of the Subject

- To introduce basic postulates of Boolean algebra and shows the correlation between Boolean expressions
- To introduce the methods for simplifying Boolean expressions
- To outline the formal procedures for the analysis and design of combinational circuits and sequential circuits
- To introduce the concept of memories and programmable logic devices.
- To illustrate the concept of synchronous and asynchronous sequential circuits

## 2. Need and Importance for Study of the Subject

- It helps the students how to design and implement combinational and sequential circuits
- It helps the students how the design procedure produce highly complex digital circuits at low costs.
- It helps the students how to write the Verilog program code to execute the complex digital logic circuits

## 3. Industry Connectivity and Latest Developments

- There are a variety of career opportunities in product companies, design services companies and electronic design automation (EDA) companies like cadence, synopsis, etc.,
- Digital electronics circuits are used in daily life. Digital camera, music system, laptop, etc., and also in medical field digital devices plays a vital role.

## 4. Industrial Visit (Planned if any):

One day visit to **ARASAN CHIP SYSTEMS INC** planned in the third week of September.

## DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

***Detailed Lesson Plan***

Name of the Subject& Code: **EC 8392- DIGITAL ELECTRONICS**



| S.No | Unit | Topics to be Covered | No. of Hours | Cummulative Hours | Text |
|---|---|---|---|---|---|
| **UNIT I : MINIMISATION TECHNIQUES AND LOGIC GATES** | | | | | |
| 1 | I | Boolean postulates and laws – De-Morgan's Theorem- Principle of Duality | 1 | 1 | T1,T2, R2 |
| 2 | | Boolean expression –Minimization of Boolean expressions | 1 | 2 | T1,T2, R1,R2 |
| 3 | | Minterm – Maxterm - Sum of Products (SOP) Product of Sums (POS) | 1 | 3 | T1,T2, R2 |
| 4 | | Karnaugh map Minimization | 1 | 4 | T1,T2, R1,R2 |
| 5 | | Don't care conditions | 1 | 5 | T1,T2 |
| 6 | | Quine-McCluskey method of minimization | 1 | 6 | T2,R1 |
| 7 | | AND, OR, NOT, NAND, NOR, Exclusive – OR and Exclusive – NOR | 1 | 7 | T1,T2 |
| 8 | | Implementations of Logic Functions | 1 | 8 | T1,T2 |

| | | | | | |
|---|---|---|---|---|---|
| | | using gates, NAND, NOR implementation | | | |
| 9 | | Multi level gate implementations, Multi output gate implementations | 1 | 9 | T2 |
| 10 | | TTL and CMOS Logic and their characteristics –Tristate gates. | 1 | 10 | T1,T2 |
| **UNIT II : COMBINATIONAL CIRCUITS** | | | | | |
| 11 | | Design procedure – Half adder – Full Adder – Half Subtractor – Full Subtractor | 1 | 11 | T1,T2 |
| 12 | | Parallel-binary adder, binary Subtractor – Fast Adder Carry Look Ahead adder | 1 | 12 | T1,T2 |
| 13 | | Serial Adder/ Subtractor | 1 | 13 | T2 |
| 14 | | BCD adder | 1 | 14 | T1 |
| 15 | | Binary Multiplier – Binary Divider | 1 | 15 | T1,T2 |
| 16 | II | Multiplexer/ Demultiplexer | 1 | 16 | T1,T2 |
| 17 | | Encoder / decoder | 1 | 17 | T1,T2 |
| 18 | | Parity checker, Parity generator | 1 | 18 | T2 |
| 19 | | Code converters | 1 | 19 | T2 |
| 20 | | Magnitude Comparator | 1 | 20 | T1 |
| **UNIT III: SEQUENTIAL CIRCUITS** | | | | | |
| 21 | III | Latches, Flip flops SR, JK, T, D– Characteristic table and equation – Application table | 1 | 21 | T1,T2, R1,R2 |
| 22 | | Master slave Flip flop | 1 | 22 | T1,T2 |
| 23 | | Edge triggering –Level Triggering | 1 | 23 | T1,T2 |

| | | | | | |
|---|---|---|---|---|---|
| 24 | | –Realization of one flip flop using other flip flops | 1 | 24 | T2 |
| 25 | | Serial adder/ Subtractor | 1 | 25 | T2 |
| 26 | | Asynchronous Ripple or serial counter | 1 | 26 | T1,T2, R1 |
| 27 | | Asynchronous Up/Down counter | 1 | 27 | T2 |
| 28 | | Synchronous counters | 1 | 28 | T1,T2 |
| 29 | | Synchronous Up/Down counters | 1 | 29 | T2 |
| 30 | | Design of Synchronous counters | 1 | 30 | T1,T2 |
| 31 | | Modulo – n counter | 1 | 31 | T1,R1 |
| **UNIT IV : MEMORY DEVICES** | | | | | |
| 32 | IV | Classification of memories – ROM - ROM organization - PROM – EPROM – EEPROM – EAPROM | 1 | 32 | T1,T2, R1 |
| 33 | | RAM – RAM organization – Write operation – Read operation | 1 | 33 | T1,T2 |
| 34 | | Memory cycle - Timing wave forms | 1 | 34 | T1,T2 |
| 35 | | Memory decoding – memory expansion | 1 | 35 | T1,T2 |
| 36 | | Static RAM Cell-Bipolar RAM cell – MOSFET RAM cell – Dynamic RAM | 1 | 36 | T2 |
| 37 | | Programmable Logic Devices– Programmable Logic Array (PLA) | 1 | 37 | T2,R1 |
| 38 | | Programmable Array Logic (PAL) | 1 | 38 | T1,T2 |
| 39 | | Field Programmable Gate Arrays (FPGA) | 1 | 39 | T1,T2, R1 |
| 40 | | Implementation of combinational logic circuits using ROM, PLA, PAL | 1 | 40 | T1,T2 |
| **UNIT V SYNCHRONOUS & ASYNCHRONOUS SEQUENTIAL CICUITS** | | | | | |
| 41 | | Synchronous Sequential Circuits: | 1 | 41 | T2,R2 |

| | | | | | |
|---|---|---|---|---|---|
| | V | General Model | | | |
| 42 | | Classification – Design – Use of Algorithmic State Machine | 1 | 42 | T1,T2 |
| 43 | | Analysis of Synchronous Sequential Circuits | 1 | 43 | T1,T2 |
| 44 | | Asynchronous Sequential Circuits: Design of fundamental mode circuit | 1 | 44 | T1,T2, R1 |
| 45 | | Incompletely specified State Machines | 1 | 45 | T2 |
| 46 | | Problems in Asynchronous Circuits | 1 | 46 | T1,T2 |
| 47 | | Design of Hazard Free Switching circuits | 1 | 47 | T1,T2 |
| 48 | | Design of Combinational and Sequential circuits using VERILOG | 1 | 48 | T1 |

# TABLE OF CONTENTS

# UNIT-I

## MINIMISATION TECHNIQUES AND LOGIC GATES

## PART A

1. **Prove the Boolean theorems(a) x + x = x ; (b) x + xy = x**

**(a) x + x = x**

x + x = (x + x). 1

= (x + x)(x+x')

= x + xx'

= x + 0

x + x = x

**(b) x + xy = x**

x + xy = x . 1 +xy

= x(1+y)

= x(y+1)

= x.1

x + xy = x

2. **Define Noise Margin.**

It is the maximum noise voltage added to an input signal of a digital circuit that does not cause an undesirable change in the circuit output. It is expressed in volts.

3. **State De Morgan Theorem. (MAY/JUN 2013)**

1. $\overline{AB} = \overline{A} + \overline{B}$

The complement of product is equal to the sum of the complement.

2. $\overline{A+B} = \overline{A}\,\overline{B}$

The complement of sum is equal to the product of the complement.

4. **State Distributive Law. (NOV/DEC 2013)**

A (B+C)=AB+AC

It states that OR in several variable and AND in the result with single variable is equivalent to AND in the result with a single variable with each of the single variables and then OR in the product.

5. **What is Prime Implicant? (NOV/DEC 2013)**

Each group in a K MAP gives us a product term and summation of all product term gives us a boolean expression. Therefore each product term implies the function and hence each product term is an implicant of the function. All the implicant of the function is the termed as prime implicant.

6. **What are Don't Care Terms? (MAY/JUN 2013)**

In some logic circuits certain input condition never occur, therefore the output of such inputs are indicated by"X" (or) DON'T CARE OUTPUTS.

$Z=\Sigma_m(1,3)+d(5,6,7)$

Here (1,3) are minterms ,d (5,6,7)represent DON'T cares.

7. **Apply DE MORGAN Theorem to [(A+B)+C]' (MAY/JUN 2014)**

[(A+B)+C]' = (A + B) ' . C'

= A'.B'.C'

8. **Simplify the following Boolean expression into one literal.**

**W'X(Z' + YZ)+X(W + Y'Z).**

W'X(Z' + YZ)+X(W + Y'Z) = W'X(Z'+YZ) + X(W+Y'Z)

= X (W'(Z'+Y) +W+Y'Z)

= X (W'Z+W+Y+Y'Z)

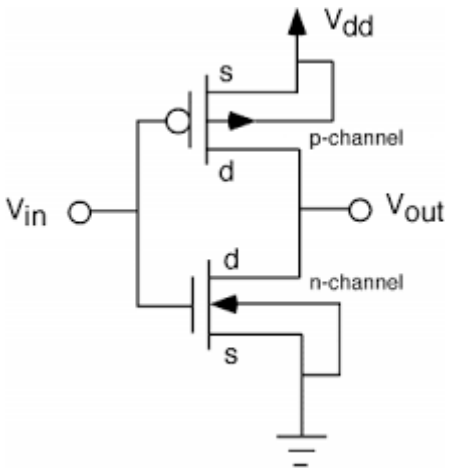
= X(W'Z'+W+Y+Y'Z)

= X(W+Z'+Y+Z)

= X(W +1 +Y)

= X

9. **Draw the CMOS inverter circuit. (NOV/DEC 2014)**



**10. Express the function of Y =A+ B'C**

**Y =A+ B'C**

The function has three variables A,B,C. The first term A is missing two variables. Therefore,

A = A(B+B')

= AB + AB'

= AB(C+C')+AB'(C+C')

= ABC + ABC' +AB'C +AB'C'

The second term B'C is missing one variable,

B'C = B'C(A+A') = AB'C +A'B'C

Combining all terms,

Y = A + B'C

Y = ABC + ABC' +AB'C +AB'C'+ AB'C +A'B'C

**11. Define Minterm & maxterm**

- N variables forming an OR term, with each variable being primed or unprimed, provide $2^n$ possible combinations called minterm.
- N variables forming an AND term, with each variable being primed or unprimed, provide $2^n$ possible combinations called maxterm.

**12. Write a note on Tristate gates.**

A three state gate exhibits three output states:

(1) A low level state

(2) High level state

(3) High impedance state

**13. State the advantage of CMOS Logic. (APR/MAY 2015)**

1. Power dissipation is very small in the range of Nw.

2. It operate in wide range of power supply voltage

3. The fan-out is more than TTL.

**14. . Define the term Fan out.**

It specifies the number of standard loads that the output of typical gate can drive without impairing its normal operation.

A standard load is usually defined as the amount of current needed by an input of another similar gate of same family.

**15.Simplify the following expression X.Y+X(Y+Z)+Y(Y+Z) (NOV/DEC 2016)**

= XY+XY+XZ+YY+YZ

=XY+XZ+Y+YZ [Y.Y=Y]

=XY+XZ+Y(1+Z) [1+Z=1]

=XY+XZ+Y

=Y(X+1)+XZ [X+1=1]

=Y+XZ

**16.Why totem pole outputs cannot be connected together. (NOV/DEC 2016)**

Totem pole outputs cannot be connected together because such a connection might produce excessive current and may result in damage to the device

## PART-B

**1. Simplify the function F using Quine Mccluskey method and verify the result using K-Map $F(A,B,C,D) = \sum(1,2,3,5,7,9,10,11,13,15)$ Model-MAY2016, 2015, 2014**

**Solution:**

**Step 1:** List the minterms in binary form

| Minterm | Binary number |
|---|---|
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 5 | 0101 |
| 7 | 0111 |
| 9 | 1001 |
| 10 | 1010 |
| 11 | 1011 |
| 13 | 1101 |
| 15 | 1111 |

**Step 2:** Group the minterms according to number of 1's

| No of 1's | Minterm | Binary Number |
|---|---|---|
| 1 | 1 | 0001 |
| | 2 | 0010 |
| 2 | 3 | 0011 |
| | 5 | 0101 |
| | 9 | 1001 |
| | 10 | 1010 |
| 3 | 7 | 0111 |
| | 11 | 1011 |
| | 13 | 1101 |
| 4 | 15 | 1111 |

**Step 3:** Compare each binary number in each group with every term in the adjacent higher group for they differ only by one position. Repeat this step for various cell combinations

2-cell combination

| Minterms | Binary number |
|---|---|
| 1,3 | 00_1 ✓ |
| 1,5 | 0_01 ✓ |
| 1,9 | _001 ✓ |
| 2,3 | 001_ ✓ |
| 2,10 | _010 ✓ |
| 3,7 | 0_11 ✓ |
| 3,11 | _011 ✓ |
| 5,7 | 01_1 ✓ |
| 5,13 | _101 ✓ |
| 9,11 | 10_1 ✓ |
| 9,13 | 1_01 ✓ |
| 10,11 | 101_ ✓ |
| 7,15 | _111 ✓ |

| | |
|---|---|
| 11,15 | 1_11 ✓ |
| 13,15 | 11_1 ✓ |

4-cell combination

| Minterms | Binary number |
|---|---|
| 1,3,5,7 | 0_ _1 ✓ |
| 1,3,9,11 | _0_1 ✓ |
| 1,9,5,13 | _ _01 ✓ |
| 2,3,10,11 | _01_ |
| 3,11,7,15 | _ _11 ✓ |
| 5,7,13,15 | _1_1 ✓ |
| 9,11,13,15 | 1_ _1 ✓ |

8-cell combination

| Minterms | Binary number |
|---|---|
| 1,3,5,7,9,11,13,15 | _ _ _1 |

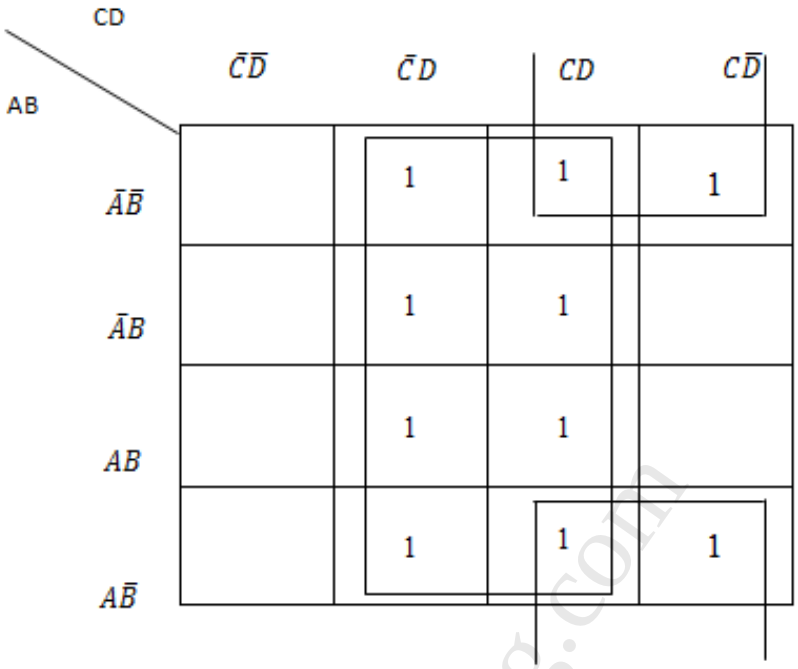
**Step 4:** Form the prime implication table and find the Boolean expression

| | 1 | 2 | 3 | 5 | 7 | 9 | 10 | 11 | 13 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1,3,5,7,9,11,13,15 | * | | * | * | * | * | | * | * | * |
| 2,3,10,11 | | * | * | | | | * | * | | |
| | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ |

$$F = D + \bar{B}$$

**Step 5:** Draw the K-Map and verify the Boolean expression




$$F = D + \bar{B}$$

**2. Simplify the function F using Tabulation method, Model- <u>NOV/DEC 2015</u>**

$$f(A, B, C, D) = \Sigma m\ (1, 3, 5, 8, 9, 11, 15) + d(2, 13)$$

**<u>Solution</u>**

**<u>Step 1</u>**

Grouping of minterms/don't care terms according to number of 1's.

| Group | Minterm/ don't care term | Variables A | B | C | D | Check for inclusion in group of 2 |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 1 | ✓ |
| | 2* | 0 | 0 | 1 | 0 | ✓ |
| | 8 | 1 | 0 | 0 | 0 | ✓ |
| 2 | 3 | 0 | 0 | 1 | 1 | ✓ |
| | 5 | 0 | 1 | 0 | 1 | ✓ |
| | 9 | 1 | 0 | 0 | 1 | ✓ |
| | 11 | 1 | 0 | 1 | 1 | ✓ |
| 3 | 13* | 1 | 1 | 0 | 1 | ✓ |
| 4 | 15 | 1 | 1 | 1 | 1 | ✓ |

**Step 2**

Grouping of 2 minterms/don't care terms

| Group | Minterms/ don't care terms | Variables A | B | C | D | Check for inclusion in group of 4 |
|---|---|---|---|---|---|---|
| 1 | 1, 3 | 0 | 0 | — | 1 | ✓ |
| | 1, 5 | 0 | — | 0 | 1 | ✓ |
| | 1, 9 | — | 0 | 0 | 1 | ✓ |
| | 2*, 3 | 0 | 0 | 1 | — | |
| | 8, 9 | 1 | 0 | 0 | — | |
| | 3, 11 | — | 0 | 1 | 1 | ✓ |
| | 5, 13* | — | 1 | 0 | 1 | ✓ |
| 2 | 9, 11 | 1 | 0 | — | 1 | ✓ |
| | 9, 13* | 1 | — | 0 | 1 | ✓ |
| 3 | 11, 15 | 1 | — | 1 | 1 | ✓ |
| | 13, 15 | 1 | 1 | — | 1 | ✓ |

**Step 3**

Grouping of 4 minterms/don't care terms

| Group | Minterms/ don't care terms | Variables A | B | C | D |
|---|---|---|---|---|---|
| 1 | 1, 3, 9, 11 | — | 0 | — | 1 |
| | 1, 5, 9, 13* | — | — | 0 | 1 |
| | 1, 9, 3, 11 | — | 0 | — | 1 |
| | 1, 9, 5, 13* | — | — | 0 | 1 |
| | 9, 11, 13*, 15 | 1 | — | — | 1 |
| 2 | 9, 13*, 11, 15 | 1 | — | — | 1 |

**Step 4**

Prime Implication Table and Boolean Expression

| PI terms | Decimal numbers | Minterms/don't care terms 1 | 2* | 3 | 5 | 8 | 9 | 11 | 13* | 15 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\overline{B}D$ | 1, 3, 9, 11 | × | | × | | | × | × | | |
| $\overline{C}D$ | 1, 5, 9, 13* ✓ | × | | | ⊗ | | × | | × | |
| $AD$ | 9, 11, 13*, 15 ✓ | | | | | | × | × | × | ⊗ |
| $A\overline{B}C$ | 2*, 3 | | × | × | | | | | | |
| $A\overline{B}C$ | 8, 9 | ✓ | | | | ⊗ | × | | | |
| | | | | | ✓ | ✓ | | | | ✓ |

$$f(A, B, C, D) = \overline{B}D + \overline{C}D + AD + A\overline{B}\overline{C}$$

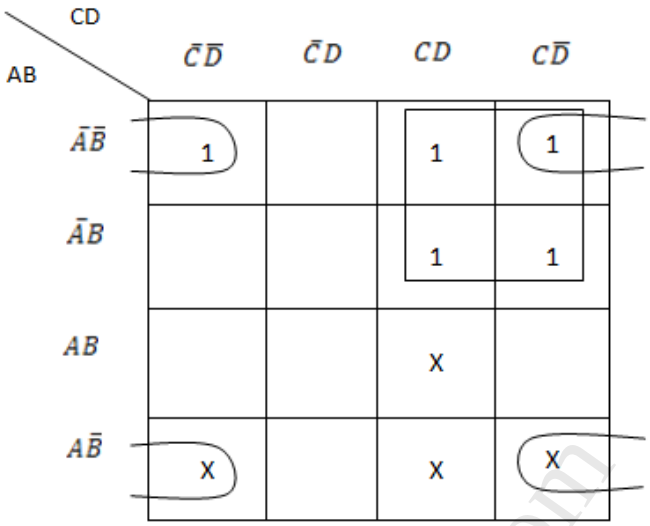
**3.Simplify the function in SOP and POS using K-Map**

$F = \sum(0,2,3,6,7) + d(8,10,11,15)$

**Solution:**

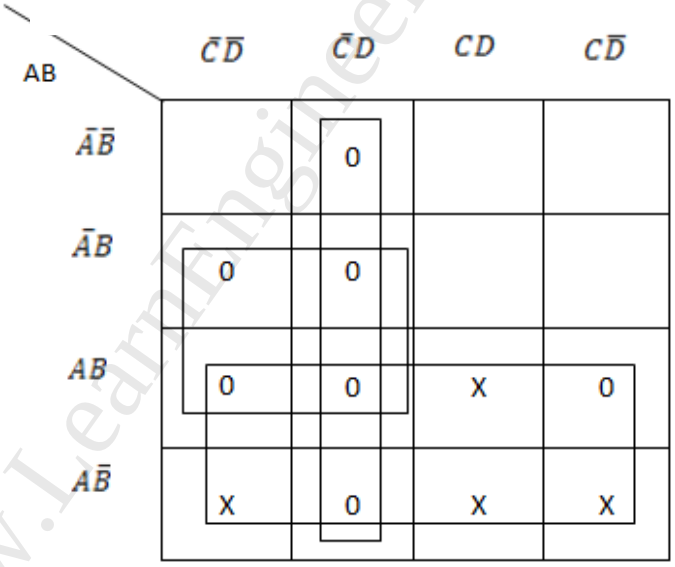
(i) Sum of Products (SOP)




$$F = \bar{A}\bar{C} + \bar{B}\bar{D}$$

(i) Product of Sum (POS)




$$F = A + \bar{B}\bar{C} + \bar{C}\bar{D}$$

**4. Implement basic logic gates using UNIVERSAL gates. NOV 2015, MAY 2015**

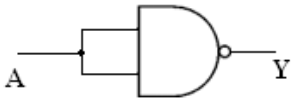
OR, AND and NOT gates are the three basic logic gates as they together can be used to construct the logic circuit for any given Boolean expression. NOR and NAND gates have the property that they individually can be used to hardware-implement a logic circuit corresponding to any given Boolean expression. That is, it is possible to use either

only NAND gates or only NOR gates to implement any Boolean expression. This is so because a combination of NAND gates or a combination of NOR gates can be used to perform functions of any of the basic logic gates. It is for this reason that NAND and NOR gates are universal gates.

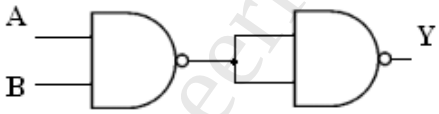
**Implementation using NAND gate:**

**(a) NOT gate**



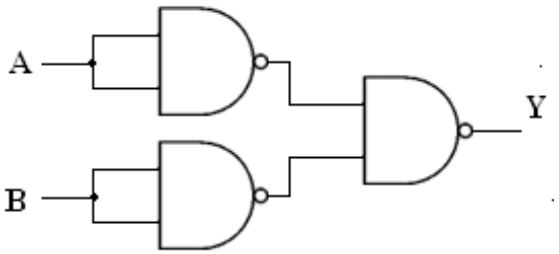

| A | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |

**(b) AND gate**




| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**(c) OR gate**

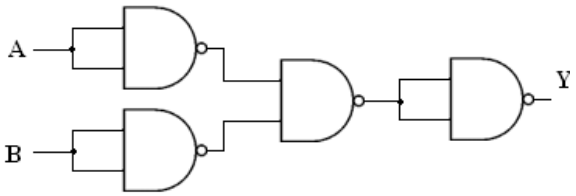



| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |

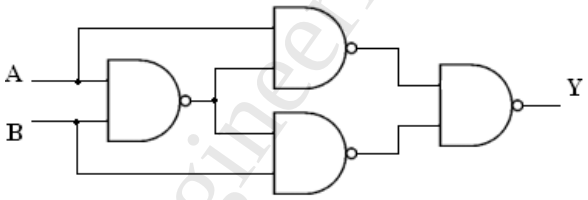
| | | |
|---|---|---|
| 1 | 1 | 1 |

**(d)NOR gate**




| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

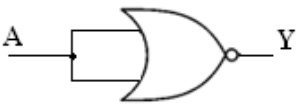
**(e)Ex-OR gate**




| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Implementation using NOR gate:**

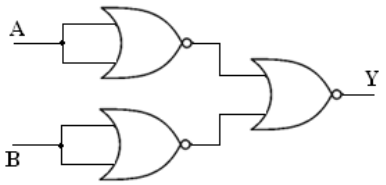
**(a) NOT gate**




| A | Y |
|---|---|
| 0 | 1 |
| 1 | 0 |

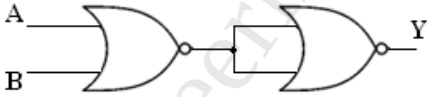
 

**(b) AND gate**



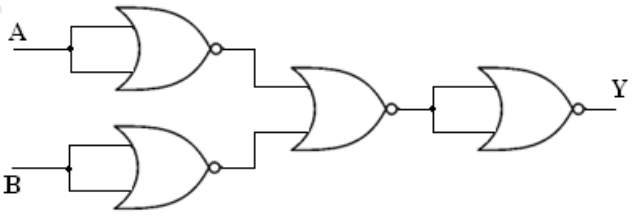

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**(c) OR gate**




| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

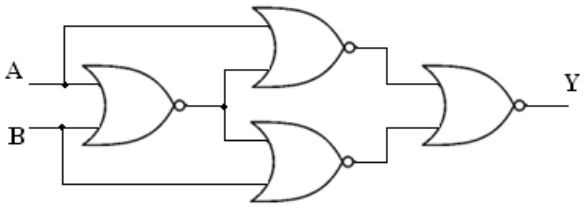
**(d) NAND gate**




| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**(e) Ex-NOR gate**




| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

These gates need to be connected to an external resistor, called the pull-up resistor, between the output and the DC power supply to make the logic gate perform the intended logic function. Depending on the logic family used to construct the logic gate, they are referred to as gates with open collector output (in the case of the TTL logic family) or open drain output (in the case of the MOS logic family).

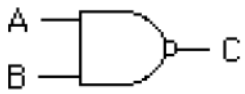
**5. Explain the operation of TTL with neat circuit diagram.**

**NOV2015, 2014 MAY 2016**

The standard TTL gate was the first version in the TTL family. This basic gate was then designed with different resistor values to produce gates with lower power dissipation or with higher speed. In the low-power TIL gate the resistor values are higher than in the standard gate in order to reduce the power dissipation but the propagation delay is increased. In the high-speed TTL gate, resistor values are lowered to reduce the propagation delay, but the power dissipation is increased.

LOW (L) voltages in the range 0 V to 0.8 V are considered to be logic 0, and HIGH (H) voltages in the range 2.0 V to 5.5 V are considered to be logic 1. Fig. [a] illustrates the voltage levels for all possible input combinations to a two-input TTL NAND gate.




| A | B | C |
|---|---|---|
| L | L | H |
| L | H | H |
| H | L | H |
| H | H | L |

**Fig. [a] Voltage Level Table for a Two-input TTL NAND Gate**

A voltage transfer curveis a graph of the input voltage to a gate versus its output voltage. Fig [b] shows the transfer curve for TTL inverter without any fanout. When the input voltage is 0 V, the output is HIGH at 3.3 V. As the input voltage is increased from 0 to 0.7 V, the output remains relatively constant (Region I). Beyond 0.7 V to about 1.2 V, the output decreases more gradually with increasing input voltage (Region II). The *threshold voltage*, the voltage on the transfer curve at which $V_{out} = V_{in}$ and occurs in Region III, is found at the intersection of the transfer curve and the line $V_{out} = V_{in}$. Finally, in Region IV, the output remains constant at 0.2V as the input voltage is increased.

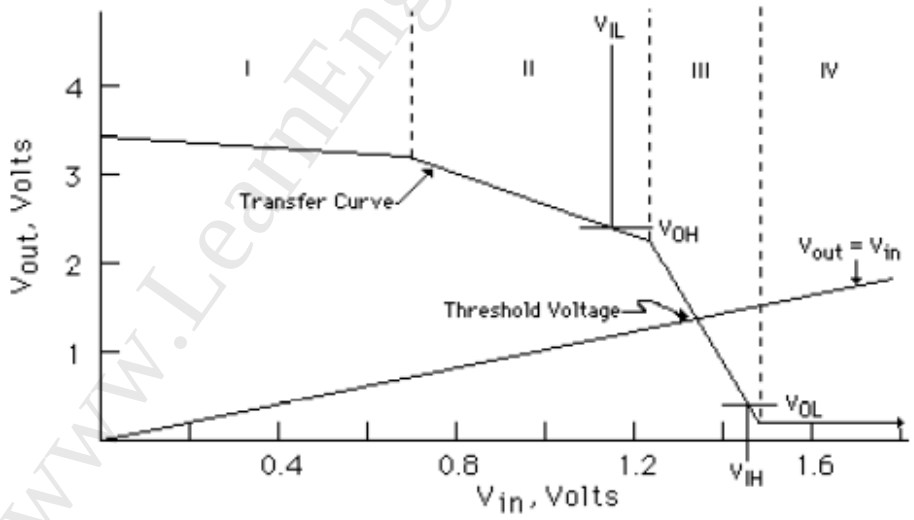



**Fig. [b] Voltage Transfer Curve for a TTL Gate**

TTL gates come in three different types of output configuration:

1. Open-collector output
2. Totem-pole output
3. Three-state output

**Open-Collector Output Gate**

Open collector output gates can only drive their output LOW; for input combinations where the output should be HIGH, an external pull-up resistor connected the supply voltage is needed to produce the HIGH. The outputs of open-collector gates to be wired together; the result is to effectively AND all the output signals together. The basic TIL gate shown in Fig.[c] is a modified circuit of the DTL gate. The multiple emitters in transistor *Q1* are connected to the inputs. The base-collector junction of *Q1* acts as *pn*junction diode. The output of the TTL gate is taken from the open collector of *Q3.* A resistor connected to *Vcc*must be inserted externally for the output to "pull up" to the high voltage level when *Q3* is off; otherwise the Output acts as an open circuit. The two voltage levels of the TTL gate are 0.2 V for the low level and from 2.5to *5* v for the high level. The basic circuit is a NAND gate; if any input is low the corresponding base-emitter junction in *Q1* is forward biased. If all inputs are high both *Q2* and *Q3* conduct and saturate. Since all inputs are high and greater than 2.4 V. the base-emitter junctions of *Q*1 are all reverse biased. When output transistor *QJ* saturates the output voltage goes low to 0.2 V. This confirms the conditions of a NAND operation.

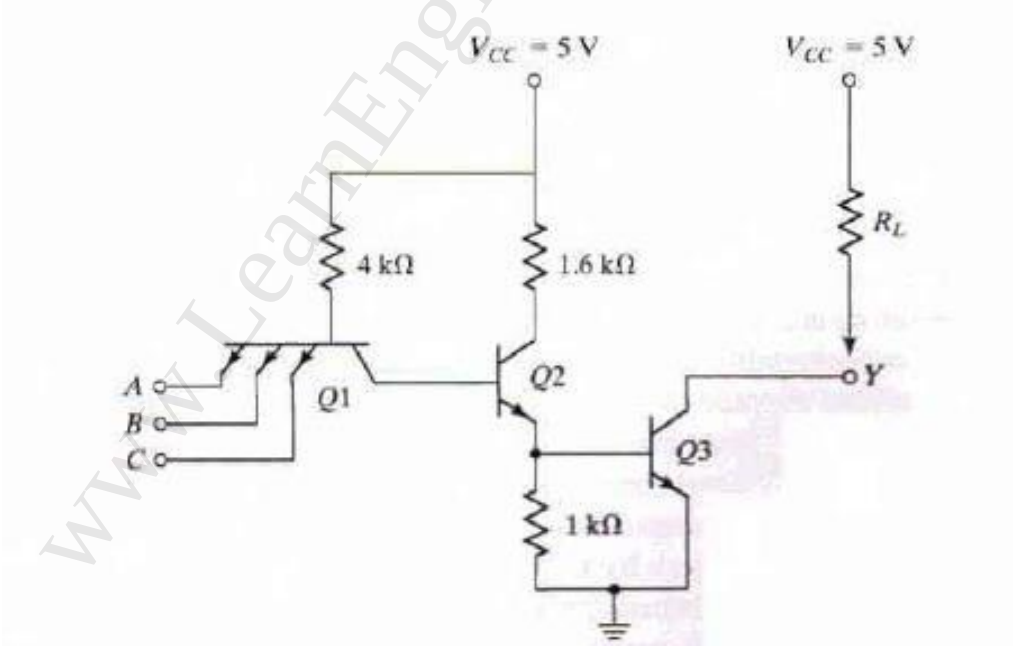



**Fig. [c] Open collector TTL NAND gate**

With the outputs of open-collector gate s connected together the common output is high only when *all* output transistors are off (or high). If an output transistor conducts it forces the output into the low state. The wired logic performed with open-collector TTL gates is depicted in Fig [d].

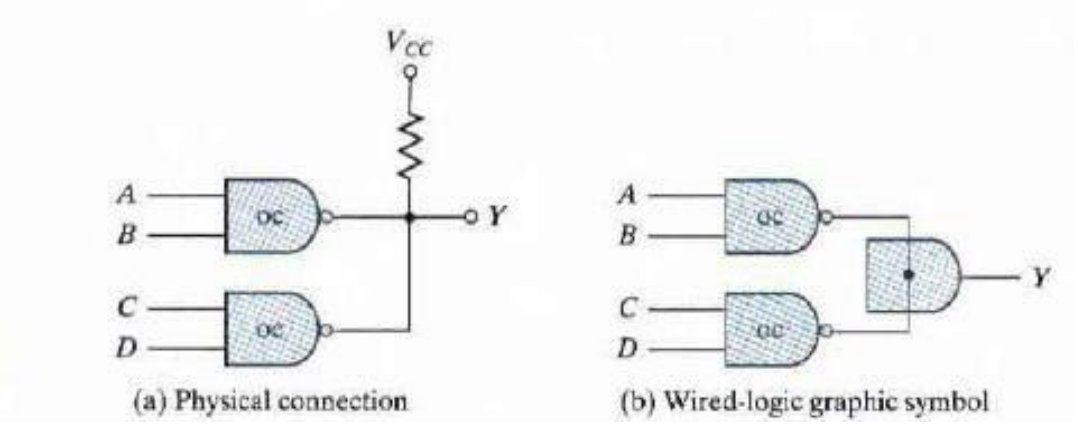



**Fig.[d] Wired-AND of two open collector gates $Y = \overline{(AB + CD)}$**

The AND function formed by connecting the two outputs together is called a wired-AND function. Open-collector gates can be tied together to form a common bus. At any time, all gate outputs tied to the bus except one must be maintained in their high state.

## Totem-Pole Output

The output impedance of a gate is normally a resistive plus a capacitive load. The capacitive load consists of the capacitance of the output transistor the capacitance of the fan-out gates and any stray wiring capacitance. When the output changes from the low to the high state he output transistor of the gate goes from saturation to cutoff and the total load capacitance C charges exponentially from the low to the high voltage level with a time constant equal to RC For the open-collector gate. *R* is the external resistor marked $R_L$. For a typical operating value of C = 15 pF and $R_L$= 4 kʊ the propagation delay of a TTL open-collector gate during the turnoff time is 35 ns. With an active *pull-up* circuit replacing the passive pull-up resistor $R_L$ the propagation delay is reduced to 10 ns. This configuration shown in Fig[e] is called a *totem-pole* output because transistor *Q4* "sits" upon *Q3.* The TIL gate with the totem-pole output is the same as the open-collector gate except for the output transistor *Q4* and the diode *D.1* When the output *Y* is in the low state *Q2* and *Q3* are drive into saturation as in the open-collector gate. The voltage in the collector of *Q2* is

$$V_{BE}(Q_3) + V_{CE}(Q_2) \text{ or } 0.7 + 0.2 = 0.9V$$

The output $Y = V_{CE}(Q_2)$= 0.2 V. Transistor *Q4* is cut off because its base must be one $V_{BE}$drop plus one diode drop or 2 x 0.6 =1.2 V to start conducting. Since the collector of *Q2*is connected to the base of *Q4* the latter' s voltage is only 0.9 V instead of the required 1.2 V. so *Q4* is cut off. The reason for placing the diode in the circuit is to

provide a diode drop in the output path and thus ensure that *Q4* is cut off when *Q3* is saturated. When the output changes to the high state because one of the inputs drop to the low state, transistors *Q2* and *Q3* go into cutoff.

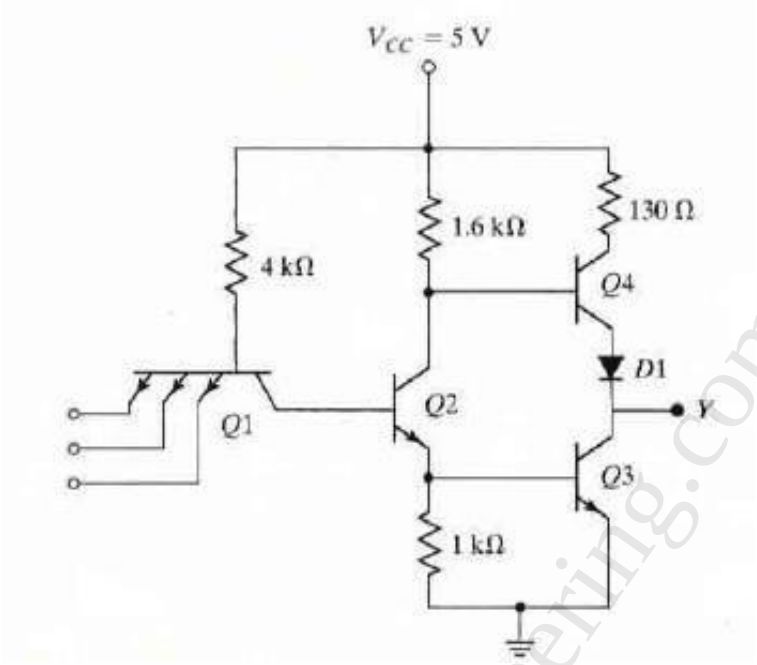



**Fig.[e]TTL gate with totem-pole output**

However the output remains momentarily low because the voltages across the load capacitance cannot change instantaneously. As soon as *Q2* turns off *Q4* conducts and the current needed to charge the load capacitance causes *Q4* to saturate momentarily. As a consequence, the transition from the low to high level is much faster. As the capacitive load charges, the output voltage rises and the current in Q4 decreases bringing the transistor into the active region. The wired-logic connection is not allowed with totem-pole output circuits. When two totem poles are wired together with the output of one gate high and the output of the second gate low, the excessive amount of current drawn can produce enough heat to damage the transistors in the circuit.

## Three-State Gate

The outputs of two TTL gates with to tem-pole structures cannot be connected together as in open-collector outputs. There is, however, a special type of totem-pole gate that allows the wired connection of outputs for the purpose of forming a common-bus system. When a totem-pole output TTL gate has this property, it is called a *three-state* gate. A three -state gate exhibits three output states:

(1) a low-level state when the lower transistor in the totem pole is on and the upper transistor is off

(2) a high -level state when the upper transistor in the totem pole is on and the lower transistor is off

(3) a third state when both transistors in the totem pole are off. The third state is an open-circuit, or high-impedance, state that allows a direct wire connection of many outputs to a common line.

Three-state gates eliminate the need for open-collector gates in bus configurations. Figure (g-a) shows the graphic symbol of a three-state buffer gate. When the control input C is high, the gate is enabled and behaves like a normal buffer, with the output equal to the input binary value. When the control input is low, the output is an open circuit, which gives high impedance (the third state) regardless of the value of input *A.* Some three-state gates produce a high -impedance state when the control input is high. This is shown symbolically in Fig. (g-b), where we have two small circles, one for the inverter output and the other to indicate that the gate is enabled when C is low.

The circuit diagram of the three-state inverter is shown in Fig. [g-c]. Transistors *Q6, Q7* and *Q8* associated with the control input form a circuit similar to the open-collector gate. Transistors *Q1- Q5* associated with the data input, form a totem-pole TTL circuit. The two circuits are connected together through diode *D1*When the control input is high transistor *Q8* turns on and the current flowing from *Vcc* through diode *D*1 causes transistor *Q8* to saturate.

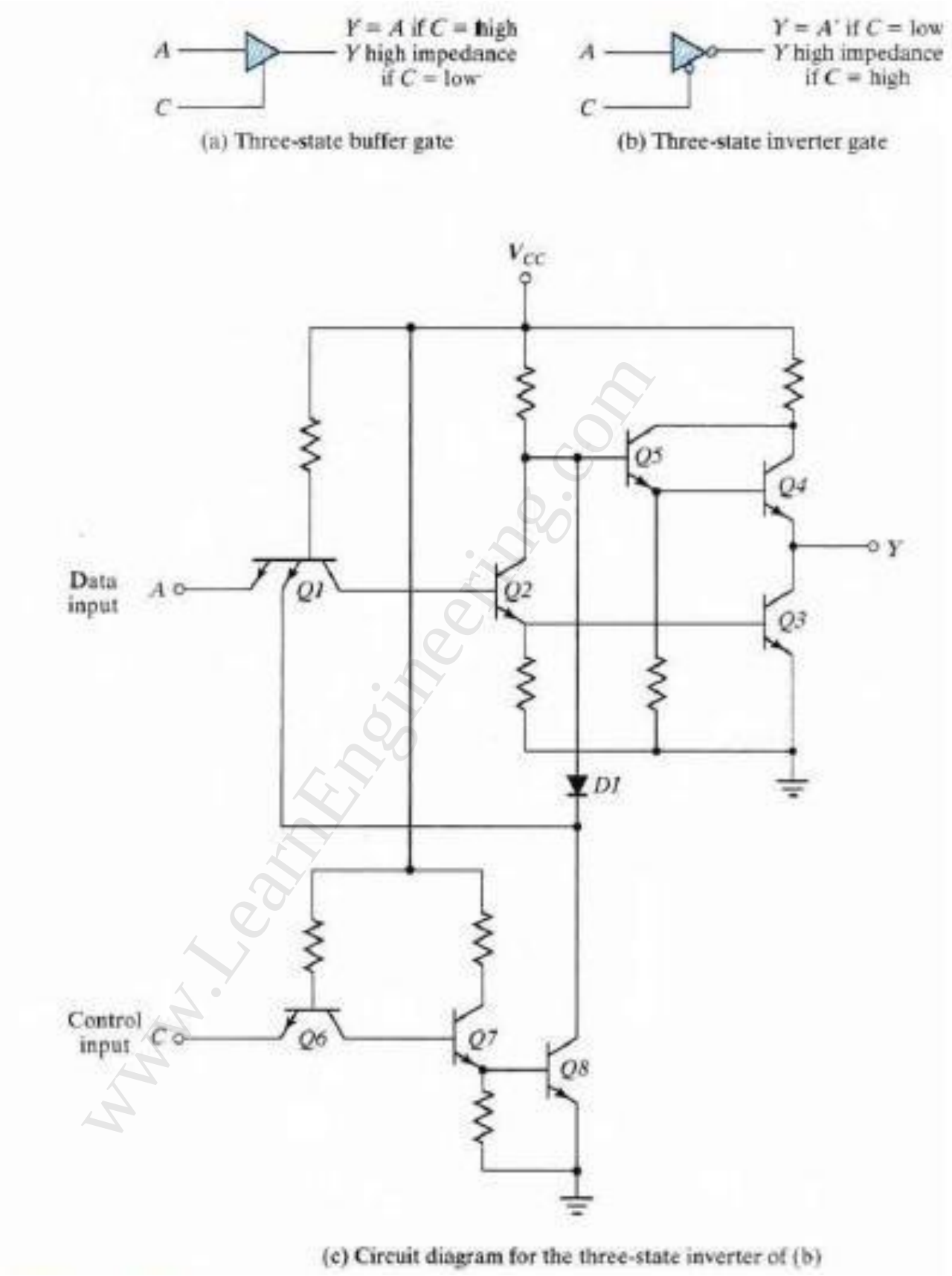




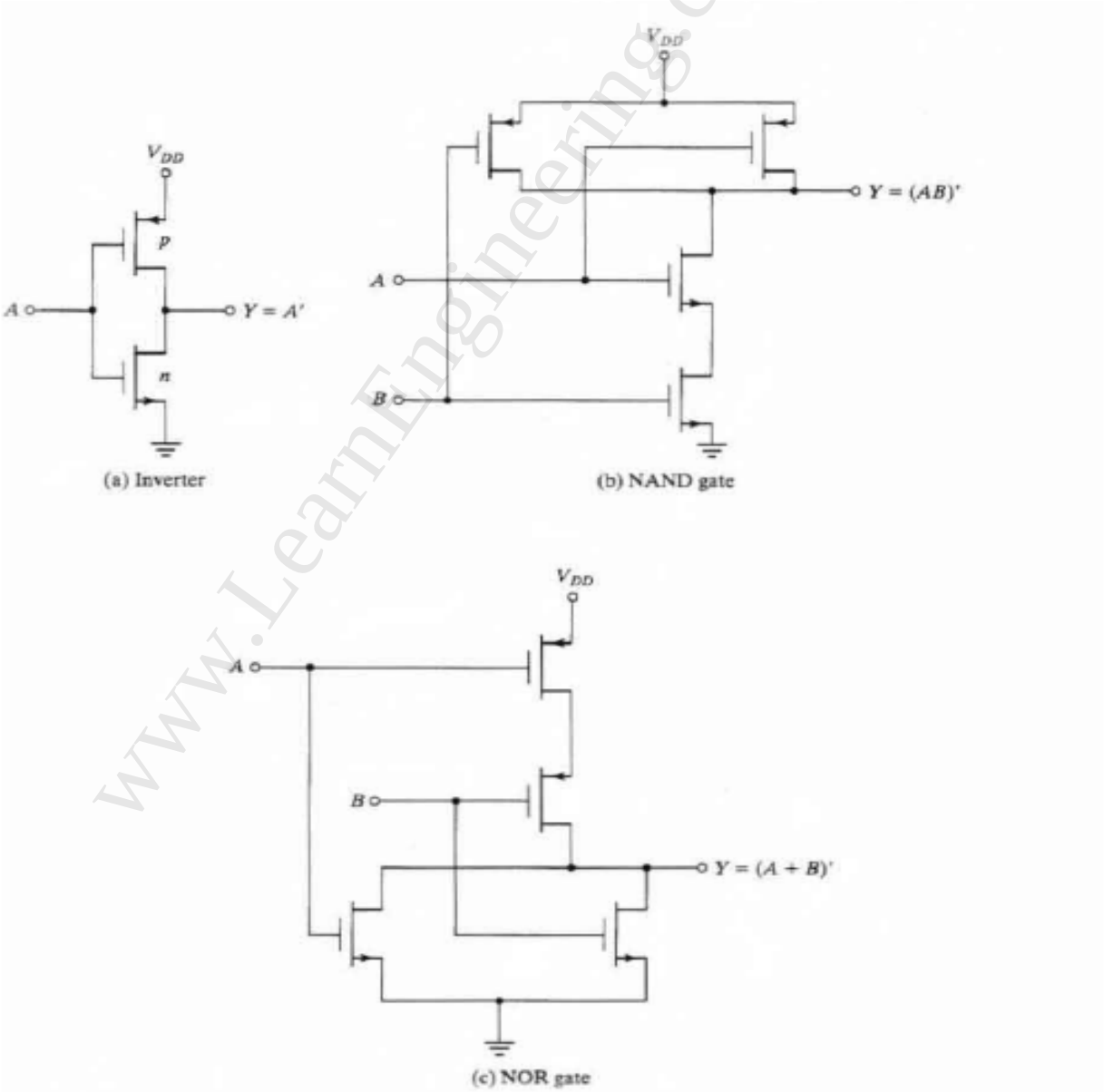

**Fig.[g]Three state TTL gate**

**6. Explain the characteristics of CMOS logic family.**

Complementary MOS (CMOS) circuits take advantage of the fact that both n-channel and p -channel devices can be fabricated on the same substrate. CMOS circuits consist of both types of MOS devices interconnected to form logic functions. The basic circuit is the inverter, which consists of one p-channel transistor and one n-channel transistor as shown in Fig. [a]. The source terminal of the p-channel device is at VDD and the source terminal of the n-channel device is at ground. The value of VDD may be anywhere from +3 to +18 V. The two voltage levels are 0v for the low level and VDD for the high level 5 V.

**Fig.[a]CMOS Logic gates**



## CMOS Characteristics

The voltage transfer curve for a typical CMOS logic gate is shown in Fig[b]. Note that the curves in the transition region are almost vertical. This narrow transition region is the reason for CMOS logic's high noise immunity. Not much voltage range is covered in the transition from one state to the other. In contrast to TTL devices, the threshold voltage depends on the supply voltage and is approximately half the supply voltage.

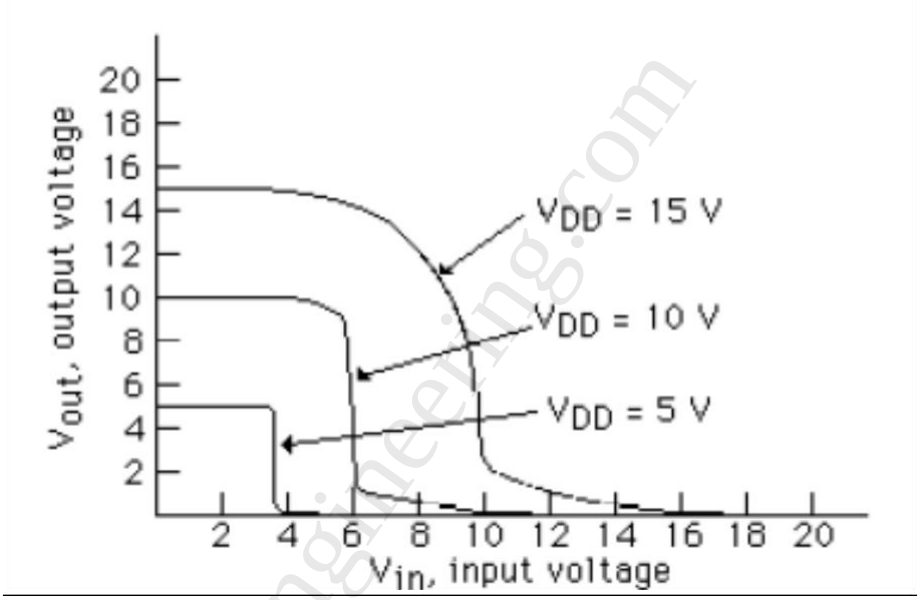



**Fig.[b]CMOS Voltage Transfer Curve**

As with TTL logic, **current spiking** occurs during switching. Hence, bypass capacitors are used in CMOS logic design as well. However, they are not as critical as in TTL logic design because of CMOS's high noise immunity.

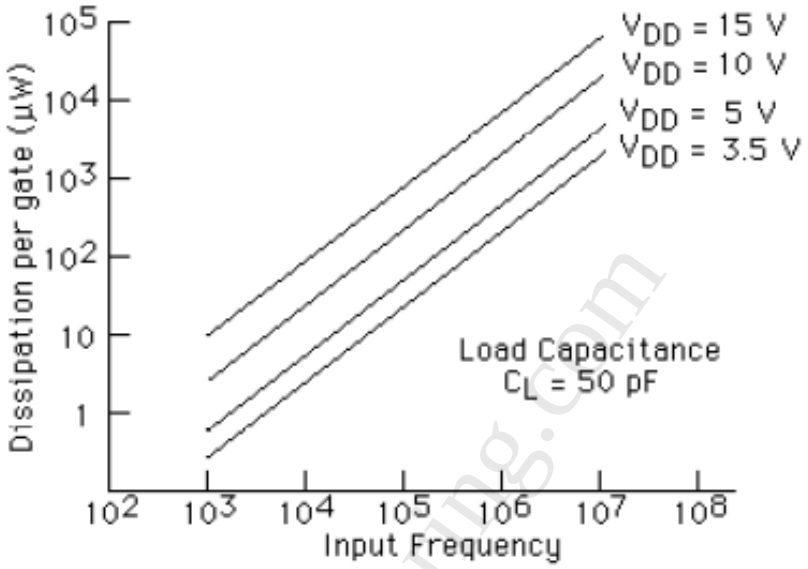
Whereas the typical quiescent (static) **power dissipation** of TTL IC's was about 40 mW, the power dissipation of CMOS IC's are typically 25 nW. However, as the frequency of switching increases, dynamic power dissipation becomes important, as illustrated in Fig[c]. Above 1 MHz, the dynamic power dissipation predominates and can exceed TTL power dissipation.

**The input impedance**, in either state, of CMOS gates is typically $10^{12}$ Ω. The input capacitance is 5 pF. The output impedance depends on the particular device, and is on the order of 1 kΩ, for either state.

The **propagation delay** times for CMOS devices are relatively long due to their high output impedance. Typical delay times are 60 nsec for VDD = 5 V, and 25 nsec for

VDD = 10 V. Doubling the supply voltage more than doubles the speed of a CMOS gate. The rise and fall transition times are typically 70 nsec. for VDD = 5 V. Thus CMOS devices operate significantly slower than TTL devices.




**Fig[c] Dynamic Power Dissipation of a Typical CMOS Logic Gate**

The HIGH and LOW **noise margins** for $V_{DD}$ = 5 V, are DC0 = $V_{IL}$ - $V_{OL}$ = 1.5 - 0.01 ≈ 1.5 V, and DC1 = $V_{OH}$ - $V_{IH}$ = 4.99 V - 3.5 V ≈ 1.5 V. In general, noise immunity is a minimum of 30% of $V_{DD}$, and typically 45% of $V_{DD}$. Thus, CMOS devices are good in noisy environments such as automobiles and industrial plants. The HIGH and LOW noise margins are essentially equal because the output impedance, output voltage, and threshold voltages are symmetrical with respect to the supply voltage.

**Floating inputs** in CMOS logic guarantee neither LOW nor HIGH outputs and cause increased susceptibility to noise, as well as excessive power dissipation. Hence, all unused inputs should be connected to VDD or VSS, as appropriate.

The **fan-out** of CMOS devices is usually greater than 50 because the input current requirement of CMOS logic is nil. However, current is required to charge and discharge the capacitance of CMOS inputs during logic transitions. Hence, the greater the fan-out the longer the propagation delay. For example, with VDD = 5 V, the propagation delay will increase from 60 nsec when the output drives 1 input to 150 nsec when the output drives 10 inputs. As a rule of thumb, you can assume the load will be 5 pF per CMOS input plus 5 to 15 pF for stray wiring capacitance.

Outputs of CMOS gates, like those of totem-pole TTL gates, should never be connected together. Also, the power supply should be turned on before applying any logic signals to a CMOS device and the logic signals should be removed before turning off the power supply, otherwise the device could be damaged.

# UNIT II- COMBINATIONAL CIRCUITS

## PART A

1. **Write the design procedure of combinational circuits.**

   The procedure involves the following steps,

   i. The problem is stated
   ii. The number of available input variables and required output variables is determined.
   iii. The input and output variables are assigned letter symbols.
   iv. The truth table that defines the required relationships between inputs and outputs is derived.
   v. The simplified Boolean function for each output is obtained.
   vi. The logic diagram is drawn.

2. **Compare serial adder and parallel adder.**

| S.No | Parallel adder | Serial adder |
|---|---|---|
| 1 | The Parallel adder uses registers with parallel load. | Serial adder uses shift registers. |
| **2** | The number of full-adder circuits in the parallel adder is equal to the number of bits. | Serial adder requires only full-adder circuits and a carry flip-flop. |
| **3** | Parallel adder is combinational circuit. | Serial adder is a sequential circuit. |
| **4** | Parallel adder is faster than serial adder. | Serial adder is slower than parallel adder. |

3. **Define Half adder & Full adder.**

   - The logic circuit that performs the addition of two bits is a half adder.
   - The logic circuit that performs the addition of three bits is a full adder.

4. **Give the logic expressions for sum and carry in Full adder circuit.**

   $S = x \oplus y \oplus z$

   $C = xy + yz + zx$

5. **What is priority encoder? (MAY/JUN 2014)**

   It is an encoder circuit that includes the priority function. The function of the priority encoder is such that if two or more inputs are equal to 1,the input having the highest priority will take precedence.

6. **Give the logic expressions for sum and carry in Full adder circuit.(NOV/DEC 2016)**

   S = x y z

   C = xy + yz +zx

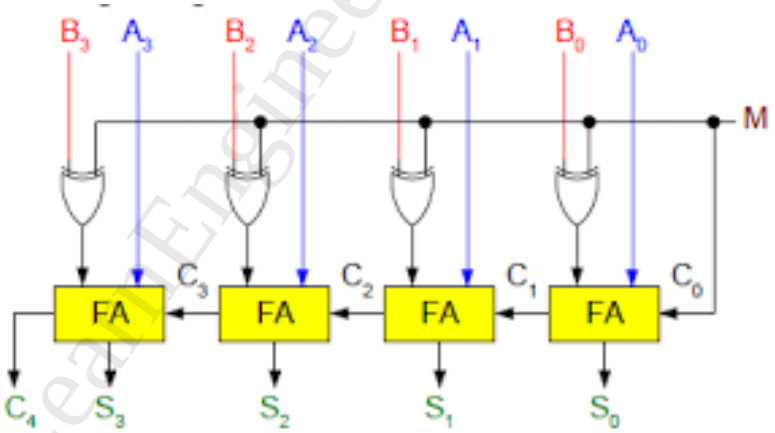
7. **Give any four examples for combinational circuits. (NOV/DEC 2013)**

   i. Half adder, Full adder

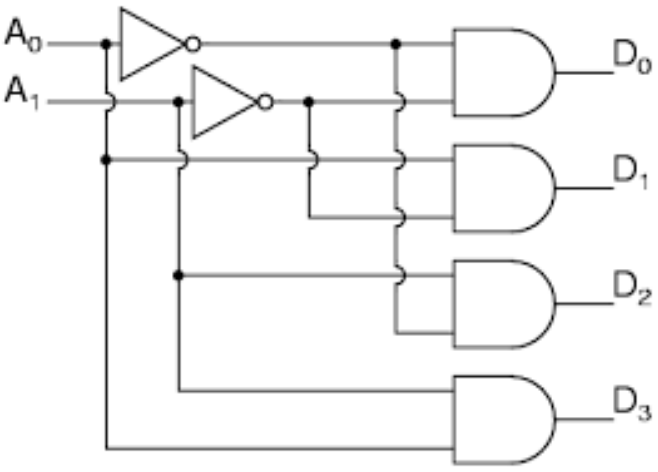
   ii. Magnitude Comparator

   iii. Decoder and Encoder

8. **Construct 4-bit parallel adder/subtractor using Full adder & X-OR gates. (NOV/DEC 2014)**

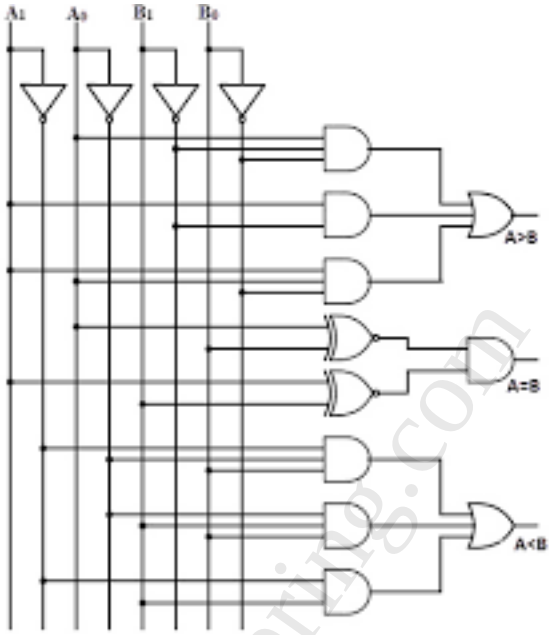



9. **Convert a 2 to 4 line decoder with enable input to 1x 4 Demultiplexer. (NOV/DEC 2014)**

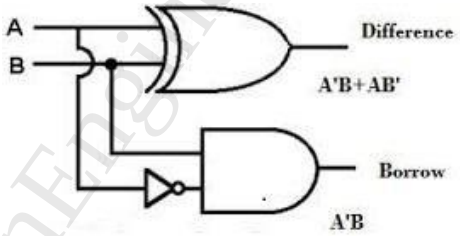


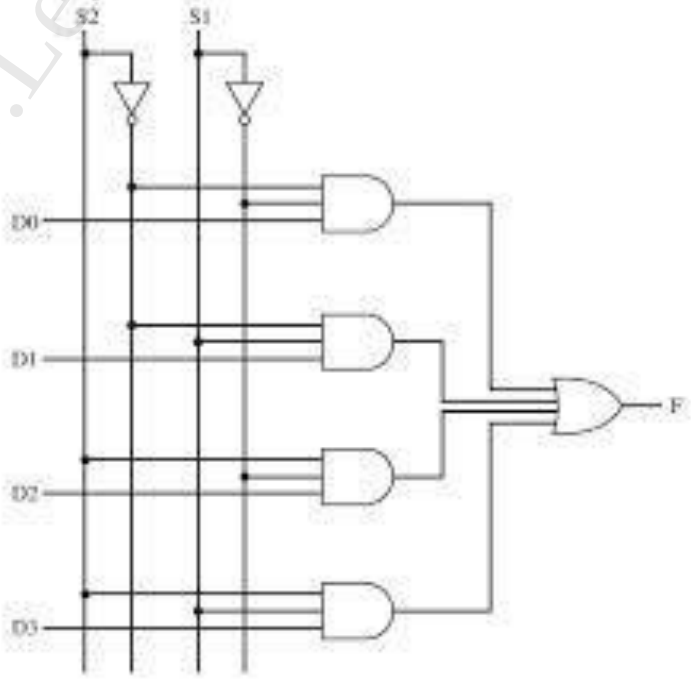
**10. Draw the logic circuit of 2-bit comparator. (APR/MAY 2015), (MAY/JUN 2014)**




**11. Design a Half Subtractor using basic gates. (MAY/JUN 2013)**




**12. Draw the logic diagram of 4 line to 1 line multiplexer. (MAY/JUN 2013)**



**13. List out various application of multiplexer. (NOV/DEC 2013)**

Multiplexers are used in data selection, data routing, operations sequencing, parallel to serial conversion, waveform generation and logic function generation.

**14. Write the difference between DEMUX and DECODER. (APR/MAY 2015)**

| S.NO | DEMUX | DECODER |
|---|---|---|
| 1 | It is a digital circuit that receives information on a single line and transmits the information on one of the several output lines. It consists of a one input line, 'n' selection lines and '$2^n$' output lines. | A decoder is a combinational circuit that converts n number of input lines to a maximum of $2^n$ number of output lines. |
| 2 | It contains 'n' selection lines. | Selection lines are not present. |

**15. Write the expression for borrow and difference in Full Subtractor circuit.**

Borrow B = $\bar{x}$ y +$\bar{x}$z+ yz

Difference D = x⊕ y ⊕ z

**16. Write about the design procedure for combinational circuits (NOV/DEC 2013)**

1.From the given specification determine the number of inputs and outputs and assign a symbol to each input and output.

2.Derive the truth table that defines the required relationship between inputs and outputs.

3.Obtain the Boolean function for each output as a function of the input variables using Karnaugh Map

4.Draw the logic diagram based on the Boolean function obtained in step 3.

## PART B

**1. Explain about Carry look ahead adder**

Most other arithmetic operations, e.g. multiplication and division are implemented using several add/subtract steps. Thus, improving the speed of addition will improve the speed of all other arithmetic operations.

Accordingly, reducing the carry propagation delay of adders is of great importance. Different logic design approaches have been employed to overcome the carry propagation problem.

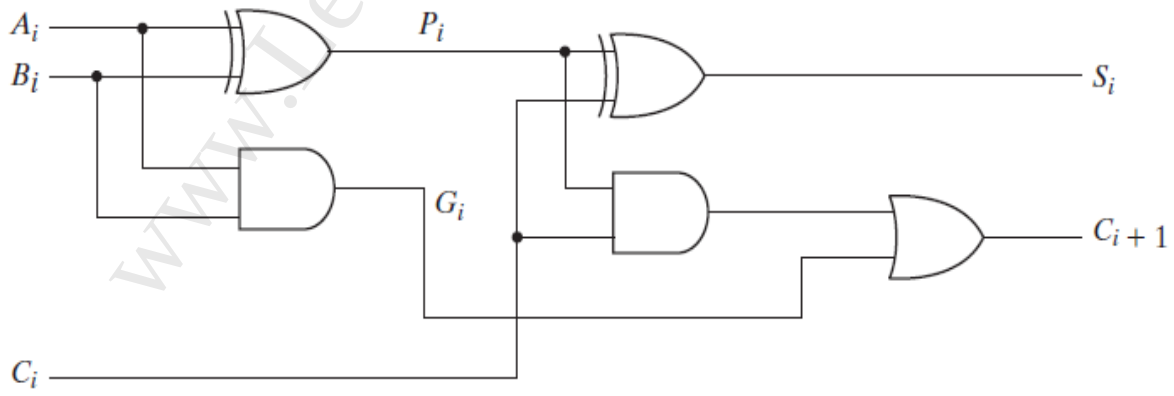
One widely used approach employs the principle of carry look-ahead solves this problem by calculating the carry signals in advance, based on the input signals. This type of adder circuit is called as **carry look-ahead adder** (CLA adder). It is based on the fact that a carry signal will be generated in two cases:

(1) when both bits Ai and Bi are 1, or

(2) when one of the two bits is 1 and the carry-in (carry of the previous stage) is 1

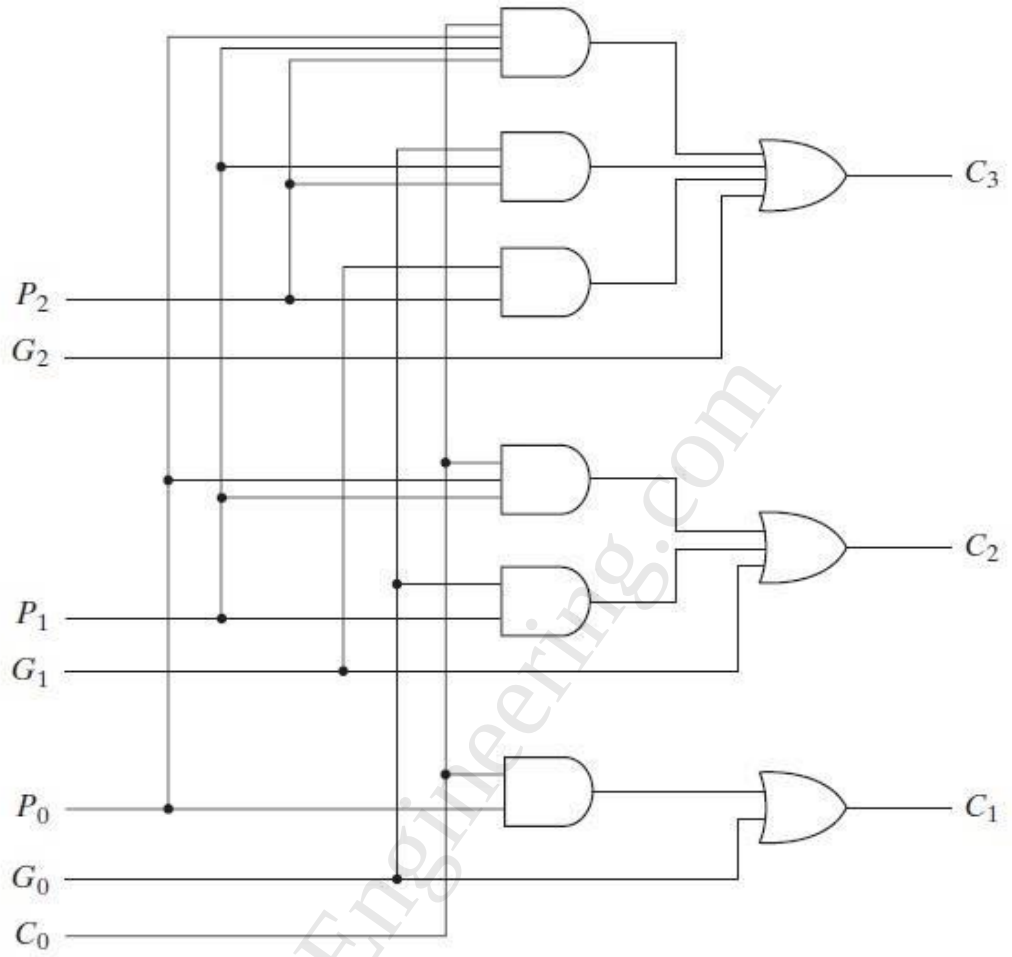
Carry look ahead adder is the most widely used technique for reducing the propagation time in parallel adder. The solution for reducing carry propagation delay time is to employ faster gates with reduced delays.

Full adder Circuit:



Logic Diagram of a look ahead carry generator




Expression:

$Pi = Ai \oplus Bi$

$Gi = Ai\ Bi$

The output sum and carry can be defined as :

$Si = Pi \oplus Ci$

$C_{i+1} = Gi + Pi\ Ci$

$C_1 = G_0 + P_0C_0$

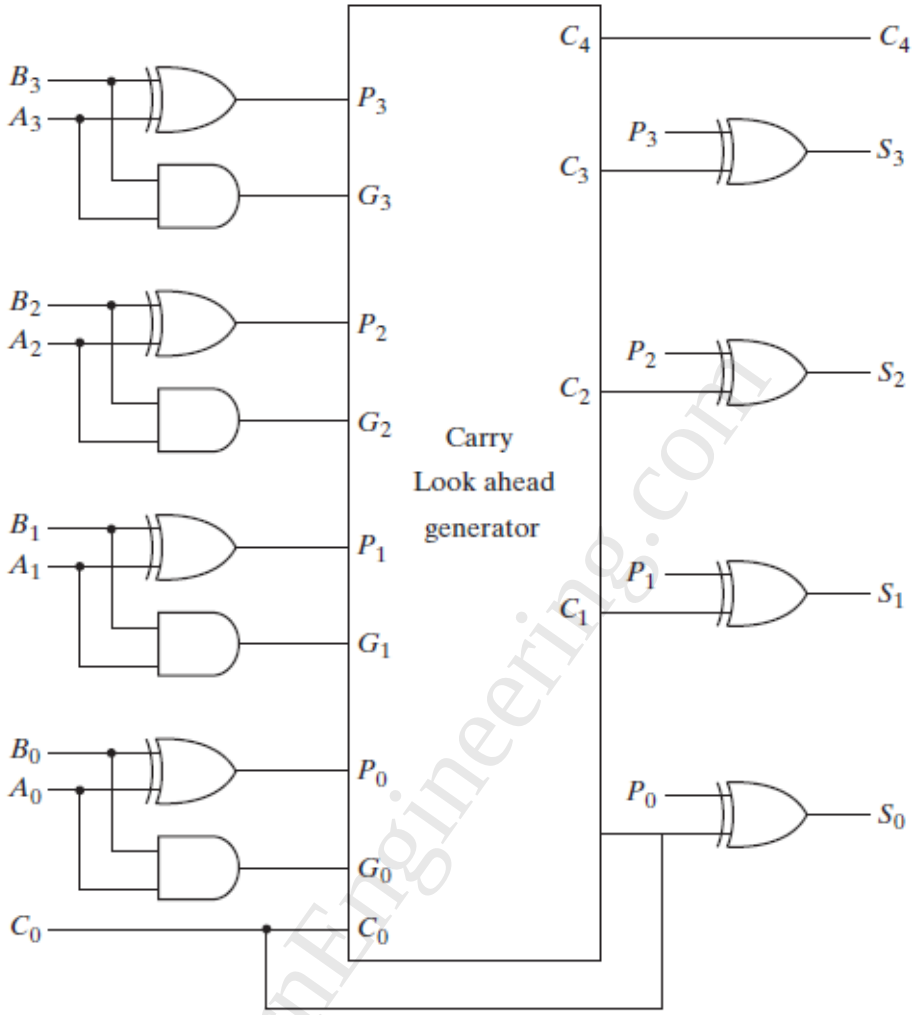
$C_2 = G_1 + P_1C_1 = G_1 + P_1 (G_0 + P_0C_0)$

$= G_1 + P_1G_0 + P_1P_0C_0$

$C_3 = G_2 + P_2C_2 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$

4 bit full adders with look ahead carry




**2. Design a 4 bit Magnitude Comparator and draw the circuit.**

Definition:

Magnitude comparator is a combinational circuit that compares TWO numbers, A and B, and then determines their relative magnitudes.

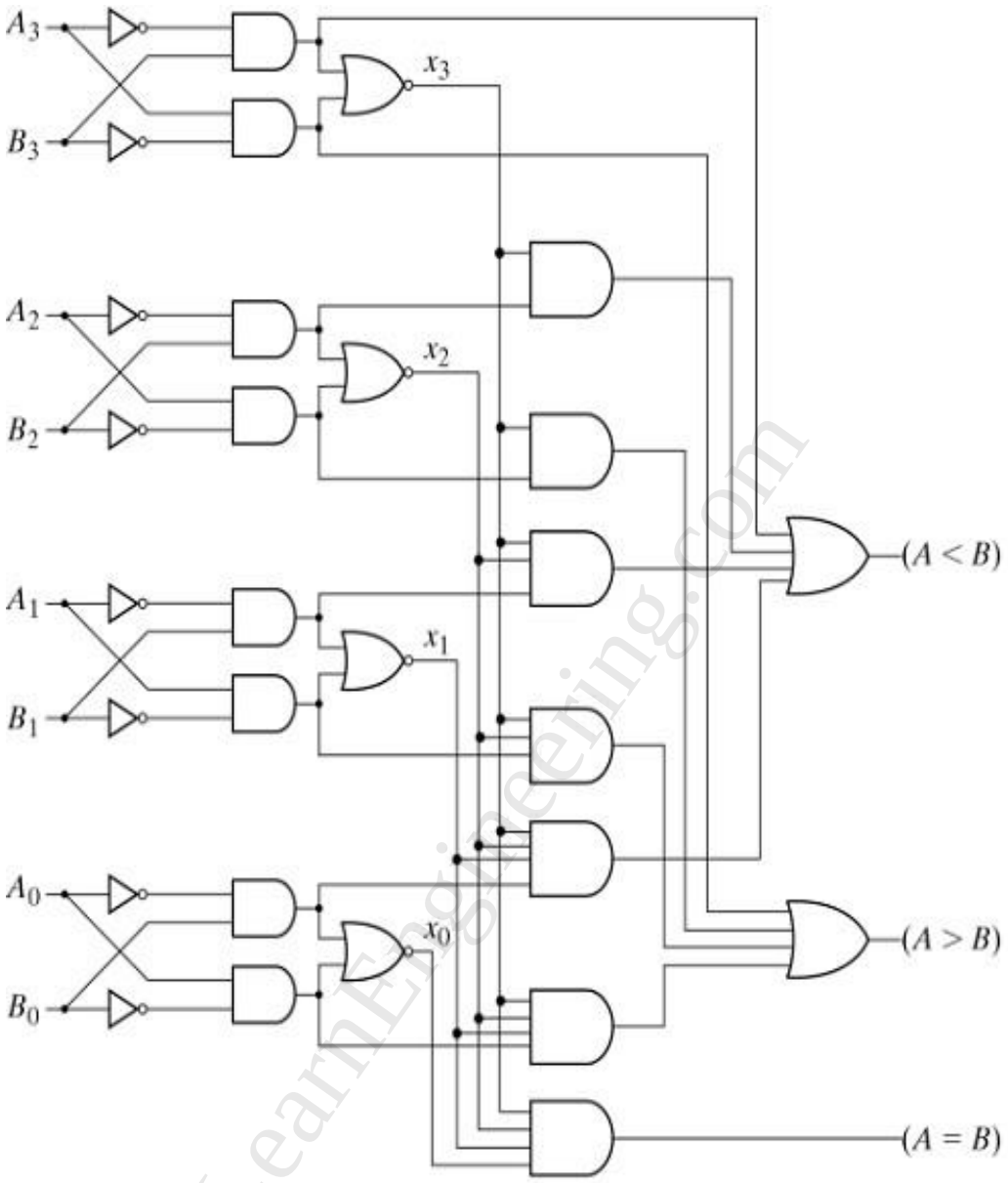
A > B

A = B

A < B

Logic Diagram




Logic Expression

$(A=B) = x_3x_2x_1x_0$

$(A>B) = A_3B'_3+x_3A_2B'_2 + x_3x_2A_1B'_1 +x_3x_2x_1A_0B'_0$

$(A<B) = A'_3B_3+x_3A'_2B_2 + x_3x_2A'_1B_1 +x_3x_2x_1A'_0B_0$

Explanation

The comparison of two numbers is an operation that determines if one number is greater than, less than, or equal to the other number.

The circuit for comparing two n-bit numbers has $2^n$ entries in the truth table.

$A=A_3A_2A_1A_0$

$B= B_3B_2B_1B_0$

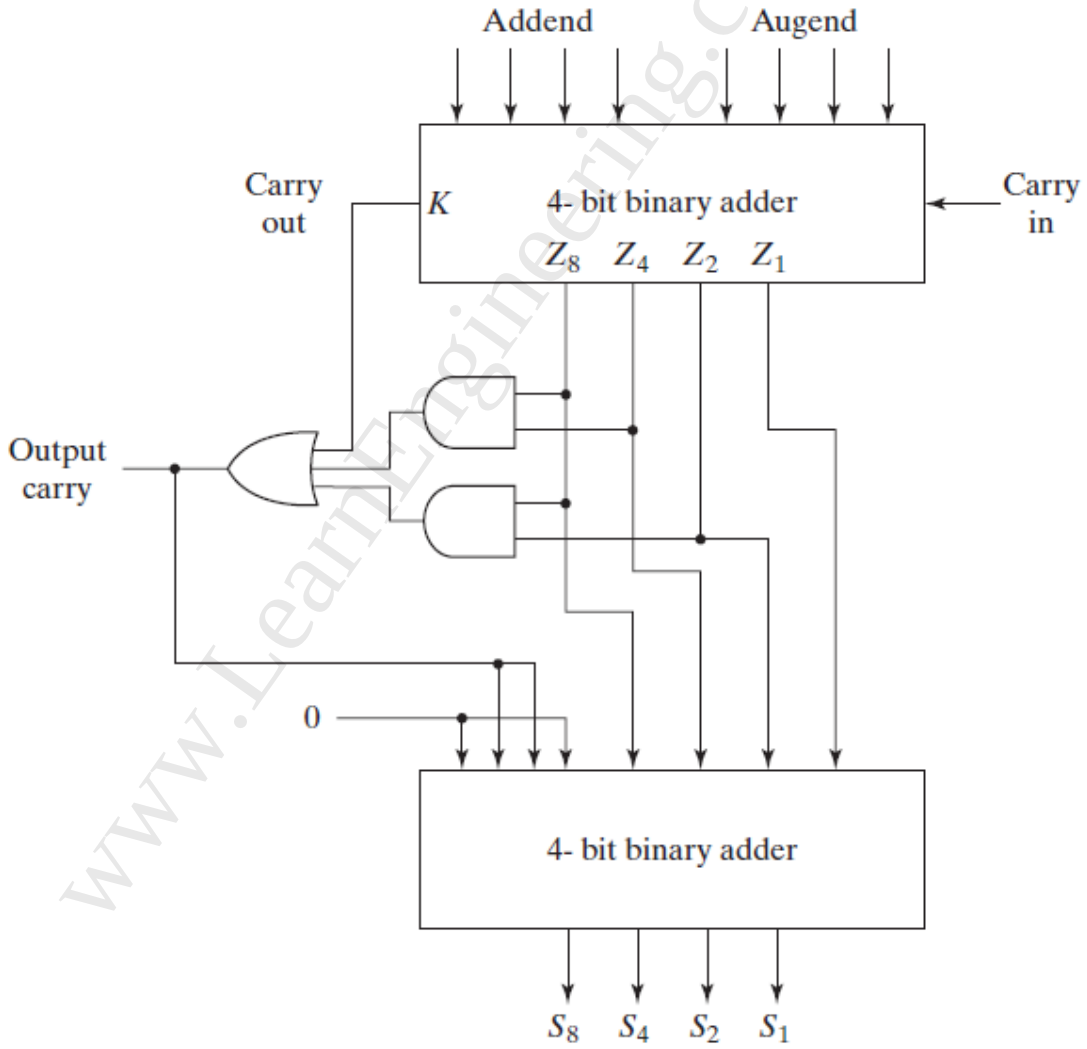
$x_i = A_iB_i + A_i'B_i'$

**3. Design a BCD Adder and explain its working with necessary circuit diagram.**

Definition

- BCD adder that adds two BCD digits and produces sum digit in BCD.
- In BCD each number is defined by a binary code of 4 bits.

Block Diagram




Expression:

$C = K + Z_8Z_4 + Z_8Z_2$

Truth Table:

| Binary Sum | | | | | BCD Sum | | | | | Decimal |
|---|---|---|---|---|---|---|---|---|---|---|
| K | $Z_8$ | $Z_4$ | $Z_2$ | $Z_1$ | C | $S_8$ | $S_4$ | $S_2$ | $S_1$ | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 3 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 4 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 5 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 6 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 7 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 8 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 9 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 10 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 11 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 12 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 13 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 14 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 15 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 16 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 17 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 18 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 19 |

Explanation:

➢ The adder will form the sum in binary and produce a result which may range from 0 to 19.These binary numbers are listed in the truth table and are labeled by the symbols K,$Z_8$,$Z_4$,$Z_2$ and $Z_1$. K is the carry .

➢ Z represent the weights 8,4,2 and 1that can be assigned to the 4 bits in the BCD adder.

➢ The output sum of two decimal digits must be represented in BCD.

**4. Explain about Binary Multiplier.**

➢ Multiplication of binary numbers is performed in the same way as with decimal numbers.

➢ The multiplicand is multiplied by each bit of the multiplier, starting from the least significant bit.The result of each such multiplication forms a partial product. Successive partial products are shifted one bit to the left.

➢ The product is obtained by adding these shifted partial products.

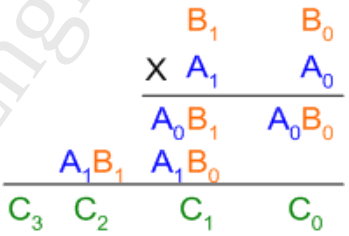
**Example 1:**

Consider an example of multiplication of two numbers,

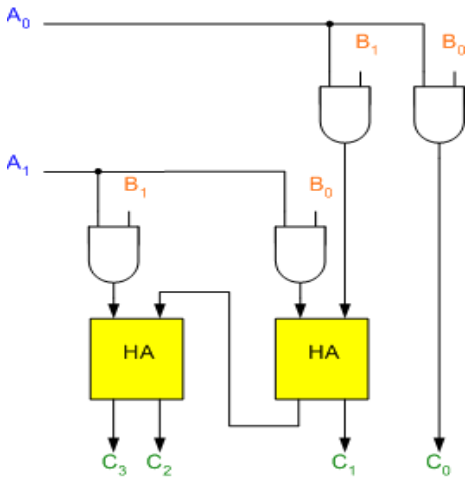
say A and B (2 bits each), C = A x B.

The first partial product is formed by multiplying the B1B0 by A0. The multiplication of two bits such as A0 and B0 produces a 1 if both bits are 1; otherwise it produces a 0 like an AND operation. So the partial products can be implemented with AND gates.

The second partial product is formed by multiplying the B1B0 by A1 and is shifted one position to the left.




The two partial products are added with two half adders (HA). Usually there are more bits in the partial products, and then it will be necessary to use FAs.



The least significant bit of the product does not have to go through an adder, since it is formed by the output of the first AND gate as shown in the Figure.

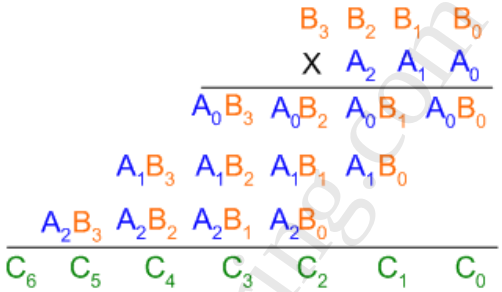
A binary multiplier with more bits can be constructed in a similar manner.

**Example 2:**

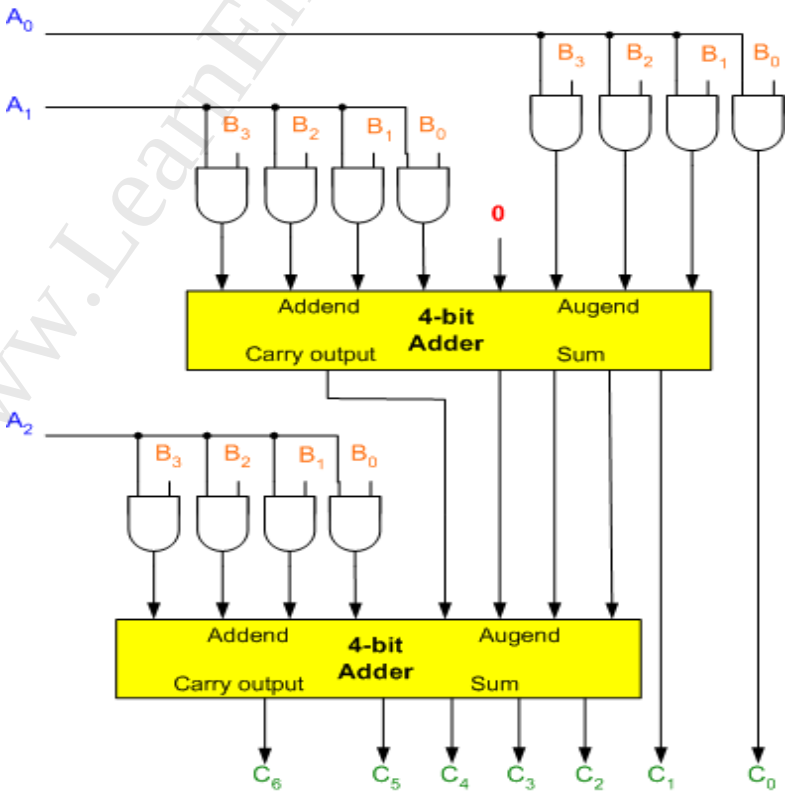
Consider the example of multiplying two numbers,

say A (3-bit number) and B (4-bit number).

Each bit of A (the multiplier) is ANDed with each bit of B (the multipcand) as shown in the Figure.




The binary output in each level of AND gates are added in parallel with the partial product of the previous level to form a new partial product. The last level produces the final product



Since J = 3 and K = 4, 12 (J x K) AND gates and two 4-bit ((J - 1) K-bit) adders are needed to produce a product of seven (J + K) bits. Its circuit is shown in the Figure.

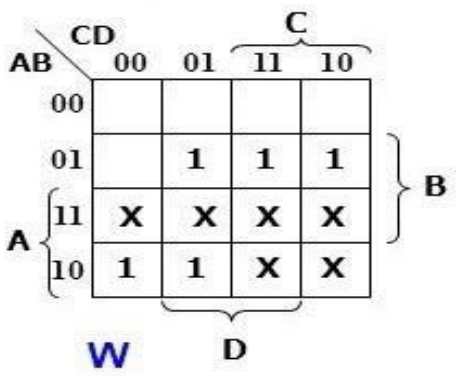
Note that 0 is applied at the most significant bit of augend of first 4-bit adder because the least significant bit of the product does not have to go through an adder.

**5. Design a 4 bit BCD to EXCESS 3 code converter. Draw the logic diagram.**

**Truth Table:-**

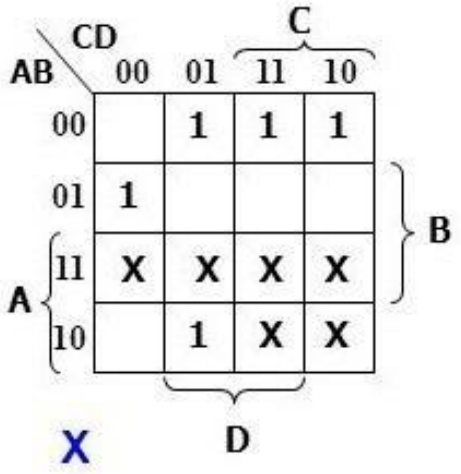
| | BCD | | | | Excess-3 | | | |
|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | W | X | Y | Z |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 3 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 8 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 10 | 1 | 0 | 1 | 0 | X | X | X | X |
| 11 | 1 | 0 | 1 | 1 | X | X | X | X |
| 12 | 1 | 1 | 0 | 0 | X | X | X | X |
| 13 | 1 | 1 | 0 | 1 | X | X | X | X |
| 14 | 1 | 1 | 1 | 0 | X | X | X | X |
| 15 | 1 | 1 | 1 | 1 | X | X | X | X |

**K-map simplification:-**

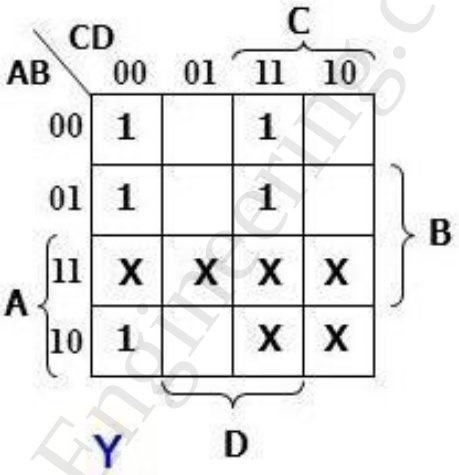



W = A + BC + BD

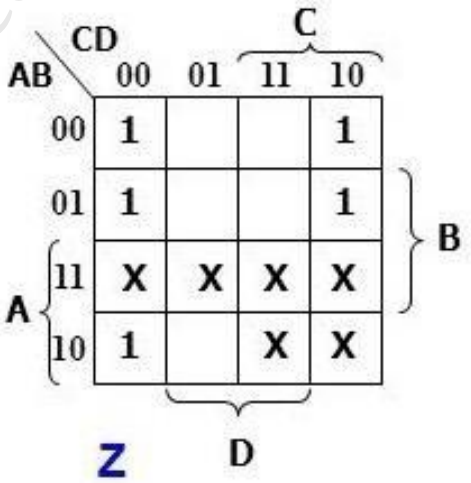
 




X =B'C + BC'D' + BD



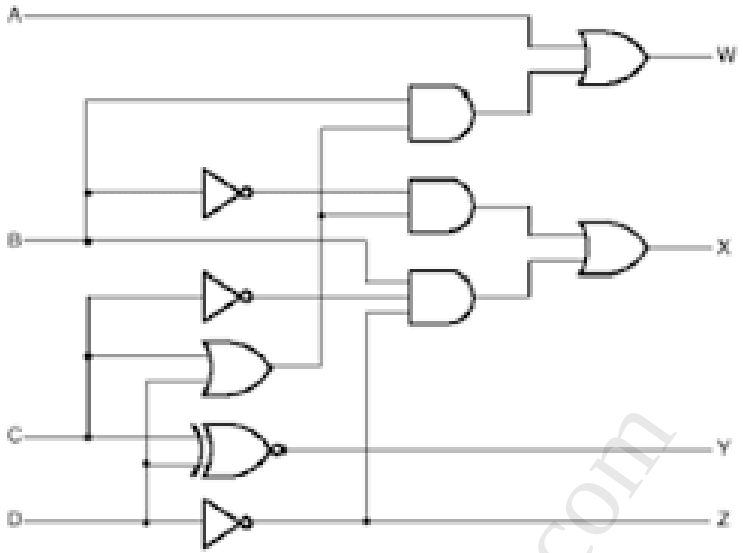

Y = CD + C'D'




Z = D'

**Logic Diagram:-**

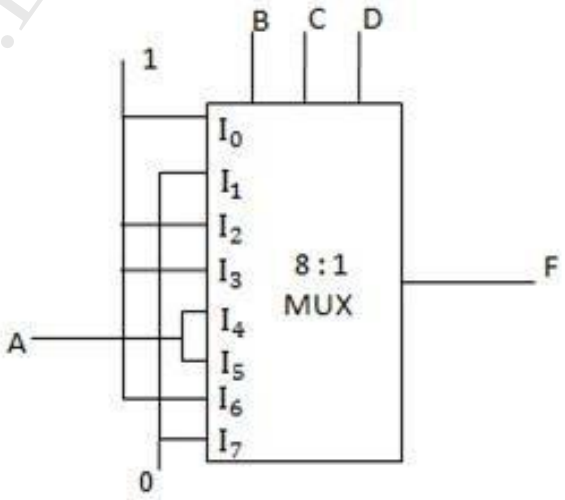



**6. Implement F(A,B,C,D) =$\Sigma_m$(0,2,3,6,8,10,11,12,13,14)using 8x1 multiplexer.**

**Implementation table:**

| | $I_0$ | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | $I_6$ | $I_7$ |
|---|---|---|---|---|---|---|---|---|
| $\bar{A}$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| A | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| | 1 | 0 | 1 | 1 | A | A | 1 | 0 |

**Logic Diagram:-**



## UNIT III- SEQUENTIAL CIRCUITS

## PART A

1. **Differentiate synchronous and asynchronous sequential circuit.**

| Synchronous sequential circuits | Asynchronous sequential circuits. |
|---|---|
| Memory elements are clocked flip-flops | Memory elements are either unlocked flip flops or time delay elements |
| Easier to design | More difficult to design |

2. **Define latches (NOV/DEC 2013)**

- It is a sequential circuit that checks all of its inputs continuously and changes its output at any time. Many times enable signal is provided with the latch.
- Latch is a bistable element that has two stable states logic 0 and logic 1.

3. **What are the classifications of Sequential circuits.**

The sequential circuits are classified on the basis of timing of their signals into two types. They are,

1) Synchronous sequential circuit.

2) Asynchronous sequential circuit.

4. **What is edge triggered flip-flop?**

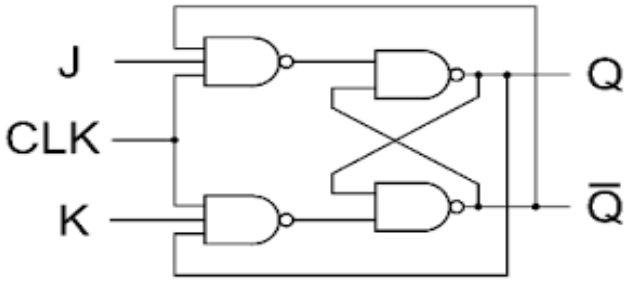
The problem of race around condition can solved by edge triggering flip flop. The term edge triggering means that the flip-flop changes state either at the positive edge or negative edge of the clock pulse and it is sensitive to its inputs only at this transition of the clock.

5. **How many flip flop are required to build a binary counter that count from 0 to 1023. (APR/MAY 2015), (MAY/JUN 2013)**

The no of flip flop required to build a binary counter is 10.because the binary equivalent of 1023 is 10 bit.

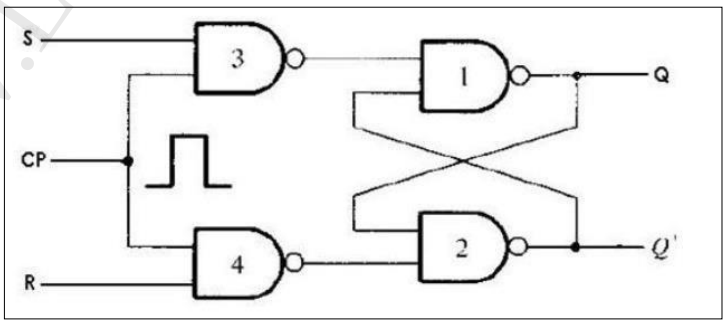
6. **Realize JK flip flop. (NOV/DEC 2014)**




7. **Compare the logic of synchronous counter and ripple counter. (NOV/DEC 2014)**

| **Synchronous counter** | **Ripple counter.** |
|---|---|
| There is no connection between output of first flip-flop and clock input of next flip-flop. | Te output of first flip-flop drives the clock for the next flip-flop. |
| All the flip-flops are clocked simultaneously | All the flip-flops are not clocked simultaneously |
| Design involves complex logic circuit as number of states increases. | Logic circuit is very simple even for more number of states. |

8. **Draw the logic diagram of clocked SR Flip flop. (MAY/JUN 2014)**



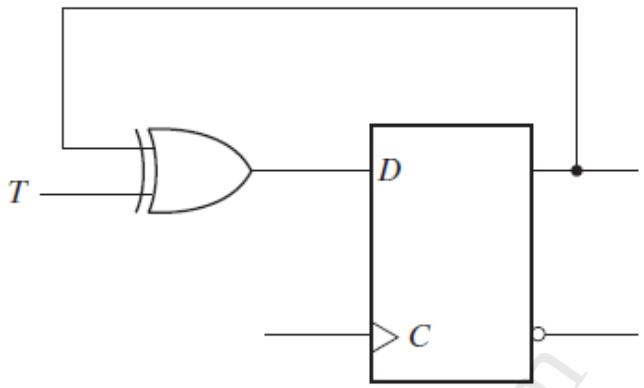

9. **What is modulo-N counter?**

A modulo-n counter will count n states. For example a mod-6 will count the sequence 000,001,010,011,100,101 and then recycles to 000.

Mod 6 counter skips 110 and 111 states and it goes through only six different states.

**10. Convert D flip flop to T flip flop. (NOV/DEC 2012)**




$D = T \oplus Q$

**11. Write short notes on digital clock. (NOV/DEC 2013)**

One of the most popular applications of counter is digital clock. A digital clock is a time clock which displays the time of day in hours, minutes and seconds. To construct an accurate digital clock a high clock frequency is required.

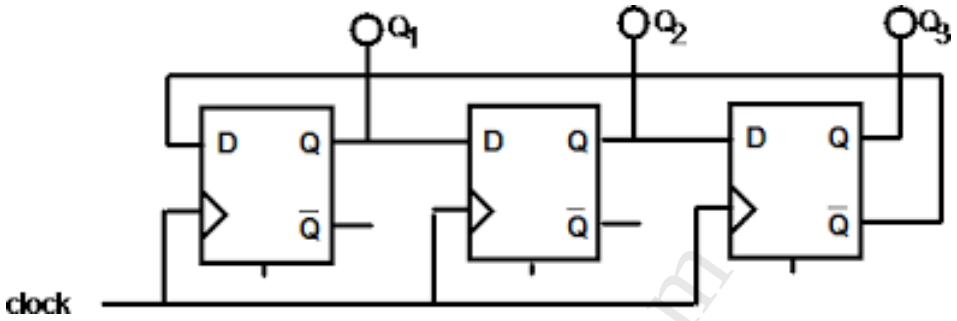
**12. Write the truth table of RS flip flop. (APR/MAY 2015)**

| C | S | R | $Q_n$ | $Q_{n+1}$ | State |
|---|---|---|---|---|---|
| 0 | X | X | 0 | 0 | No Change |
| 0 | X | X | 1 | 1 | |
| □ | 0 | 0 | 0 | 0 | No Change |
| □ | 0 | 0 | 1 | 1 | |
| □ | 0 | 1 | 0 | 0 | RESET |
| □ | 0 | 1 | 1 | 0 | |
| □ | 1 | 0 | 0 | 1 | SET |
| □ | 1 | 0 | 1 | 1 | |
| □ | 1 | 1 | 0 | X | Indeterminate |
| □ | 1 | 1 | 1 | X | |

**13. Design a 3 bit ring counter and find the mod of the designed counter. (NOV/DEC 2012)**

A 3 bit ring counter consists of 3 states. The states are 100,010 and 001



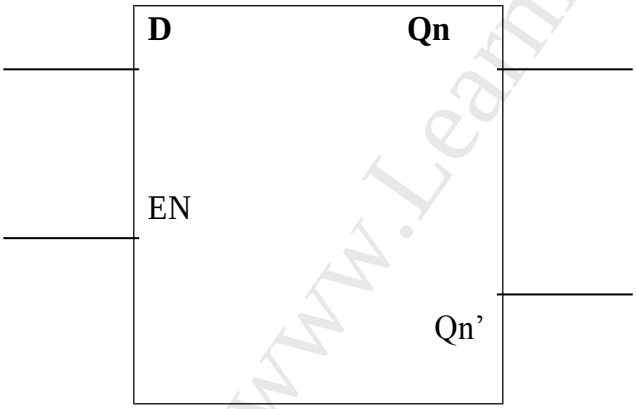

**14. Define – Race Around Condition IN FLIPFLOP (NOV/DEC 2016)**

In a JK flip-flop , when J = K = 1 and for every clock pulse applied the output changes its state.ie. the output toggles for every clock pulse applied. This condition is called as 'race around condition'.

**15. Define Programmable Counter.**

The Counters that have the capability to start counting from any desired state are called programmable counter.

**16. Draw D Latch with truth table**



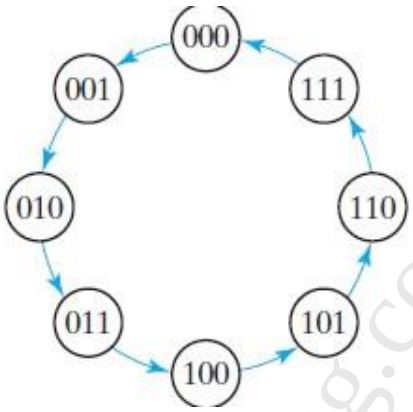

| D | Qn+1 | State |
|---|---|---|
| 0 | 0 | Reset |
| 1 | 1 | Set |

## PART B

**1. Using T Flip-flop design binary counter which counts in the sequence 000, 001, 010, 011, 100, 101, 110, 111, 000**

State Diagram:




Excitation Table:

| Count Sequence | | | Flip-flop inputs | | |
|---|---|---|---|---|---|
| $A_3$ | $A_1$ | $A_0$ | $TA_2$ | $TA_1$ | $TA_0$ |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

Maps for a 3 bit binary counter:

| | | | |
|---|---|---|---|
| | | 1 | |
| | | 1 | |

$T_{A2} = A_1 A_0$

| | | | |
|---|---|---|---|
| | 1 | 1 | |
| | 1 | 1 | |

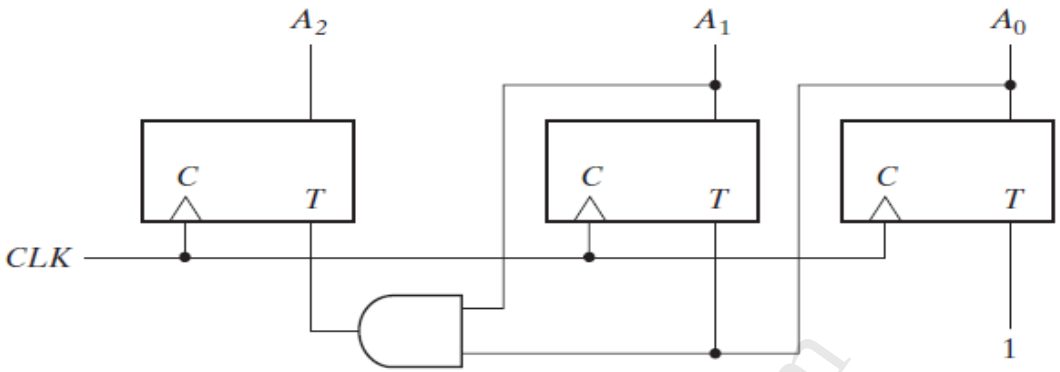
$T_{A1} = A_0$

| | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 |

$T_{A0} = 1$

Logic Diagram




Explanation:

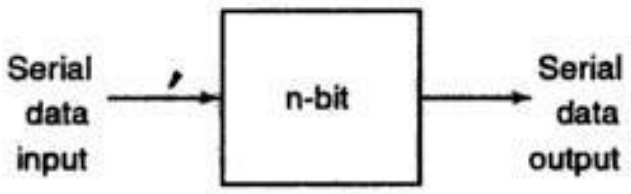
- A Counter that follows the binary sequence is called a binary counter.An n-bit binary counter consists of n flip-flops and can count in binary from 0 to $2^n$ -1.
- State diagram of a counter does not have to show input-output values along the directed lines. The only input to the circuit is the count pulse and the outputs are directly specified by the present state, and the transition state occurs every time the pulse occurs.
- The Count Sequence of a 3 bit binary counter is mentioned in excitation table.
- The flip-flop input functions from the excitation tables are simplified in the maps.
- Including the Boolean functions with the three flip-flops, the logic diagram of the counter is obtained.

**2. Explain about SISO , SIPO, PISO, PIPO Shift Register.**

**SISO (Serial-in to Serial-out)**

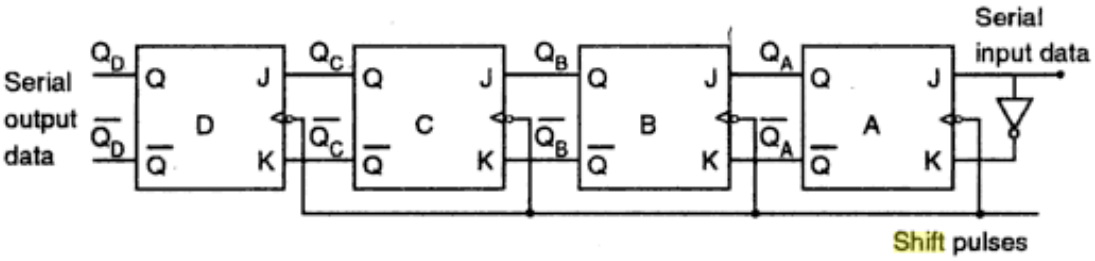
- The data is shifted serially "IN" and "OUT" of the register, one bit at a time in either a left or right direction under clock control.
- It accepts data serially.
- It produces stored information on its single output also in serial form.

Basic Block diagram:



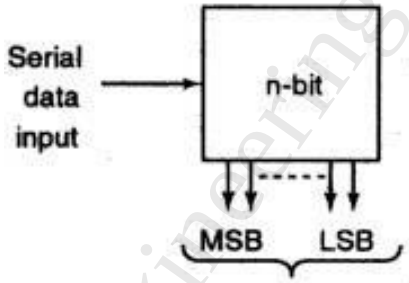
Shift Left Register using JK Flip-flop



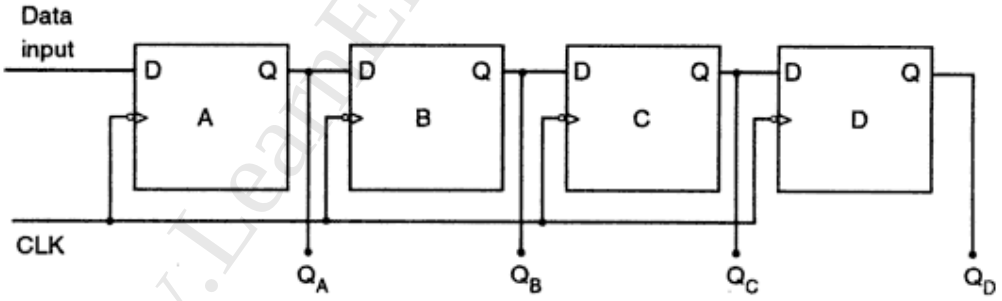

**SIPO (Serial-in to Parallel -out)**

- In this type the input are given in serial form and output are taken in parallel form.
- In this register data is shifted in serial but shifted out in parallel

Basic Block diagram:



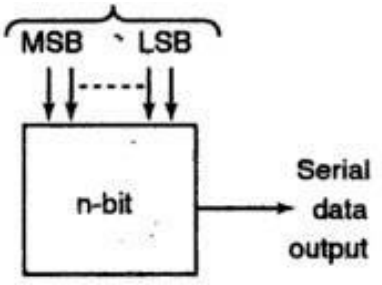

4 bit serial in parallel out shift register




**PISO (Parallel -in to Serial-out)**

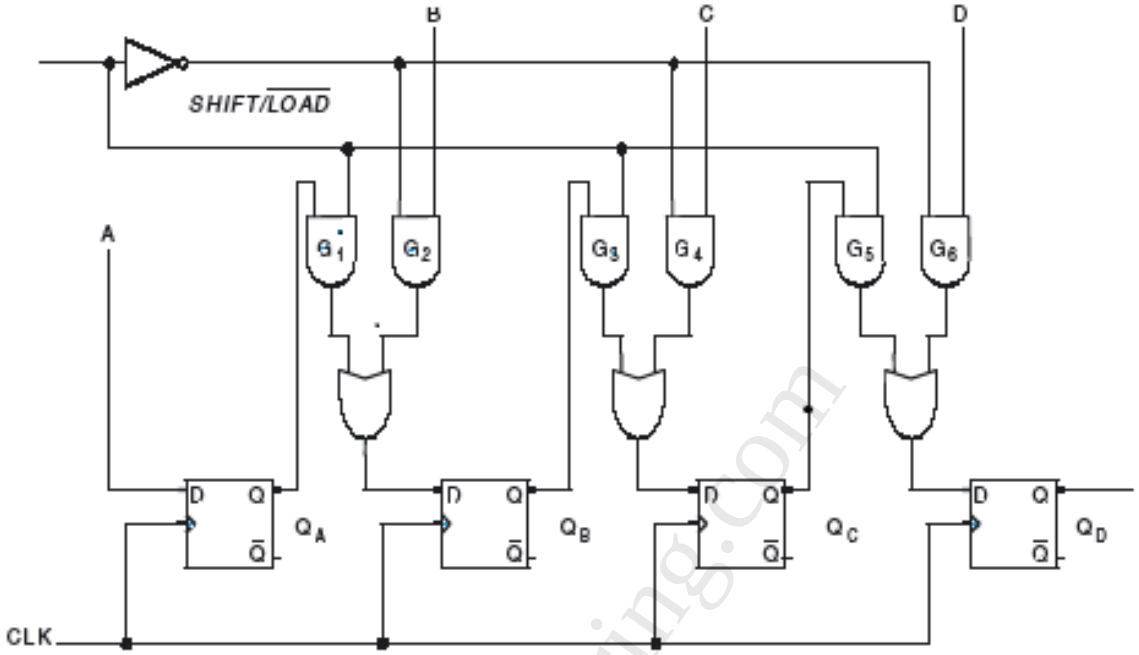
- The parallel data is loaded into the register simultaneously and is shifted out of the register serially one bit at a time under clock control.

Basic Block diagram:



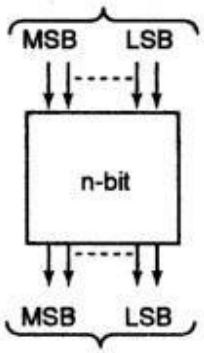
4 bit parallel in serial out shift register




**PIPO (Parallel -in to Parallel -out)**

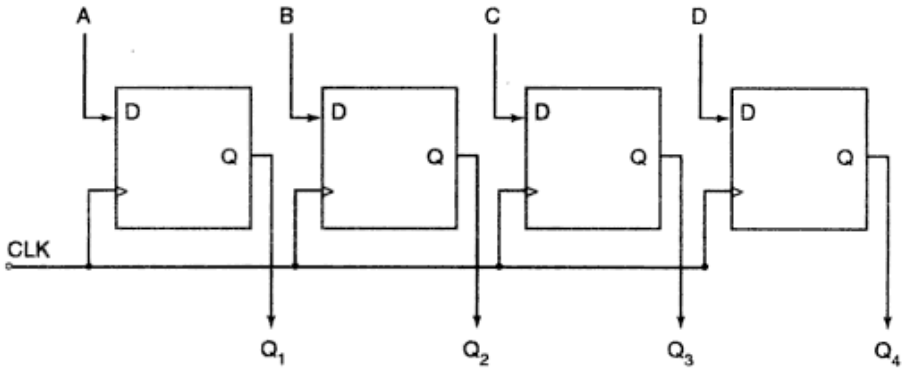
- The parallel data is loaded simultaneously into the register, and transferred together to their respective outputs by the same clock pulse.

Basic Block diagram:



4 bit parallel in parallel out shift register

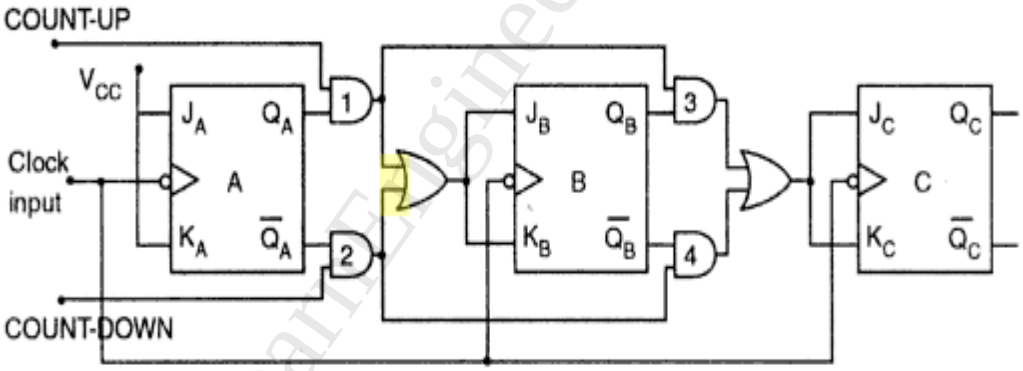



**3. Explain 3 bit Synchronous up/down counter.**

Definition:

Three Flip-flops are used for 3 bit counter. UP/DOWN counter is a combination of the up-counter and the down counter.

Logic Diagram:




Function Table:

| COUNT UP mode | | | | COUNT DOWN mode | | | |
|---|---|---|---|---|---|---|---|
| **States** | **$Q_C$** | **$Q_B$** | **$Q_A$** | **States** | **$Q_C$** | **$Q_B$** | **$Q_A$** |
| 0 | 0 | 0 | 0 | 7 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 6 | 1 | 1 | 0 |
| 2 | 0 | 1 | 0 | 5 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 4 | 1 | 0 | 0 |
| 4 | 1 | 0 | 0 | 3 | 0 | 1 | 1 |
| 5 | 1 | 0 | 1 | 2 | 0 | 1 | 0 |
| 6 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 7 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

Explanation:

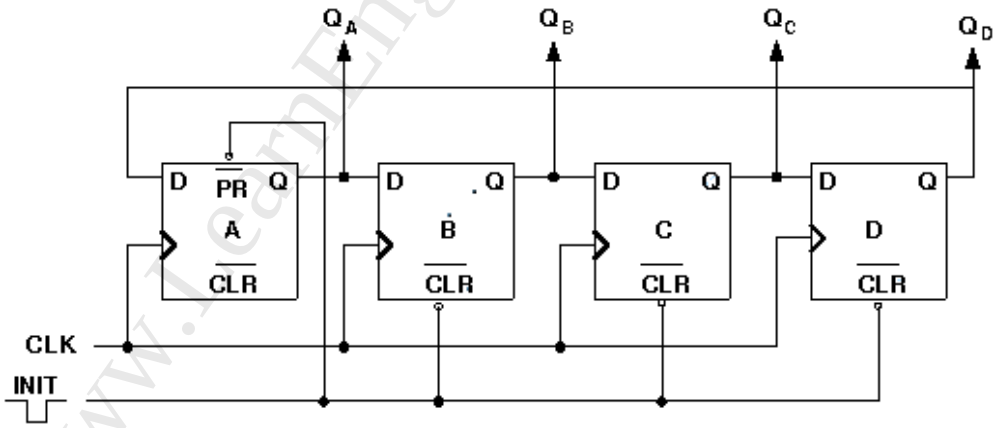
- The Control inputs COUNT –UP and COUNT-DOWN are used to allow either normal output or the inverted output of one flip-flop to the J and K inputs.
- A MOD -8 counter which will count from 000 to 111 when the COUNT-UP =1and COUNT-DOWN=0, is shown in logic diagram.
- A Logical 1 on the COUNT-UP line while COUNT-DOWN = 0 enables AND gates 1 and 3 and disables gates 2 and 4.
- This allows QA, QB outputs through the AND gates to J and K inputs of the following flip-flops, so that the counter counts up as pulses are applied.
- The reverse action takes place when COUNT-UP = 0 and COUNT-DOWN = 1

**4. Explain about 4 bit Ring Counter using D Flip flop**

Definition:

In a Ring Counter the true output of last flip-flop in a shift register is connected back to the serial input of first flip-flop and also only one flip-flop is set at any particular time while all other are cleared.

Logic Diagram:-




Truth Table:

| INIT | CLK | $Q_A$ | $Q_B$ | $Q_C$ | $Q_D$ |
|---|---|---|---|---|---|
| L | X | 1 | 0 | 0 | 0 |
| H | □ | 0 | 1 | 0 | 0 |
| H | □ | 0 | 0 | 1 | 0 |
| H | □ | 0 | 0 | 0 | 1 |
| H | □ | 1 | 0 | 0 | 0 |

Explanation:

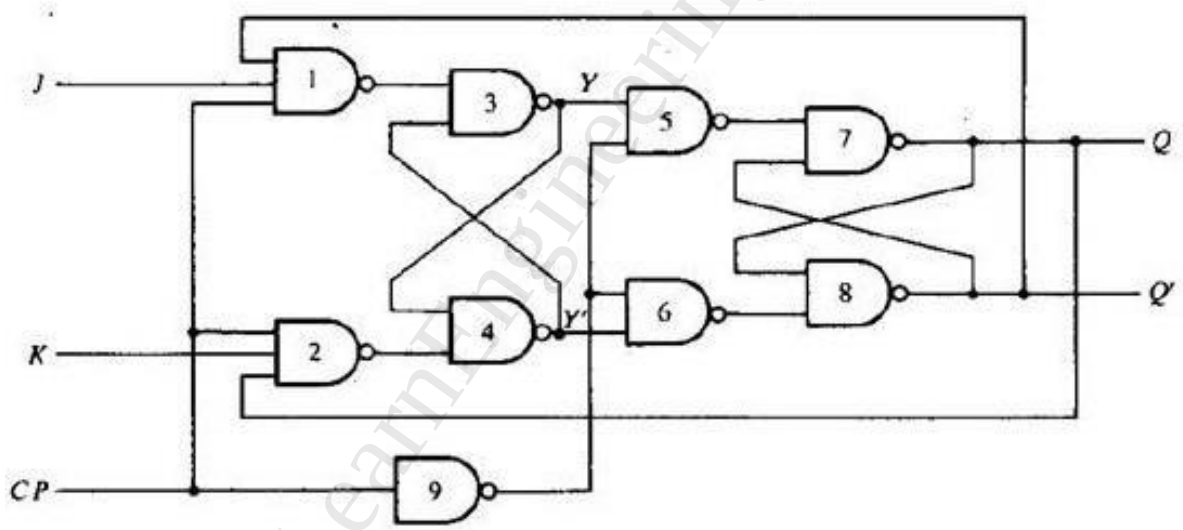
- The Circuit consists of four D flip-flops and their outputs are $Q_A$, $Q_B$, $Q_C$ , $Q_D$. The Preset input of first flip-flop and CLEAR inputs of the other three flip-flops are connected together and brought out as INIT input.
- On applying a LOW pulse at this INIT input, the first flip=flop is SET to 1 and the other three flip-flops are cleared to 0. i.e $Q_AQ_BQ_C Q_D$ = 1000.
- Likewise the process continues and truth table for this operation is described above.

**5. Explain about Master Slave JK Flip flop**

Definition:

It consists of two flip flop, in which Gates 1 to 4 form the master flip flop and Gates 5 to 8 form the SLAVE flip-flop.

Logic Diagram:-




Function Table:

| C | J | K | Q | $\bar{Q}$ | |
|---|---|---|---|---|---|
| ⎍ | 0 | 0 | | | No Change |
| ⎍ | 0 | 1 | 0 | 1 | RESET |
| ⎍ | 1 | 0 | 1 | 0 | SET |
| ⎍ | 1 | 1 | | | Toggle |

Explanation

The information present at J and K inputs transmitted to master flip on the positive edge of clock pulse, and is held there until negative edge of clock pulse occurs, after which it is allowed to pass through slave flip flop.

When positive edge of clock pulse master section is affected and slave section is isolated.

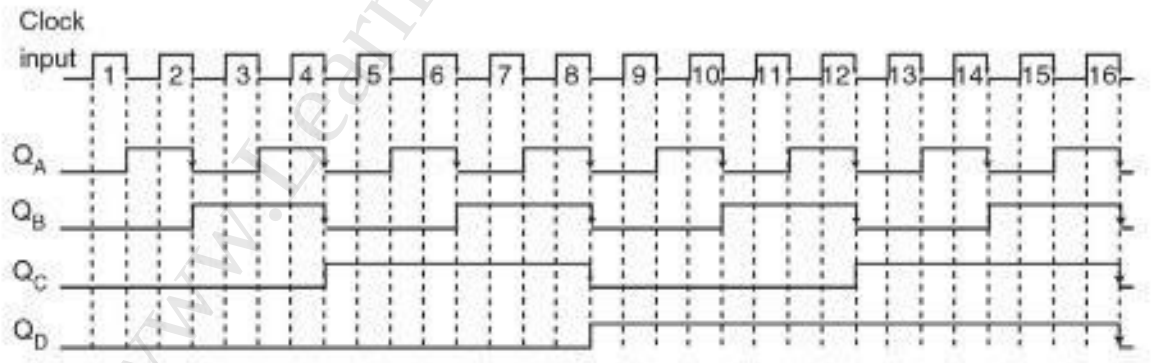
When negative edge of clock pulse master section is isolated and slave section is affected.

**6. Explain about 4 bit Asynchronous Counter.**

In asynchronous counter the external clock signal is connected to the clock input of first stage flip-flop. The clock input of the second, third and fourth stage flip-flops are triggered by its previous stage flip flop output.

**Logic diagram:-**

**Timing waveform:-**




**Truth Table:-**

| CLK | $Q_A$ | $Q_B$ | $Q_C$ | $Q_D$ |
|---|---|---|---|---|
| 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 |

# UNIT IV- MEMORY DEVICES

## PART A

1. **What are the different types of programmable logic devices? (MAY/JUN 2013)**

i. Devices with fixed architecture
   - PLA
   - PAL

ii. Devices with flexible architecture
   - FPGA

2. **How the Bi-polar RAM cell is different from MOSFET RAM cell?**

**Bi-polar RAM cell:-**

It uses TTL (Transistor –Transistor –Logic) multiple emitter technology.

**MOSFET RAM cell:-**

It uses Enhancement mode MOSFET Transistors

3. **What is Read and Write operations?**

The two operations that a random access memory can perform are the write and read operations.

The write signal specifies a transfer-in operation and the read signal specifies a transfer-out operations.

4. **What is PLA? How it differs from ROM?**

PLA is Programmable Logic Array (PLA). The PLA is a PLD that consists of a programmable AND array and a programmable OR array.

A PLA is similar to a ROM in concept; however it does not provide full decoding of the variables and does not generates all the minterms as in the ROM.

5. **What is the maximum range of a memory that can be accessed using 10 address lines?**

The maximum range of memory that can be accessed using 10 address line is $2^{10}$=1024.

6. **Compare static and dynamic RAM cell. (APR/MAY 2015)**

| Static RAM | Dynamic RAM |
|---|---|
| It contains less memory cells per unit area. | It contains more memory cells per unit area as compared to static RAM. |
| Static RAM consists of flip-flops. Each flip-flop stores one bit of binary information. | It stores the data as a charge on the capacitor. It consists of a MOSFET and capacitor on each cell. |

7. **Compare and contrast EEPROM & flash Memory. (NOV/DEC 2014),NOV/DEC 2016**

**EEPROM:**

It is similar to EPROM except that the previously programmed connections can be erased with an electrical signal instead of ultraviolet light.

**Flash Memory:**

It is similar to EEPROM but have additional built-in circuitry to selectively program and erase, without the need for a special programmer.

8. **What is a field programmable gate arrays device? (NOV/DEC 2014)**

- It is a programmable device that can be programmed at the user's location.
- The word 'field' in the name refers to the ability of the gate arrays to be programmed for a specific function by the user instead of by the manufacturer of the devices.
- The word 'array' is used to indicate a series of rows or columns of gates that can be programmed by the end user.

9. **What is Volatile and Non- Volatile memory? (NOV/DEC 2013)**

**Volatile memory:**

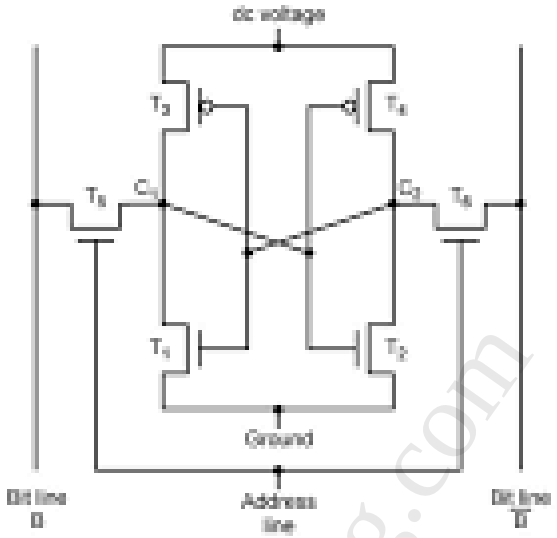
In volatile memory the contents present in the memory will be lost when the power is removed.

**Non-Volatile memory:**

In non volatile memory the contents present in the memory will not be lost even when the power is removed.

**10. Draw the structure of a static RAM cell. (MAY/JUN 2014)**




**11. List the advantages of PLDs. (MAY/JUN 2014)**

a. Low development cost
b. Less space requirement
c. High reliability
d. Easy circuit testing
e. Easy design modification

**12. Give the advantages of RAM. (NOV/DEC 2013)**

a. Fast operating speed
b. Low power dissipation
c. Compatibility
d. Economy
e. Non destructive read out

**13. Distinguish between PLA and PAL. (MAY/JUN 2013) NOV/DEC 2016**

| PLA | PAL |
|---|---|
| It is a type of fixed architecture logic devices with programmable And gates followed by | It is a type of fixed architecture logic devices with programmable And gates |

| | |
|---|---|
| programmable OR gates. | followed by fixed OR gates. |

**14. How the memories are classified? (NOV/DEC 2012)**

i.Registers, Main memory and Secondary memory

ii.Sequential Access Memory and Random Access memory

iii.Static and Dynamic Memory

iv.Volatile and Non-Volatile memory

v.Magnetic and Semiconductor Memory

**15. List the applications of ROM.**

i. It can be used as a direct substitute for any random logic of AND,OR and NOT gates.

ii. It is used to store bootstrap program that loads operating system program available in secondary memory and language interprets in personal and business computers.

iii. It is used for character generation

iv. It is used for code conversion

## PART-B

**1. Implement the combinational circuit given below, using PLA.**

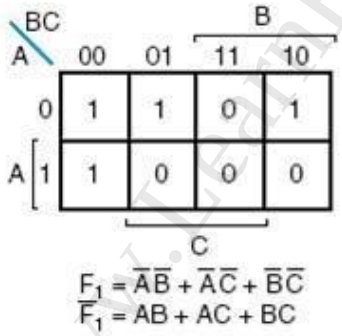
**$F_1(A,B,C) = \Sigma (0,1,2,4)$ ; $F_2(A,B,C) = \Sigma (0,5,6,7)$ .**

**MAY 2015, 2016, NOV 2014**

**Soln:**

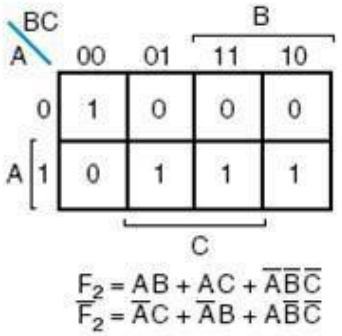
| A | B | C | $F_1$ | $F_2$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 |

Each product term in the expression requires an AND gate. To minimize the cost, it is necessary to simplify the function to a minimum number of product terms.




$F_1 = \overline{A}\overline{B} + \overline{A}\overline{C} + \overline{B}\overline{C}$

$\overline{F}_1 = AB + AC + BC$




$F_2 = AB + AC + \overline{A}\overline{B}\overline{C}$

$\overline{F}_2 = \overline{A}C + \overline{A}B + A\overline{B}\overline{C}$

Designing using a PLA, a careful investigation must be taken in order to reduce the distinct product terms. Both the true and complement forms of each function should be simplified to see which one can be expressed with fewer product terms and which one provides product terms that are common to other functions.

The combination that gives a minimum number of product terms is:

$F'_1 = AB + AC + BC$ (or) $F_1 = (AB + AC + BC)'$

$F_2 = AB + AC + A'B'C'$

This gives only 4 distinct product terms: *AB, AC, BC,* and *A'B'C'*.

So the PLA table will be as follows:

PLA programming table

| | Product term | Inputs A B C | Outputs (C) $F_1$ | Outputs (T) $F_2$ |
|---|---|---|---|---|
| AB | 1 | 1 1 – | 1 | 1 |
| AC | 2 | 1 – 1 | 1 | 1 |
| BC | 3 | – 1 1 | 1 | – |
| $\overline{A}\overline{B}\overline{C}$ | 4 | 0 0 0 | – | 1 |

For each product term, the inputs are marked with 1, 0, or – (dash). If a variable in the product term appears in its normal form (unprimed), the corresponding input variable is marked with a 1.

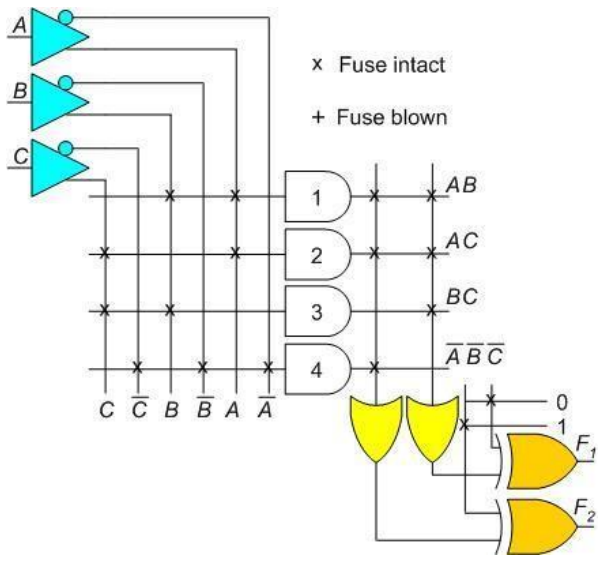
A 1 in the **Inputs** column specifies a path from the corresponding input to the input of the AND gate that forms the product term.

A 0 in the **Inputs** column specifies a path from the corresponding complemented input to the input of the AND gate. A dash specifies no connection.

The appropriate fuses are blown and the ones left intact form the desired paths. It is assumed that the open terminals in the AND gate behave like a 1 input.

In the Outputs column, a **T (true)** specifies that the other input of the corresponding XOR gate can be connected to 0, and a **C (complement)** specifies a connection to 1.

Note that output ***F₁*** is the normal (or true) output even though a **C** (for complement) is marked over it. This is because ***F₁'*** is generated with AND-OR circuit prior to the output XOR. The output XOR complements the function $F_1'$ to produce the true $F_1$ output as its second input is connected to logic 1



**2. Implement the following Boolean functions using the PAL : MAY-15**

**W(A, B, C, D) = ∑m(2, 12, 13)**

**X(A, B, C, D) = ∑m(7, 8, 9, 10, 11, 12, 13, 14, 15)**

**Y(A, B, C, D) = ∑m(0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)**

**Z(A, B, C, D) = ∑m(1, 2, 8, 12, 13).**

**Soln:**

Simplifying the 4 functions to a minimum number of terms results in the following Boolean functions:

W = ABC' + A'B'CD'

X = A + BCD

Y = A'B + CD + B'D'

Z = ABC' + A'B'CD + AC'D' + A'B'C'D

=W +AC'D' + A'B'C'D

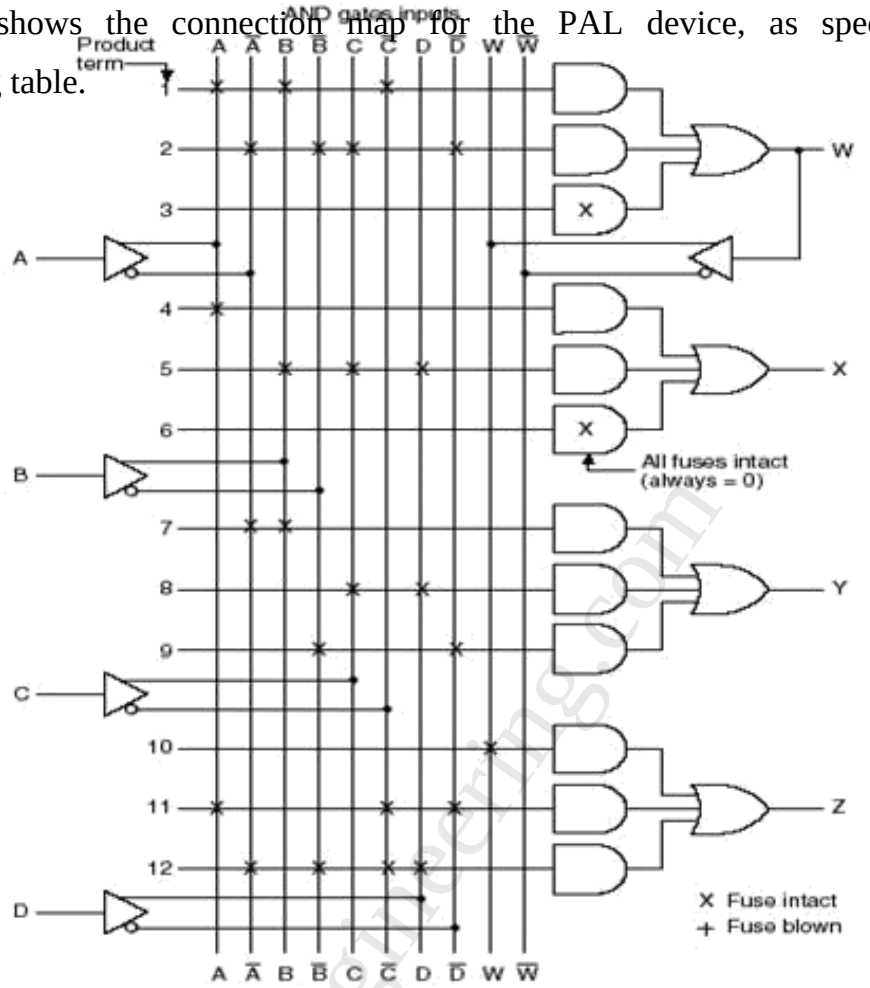
Note that the function for ***Z*** has four product terms. The logical sum of two of these terms is equal to ***W***. Thus, by using ***W***, it is possible to reduce the number of terms for ***Z*** from four to three, so that the function can fit into the given PAL device.

The PAL programming table is similar to the table used for the PLA, except that only the inputs of the AND gates need to be programmed.

| Product term | AND Inputs A | B | C | D | W | Outputs |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | — | — | $W = AB\overline{C}$ |
| 2 | 0 | 0 | 1 | 0 | — | $+\overline{A}\,\overline{B}C\overline{D}$ |
| 3 | — | — | — | — | — | |
| 4 | 1 | — | — | — | — | $X = A$ |
| 5 | — | 1 | 1 | 1 | — | $+BCD$ |
| 6 | — | — | — | — | — | |
| 7 | 0 | 1 | — | — | — | $Y = \overline{A}B$ |
| 8 | — | — | 1 | 1 | — | $+CD$ |
| 9 | — | 0 | — | 0 | — | $+\overline{B}\overline{D}$ |
| 10 | — | — | — | — | 1 | $Z = W$ |
| 11 | 1 | — | 0 | 0 | — | $+A\overline{C}\overline{D}$ |
| 12 | 0 | 0 | 0 | 1 | — | $+\overline{A}\,\overline{B}\,\overline{C}D$ |

The figure shows the connection map for the PAL device, as specified in the programming table.




**PAL circuit design**

Since both W and X have two product terms, third AND gate is not used. If all the inputs to this AND gate left intact, then its output will always be 0, because it receives both the true and complement of each input variable i.e., AA' =0

**3. Explain in detail about Field Programmable Gate Arrays (FPGAs).**

**MAY-16, NOV-15,**

An **FPGA** is a device that contains a matrix of reconfigurable gate array logic circuitry. When a FPGA is configured, the internal circuitry is connected in a way that creates a hardware implementation of the software application. Unlike processors, FPGAs use dedicated hardware for processing logic and do not have an operating system. The FPGA consists of 3 main structures:

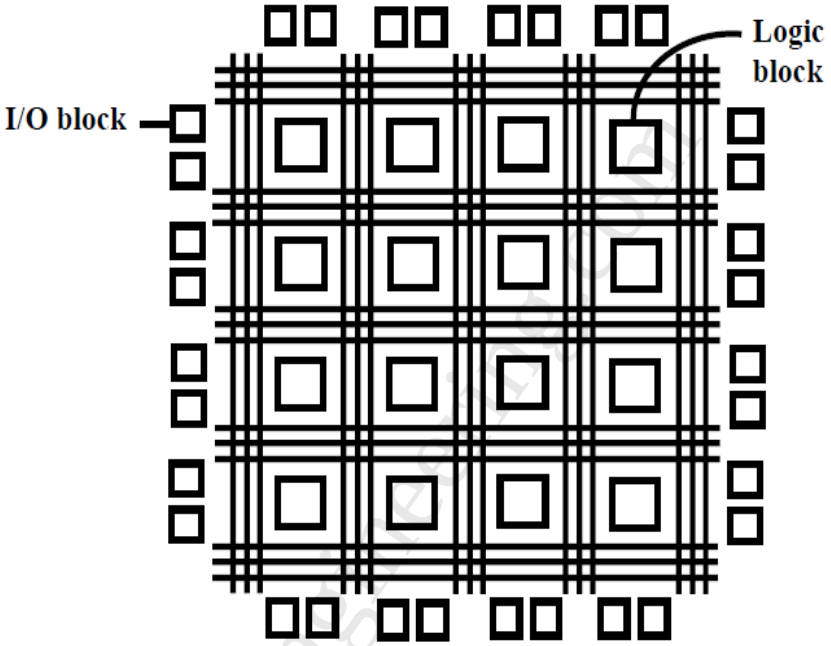
I. Programmable logic structure,

II. Programmable routing structure, and

III. Programmable Input/output (I/O).

## I. Programmable logic structure

The programmable logic structure FPGA consists of a 2-dimensional array of configurable logic blocks (CLBs).




Each CLB can be configured (programmed) to implement any Boolean function of its input variables. Typically CLBs have between 4-6 input variables. Functions of larger number of variables are implemented using more than one CLB.

In addition, each CLB typically contains 1 or 2 FFs to allow implementation of sequential logic.Large designs are partitioned and mapped to a number of CLBs with each CLB configured (programmed) to perform a particular function.These CLBs are then connected together to fully implement the target design. Connecting the CLBs is done using the FPGA programmable routing structure.

## II. Programmable routing structure

To allow for flexible interconnection of CLBs, FPGAs have 3 programmable routing resources:

- Vertical and horizontal routing channels which consist of different length wires that can be connected together if needed. These channels run vertically and horizontally between columns and rows of CLBs as shown in the Figure.

- Connection boxes, which are a set of programmable links that can connect input and output pins of the CLBs to wires of the vertical or the horizontal routing channels.
- Switch boxes, located at the intersection of the vertical and horizontal channels. These are a set of programmable links that can connect wire segments in the horizontal and vertical channels

**III. Programmable I/O**

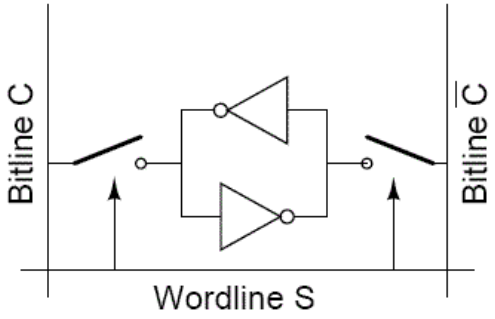
These are mainly buffers that can be configured either as input buffers, output buffers or input/output buffers.

They allow the pins of the FPGA chip to function either as input pins, output pins or input/output pins.

**4. Explain briefly about the Random Access Memory (RAM) & its classifications.**

**SRAM (Static RAM). NOV-15**

The memory circuit is said to be static if the stored data can be retained indefinitely, as long as the power supply is on, without any need for periodic refresh operation. The data storage cell, i.e., the one-bit memory cell in the static RAM arrays, invariably consists of a simple latch circuit with two stable operating points. Depending on the preserved state of the two inverter latch circuit, the data being held in the memory cell will be interpreted either as logic **'0'** or as logic **'1'**. To access the data contained in the memory cell via a bit line, we need atleast one switch, which is controlled by the corresponding word line as shown in Fig.

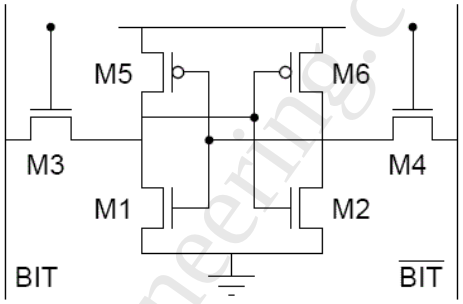



**SRAM Cell**

**CMOS SRAM Cell**

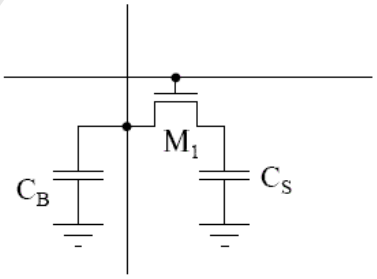
A low power SRAM cell may be designed by using cross-coupled CMOS inverters. The most important advantage of this circuit topology is that the static power dissipation is very small; essentially, it is limited by small leakage current. Other advantages of this design are high noise immunity due to larger noise margins, and the ability to operate at lower power supply voltage. The major disadvantage of this topology is larger cell size. The circuit structure of the full CMOS static RAM cell is shown in Figure 28.12. The memory cell consists of simple CMOS inverters connected back to back, and two access transistors. The access transistors are turned on whenever a word line is activated for read or write operation, connecting the cell to the complementary bit line columns.




**Full CMOS SRAM cell**

**DRAM (Dynamic RAM)**

A typical 1-bit DRAM cell is shown in Fig




**DRAM Cell**

The capacitor $C_S$ stores the charge for the cell. Transistor M1 gives the R/W access to the cell. $C_B$ is the capacitance of the bit line per unit length

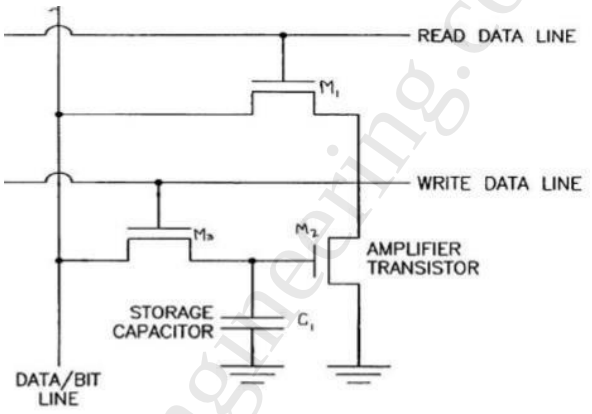
Memory cells are etched onto a silicon wafer in an array of columns (bit lines) and rows (word lines). The intersection of a bit line and word line constitutes the address of the memory cell

DRAM works by sending a charge through the appropriate column (CAS) to activate the transistor at each bit in the column. When writing, the row lines contain the

state the capacitor should take on. When reading, the sense amplifier determines the level of charge in the capacitor. If it is more than 50%, it reads it as **"1";** otherwise it reads it as **"0"**. The counter tracks the refresh sequence based on which rows have been accessed in what order. The length of time necessary to do all this is so short that it is expressed in nanoseconds (billionths of a second). e.g. a memory chip rating of 70ns means that it takes 70 nanoseconds to completely read and recharge each cell

The capacitor in a dynamic RAM memory cell is like a leaky bucket. Dynamic RAM has to be dynamically refreshed all of the time or it forgets what it is holding. This refreshing takes time and slows down the memory.



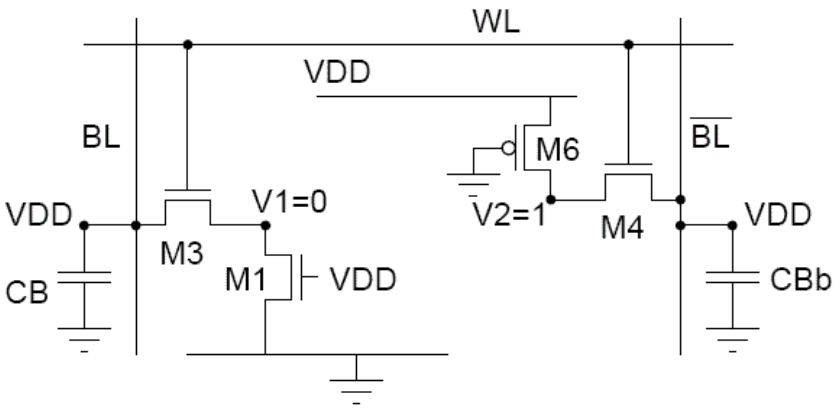

**3 Transistor DRAM**

**5. Discuss in detail about the READ/WRITE Operation of RAM. MAY-16, 15**

**READ Operation of SRAM**

Consider a data read operation, shown in Figure 28.41, assuming that logic **'0'** is stored in the cell. The transistors M2 and M5 are turned off, while the transistors M1 and M6 operate in linear mode. Thus internal node voltages are **V1 = 0** and **V2 = VDD** before the cell access transistors are turned on. The active transistors at the beginning of data read operation are shown in Figure



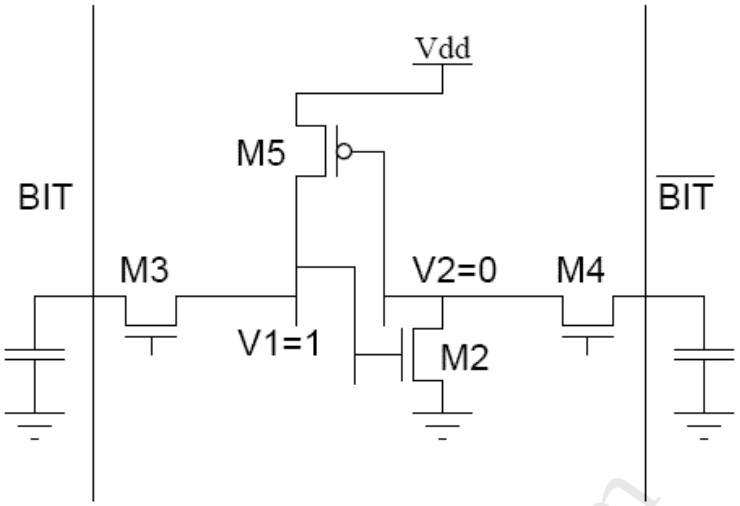

**Read Operation Circuit**

After the pass transistors M3 and M4 are turned on by the row selection circuitry, the voltage **CBb**of will not change any significant variation since no current flows through M4. On the other hand M1 and M3 will conduct a nonzero current and the voltage level of **CB** will begin to drop slightly. The node voltage **V1** will increase from its initial value of **'0'V.** The node voltage **V1** may exceed the threshold voltage of M2 during this process, forcing an unintended change of the stored state. Therefore voltage must not exceed the threshold voltage of M2, so the transistor M2 remains turned off during read phase.

**WRITE Operation of SRAM**

Consider the write **'0'** operation assuming that logic **'1'** is stored in the SRAM cell initially. Figure 28.51 shows the voltage levels in the CMOS SRAM cell at the beginning of the data write operation. The transistors M1 and M6 are turned off, while M2 and M5 are operating in the linear mode. Thus the internal node voltage **V1 = VDD** and **V2 = 0** before the access transistors are turned on. The column voltage **Vb**is forced to **'0'** by the write circuitry. Once M3 and M4 are turned on, we expect the nodal voltage **V2** to remain below the threshold voltage of M1, since M2 and M4 are designed according to



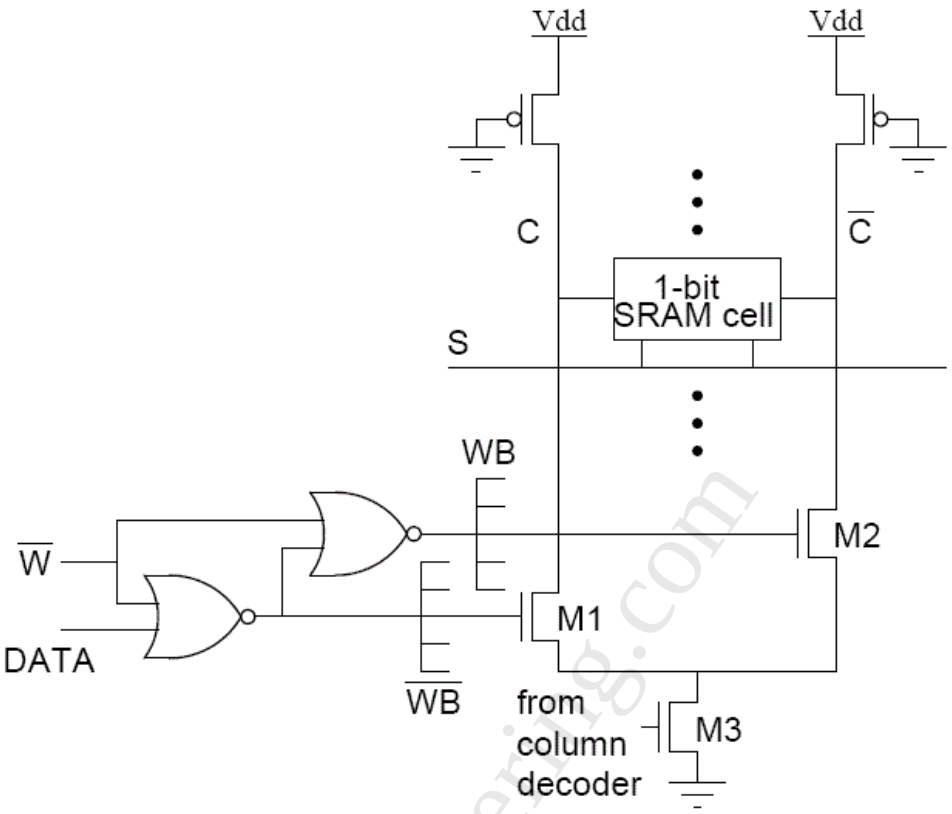

**SRAM start of write 0**

The voltage at node 2 would not be sufficient to turn on M1. To change the stored information, i.e., to force **V1 = 0** and **V2 = VDD**, the node voltage **V1** must be reduced below the threshold voltage of M2, so that M2 **turns off.**

**WRITE Circuit**

The principle of write circuit is to assert voltage of one of the columns to a low level. This can be achieved by connecting either 0 or 1 to ground through transistor M3 and either of M2 or M1. The transistor M3 is driven by the column decoder selecting the specified column. The transistor M1 is on only in the presence of the write enable signal when the data bit to be written is **'0'**. The transistor M2 is on only in the presence of the write signal and when the data bit to be written is **'1'**. The circuit for write operation is shown in Fig

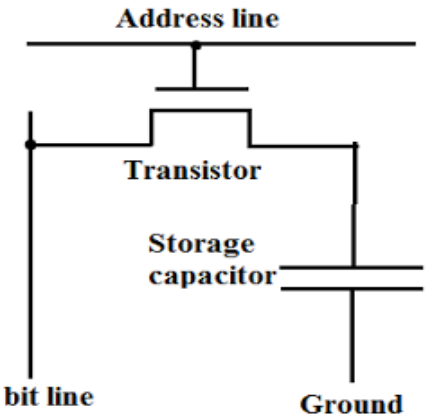
 




**Circuit for WRITE operation**

**READ & WRITE Operation of DRAM**

In Dynamic Random Access Memory CELL , transistor acts as a switch to Close (allowing current to flow) when voltage applied in address line or Open (no current flow) when no voltage applied in address line.Address Line also known as word line .Which use to signal the transistor to **close** or **open**




During the **Write operation**,a voltage is applied on the bit line and a signal applied to the address line to close the transistor.Then the voltage

applied on the bit line will transfer to capacitor and store in the capacitor. However the capacitor has tendency to discharge and has to refresh to maintain the bit.

While in **Reading Operation**, the instruction finds the bit store using the address line to read the data or bit. When the address line is selected, the transistor turns on and the charge stored on the capacitor is fled out onto a bit line and to sense amplifier. Sense amplifiers compare the capacitor.

## UNIT V SYNCHRONOUS & ASYNCHRONOUS SEQUENTIAL CICUITS

## PART A

1. **What is Hazard? Give its types. (NOV/DEC 2013), (MAY/JUN 2013) NOV/DEC 2016**

These are unwanted switching transients that may appear at the output of a circuit because different paths exhibit different propagation delays.

The two types of hazards are,

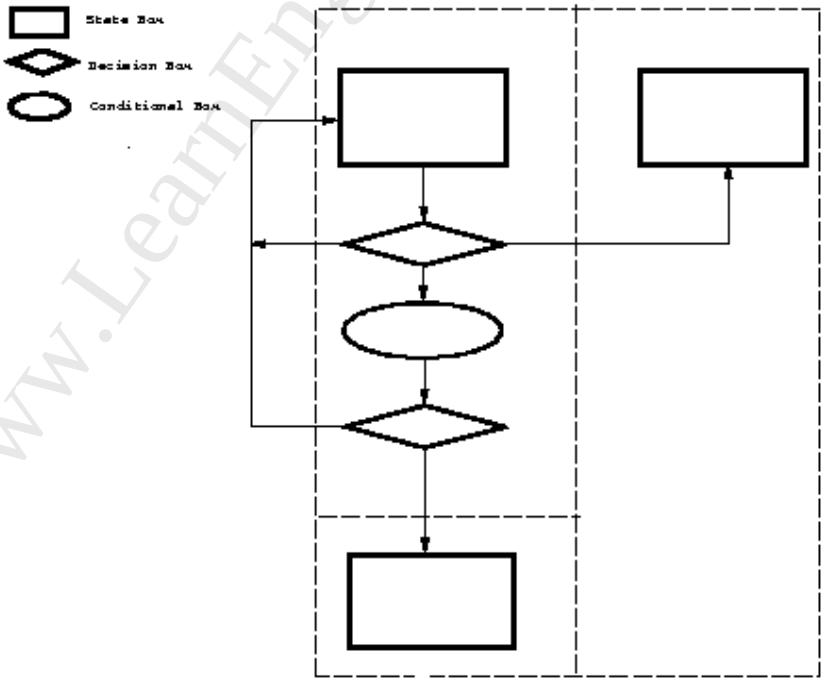
a. Static hazard

b. Dynamic hazard

2. **Design a 3 input AND gate using verilog. (NOV/DEC 2012)**

```
module gate(a,b,c,y);      input a,b,c;
    output y;
    and G(y,a,b,c)
endmodule
```

3. **Draw the general model of ASM.**




4. **Define static 1 hazard.**

Due to change in single variable if the output momentarily goes to state 0, when the output is expected to remain in state 1.Then such a type of hazard is said to be static 1 hazard.

5. **Differentiate flow chart and ASM chart. (MAY/JUN 2013)**

**Flow Chart:**

It gives a sequence of events that must occur one after the other.

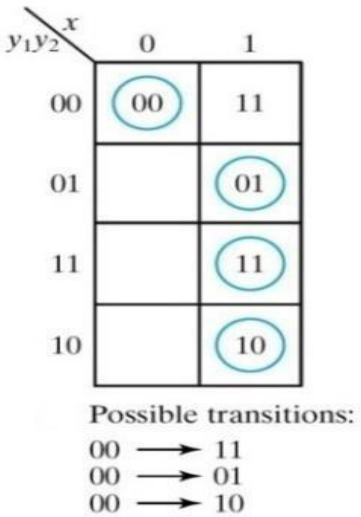
**ASM Chart:**

It represents each event occurring in a particular state time.

6. **List the problems that arise in asynchronous circuits.**
   a. Cycles
   b. Races
   c. Hazards
7. **Define ASM chart. List its three basic elements. (NOV/DEC 2014)**
   - ASM chart resemble the flow chart conventionally used in software design.
   - It is used to represent diagram of digital integrated circuit.
   - It is the method of describing the sequential operation of digital system.
   - The chart is composed of three basic elements.
     1. State box
     2. Decision box
     3. Conditional box.
8. **What is critical race condition in asynchronous sequential circuits? Give an example.**

If the final stable state depends on the order in which the state variable changes, the race condition is harmful and it is called critical race.

Example:



**9. What is state diagram? Give an example. (MAY/JUN 2014)**

The state diagram or state graph is a pictorial representation of the relationships between the present state, the input , the next state, and the output of the finite state sequential machine.

It is used to give an abstract description of behavior of a system.

**10. Write the VHDL code for a half adder. (MAY/JUN 2014)**

```
ENTITY Half adder
Port   a, b : IN
       s , c : OUT
Architecture half adder
begin
   s < = a XOR b
   c < = a AND b
END Half adder.
```

**11. What is a Synchronous sequential circuit? (NOV/DEC 2013)**

If the transitions of the sequential circuit from one state to the next state are controlled by the clock, then the circuit is called synchronous sequential circuit.

**12. Define primitive flow table.**

It is defined as a flow table which has exactly one stable state for each row in the table. The design process begins with the construction of primitive flow table.

**13. Write the Difference between Mealy and Moore model Sequential circuit.**

| MEALY MODEL | MOORE MODEL |
|---|---|
| Its output is a function of present state and present output. | Its output depends on present state only. |
| Input changes may affect output | Input changes does not affect output |

**14. What is a cycle?**

A cycle occurs when an asynchronous circuit makes a transition through a series of unstable states. If a cycle does not contain a stable state, the circuit will go

from one unstable to stable to another, until the inputs are changed.

**15. What are the steps for the design of asynchronous sequential circuit?**

a. Construction of a primitive flow table from the problem statement.
b. Primitive flow table is reduced by eliminating redundant states using the state reduction
c. State assignment is made
d. The primitive flow table is realized using appropriate logic elements.

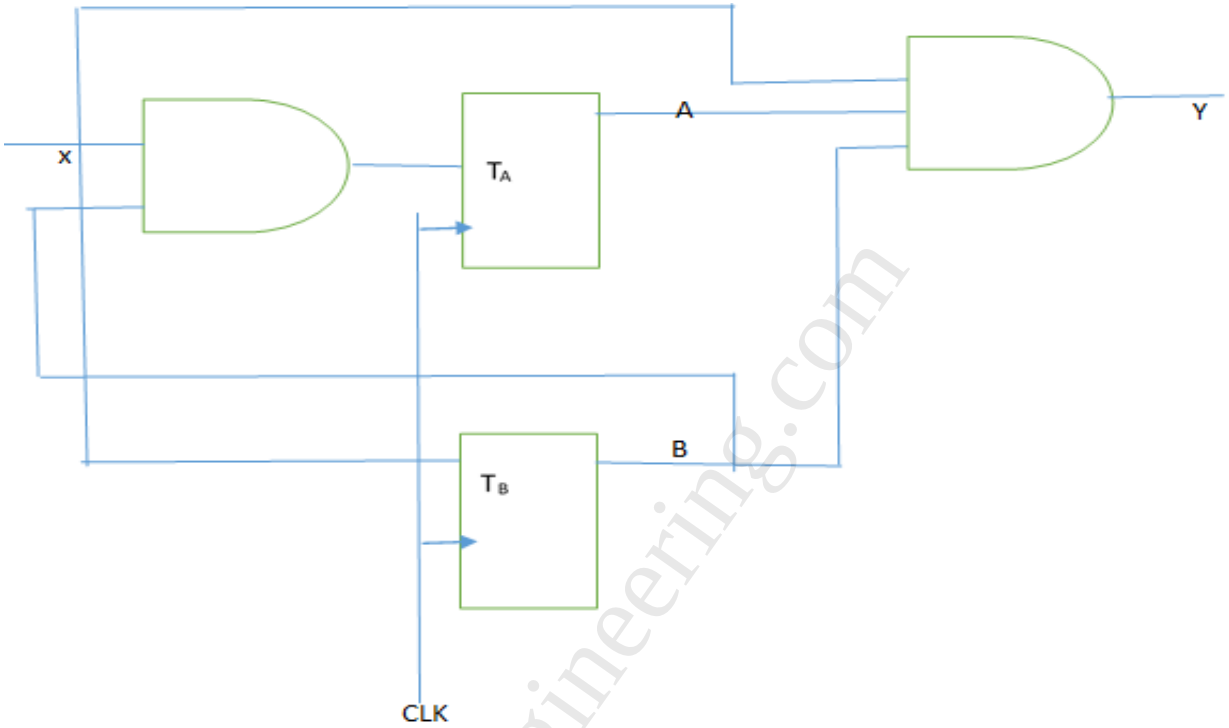
**16.Define Critical Race and Non critical Race (NOV/DEC 2016)**

| **Critical Race** | **Non critical Race** |
|---|---|
| If an input change induces the circuit to go to the same wrong state.Then it is termed as critical races.Critical races must be avoided in an asynchronous circuit | If an input change induces the circuit to go to a wrong state which is unstable will finally reaches a correct stable state.This condition is known as non critical race |

## PART-B

**1.Construct the transition table, state table and state diagram for the mealy model sequential circuit**




Flip flop input equation

$T_A=Bx$

$T_B=x$

Flip flop output equation

$Y=ABx$

**Excitation table of T Flipflop**

| $T_A$ | $A^+$ |
|---|---|
| 0 | A |
| 1 | A’ |

**Transition Table**:

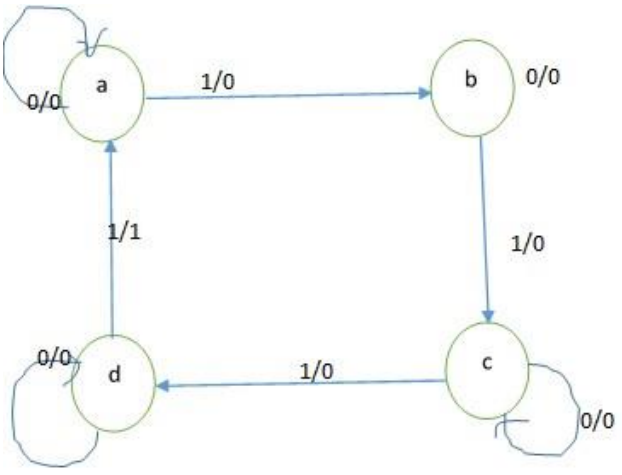
| Present State | | Input | Next State | | Flip flop Input | | Output |
|---|---|---|---|---|---|---|---|
| **A** | **B** | **x** | **$A^+$** | **$B^+$** | **$T_A$** | **$T_B$** | **Y** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

**State Table:**

Present state and next state is 2 bits. Therefore assume a=00,b=01,c=10,d=11

| Present state | Next State | | Output F | |
|---|---|---|---|---|
| | **x=0** | **x=1** | **X=0** | **X=1** |
| a | a | B | 0 | 0 |
| b | b | C | 0 | 0 |
| c | c | D | 0 | 0 |
| d | d | A | 0 | 1 |

State Diagram:



**2. Design an asynchronous sequential circuit with two inputs $x_1$ and $x_2$ and one output Z. MAY-16, NOV-15, 14**

Initially, both inputs are equal to zero. When $x_1$ or $x_2$ becomes 1,the output Z becomes 1.When the second input also becomes 1, the output changes to 0.The output stays at 0 until the circuit goes back to the initial state.

**Solution:**

**Step1**

From the word description of the problem, the operation of the circuit can be easily understood. Now, from this, a primitive state table can be drawn.

**Step2**

**Primitive state table**

The primitive state table for the given requirement is shown

Partial primitive state table

| Next state(NS),OUTPUT(Z) | | | |
|---|---|---|---|
| $X_1X_2$=00 | 1 | 11 | 10 |
| (1) ,0 | 2 | | 3 |
| | (2) ,1 | 4 | |
| | | 4 | (3) ,1 |
| 5 | | (4) 0 | 6 |
| 1 | (5) ,0 | | |
| 1 | | | (6) ,0 |

Primitive state table

| Row | Next state(NS),OUTPUT(Z) | | | |
|---|---|---|---|---|
| | $x_1x_2$=0 0 | 0 1 | 1 1 | 1 0 |
| 1 | (1) ,0 | 2 | - | 3 |
| 2 | 1 | (2) ,1 | 4 | - |
| 3 | 1 | - | 4 | (3) ,1 |
| 4 | - | 5 | (4) 0 | 6 |
| 5 | 1 | (5) ,0 | 4 | - |
| 6 | 1 | - | 4 | (6) ,0 |

**Step 3**

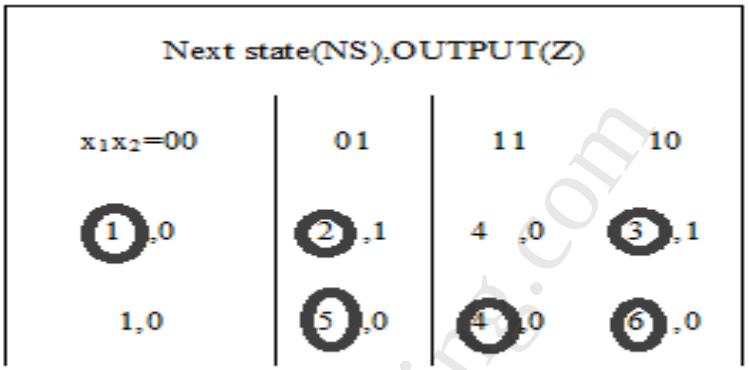
**Minimization of primitive state table**

The minimization of primitive state table has two functions

i. eliminating redundant stable states and

ii. merging those stable states which are distinguishable by the input states.

Minimized state table




**Step 4**

**State Assignment**

PS.NS and output table

| Y | Y,Z | | | |
|---|---|---|---|---|
| | $x_1 x_2$=00 | 01 | 11 | 10 |
| 0 | 0,0 | 0,1 | 1,0 | 0,1 |
| 1 | 0,0 | 1,0 | 1,0 | 1,0 |

**Step5**

If delay flip flop is used, one can obtain the excitation and output table as shown

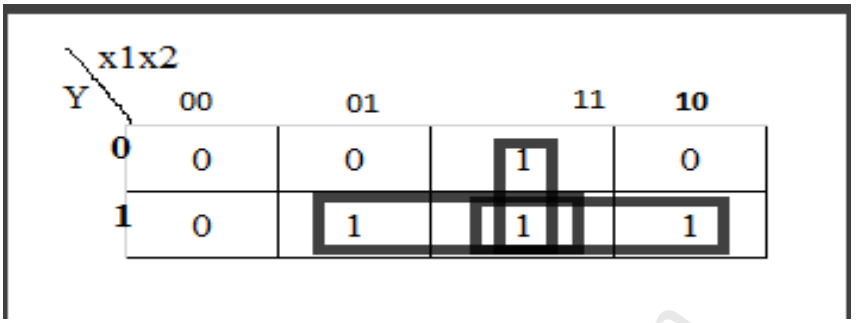
Excitation and output table

| Y | D,Z | | | |
|---|---|---|---|---|
| | $x_1 x_2$=00 | 01 | 11 | 10 |
| 0 | 0,0 | 0,1 | 1,0 | 0,1 |
| 1 | 0,0 | 1,0 | 1,0 | 1,0 |

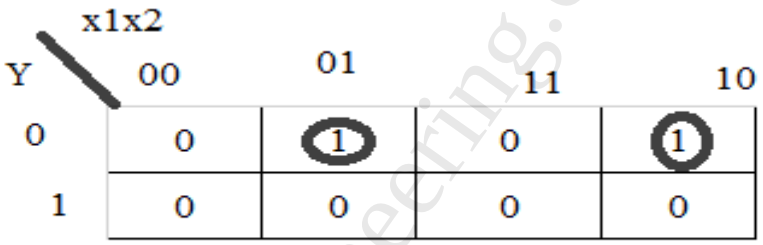
 

**Step6**

The K maps for simplifying the excitation function Dd) and output (Z) are as shown




Excitation map for D




Output map for Z

$D=x_1x_2+x_2Y+Yx_1$

$Z=Y'(x_{1XOR\ WITH}x_2)$

**Step 7**

Using the simplified expressions for D and Z, the circuit diagram for the given asynchronous sequential circuit can be drawn.

**3. Write the Verilog code for T flipflop and JK flipflop from D flipflop**

```
//T flipflop from D flipflop
module TFF(T,CLK,RST,Qn);
input T,CLK,RST;
output Qn;
wire x;
assign x= T^Qn;
DFF ct(x,CLK,Qn,RST);
```

```
Endmodule
//D flipflop
module DFF(D,CLK,Q,RST);
input D,CLK,RST;
output Q;
reg Q;
always @ (posdege CLK or negdege RST)
if(~RST)
Q=1'b0;
else Q =D;
endmodule
//JK flipflop from D flipflop
module JKD(J,K,CLK,RST,Qn);
input J,K,CLK,RST;
output Qn;
wire x;
assign x=(J&~Qn)|(~K&Qn);
DFF ct(x,CLK,Qn,RST);
endmodule
//D flipflop
module DFF(D,CLK,Q,RST);
input D,CLK,RST;
output Q;
reg Q;
always @ (posdege CLK or negdege RST)
if(~RST)
Q=1'b0;
else Q =D;
endmodule
```

**5. Explain the problems in asynchronous sequential circuit** **NOV-14**

The three problems in asynchronous sequential circuits are,

1. Cycles
2. Races
3. Hazards

**Cycles:**

If an input change includes a feedback transition through more than one unstable state then such a situation is called cycle.

The circuit goes through an unique sequence of unstable states because of an input change such a situation is called cycle.

**Races:**

When two or more feedback variable change value in response to a change in as input variable then a race condition is said to exist in an asynchronous sequential circuit.

The two types of Races are

1. Critical Race
2. Non Critical Race

**Critical Race**

If an input change induces the circuit to go to the same wrong state then it is termed as critical Race.

It must be avoided in an asynchronous circuit.

**Non Critical Race**

If an input change induces the circuit to a wrong state which is unstable will finally reaches a correct stable state. This condition is known as non critical race.

**Hazards:**

Hazards are unwanted switching transient that may appear at the output of circuit because different path exhibit different propagation delays.

Hazards are classified into

1. Static Hazard
2. Dynamic Hazard
3. Essential Hazard

**Static Hazard:**

It is a Hazard that occurs in combinational circuit,which result in a single momentary incorrect output due to change in single input variable, when the output is expected to remain in same state.

**Static 0 Hazard**

Due to change in single input variable if the output momentarily goes to state 1 when the output is expected to remain in state 0.Then such type of hazard is said to be static 0.

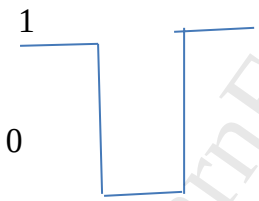
**Static 1 Hazard**

Due to change in single input variable if the output momentarily goes to state 0, when the output is expected to remain in state1, then such type of Hazard is said to be static 1 Hazard.

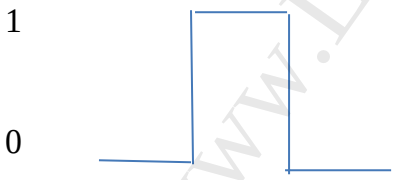
**Dynamic Hazard**

It causes the output to change three or more times when it should change from 1 to 0 or from 0 to 1.

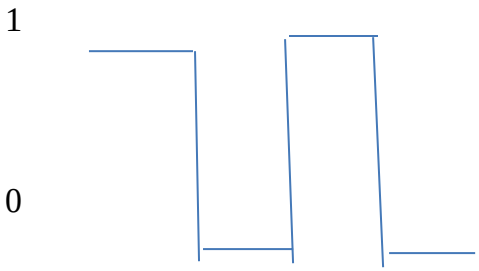
Static 1 Hazard:




Static 0 Hazard:




Dyamic Hazard:



**5. With ASM chart design a binary multiplier MAY-15**

**ASM chart design**

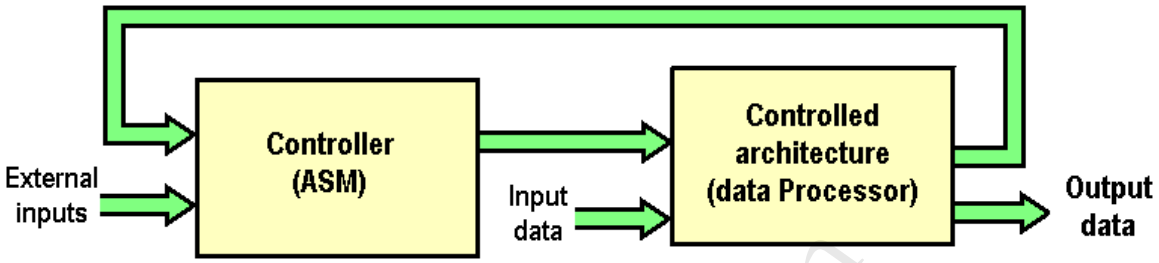
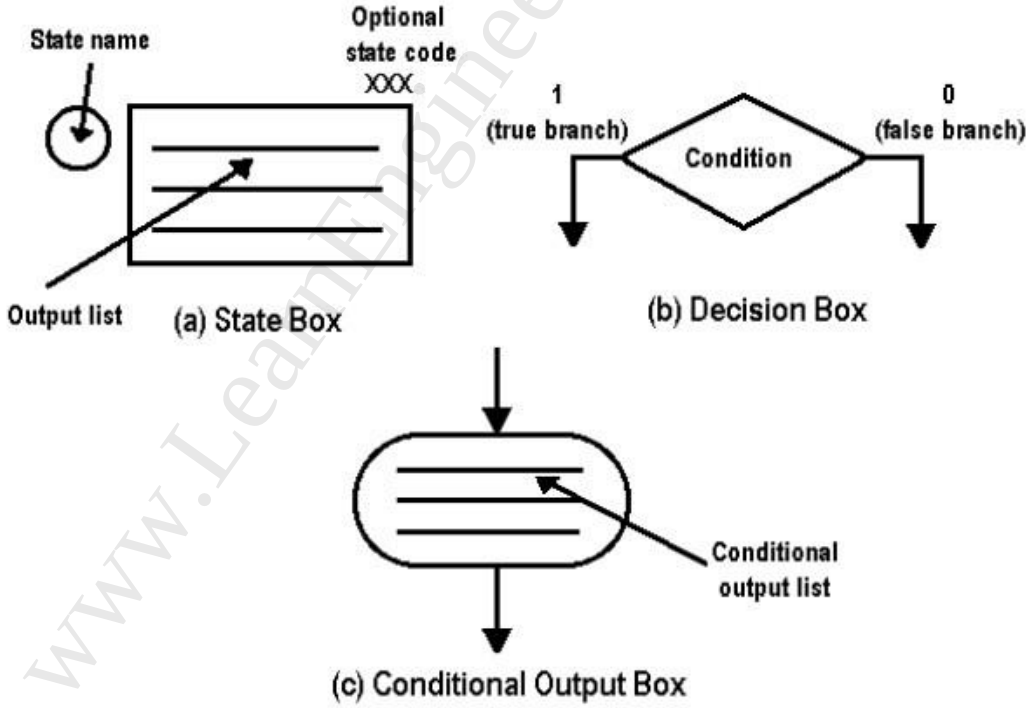



**Fig.Partition of digital system**

The algorithmic state machine SM chart can be divided into three blocks namely the state box, the decision box, and the conditional output box. They are




The state box contains an output list, state name, and optional state code. The decision box is a usual diamond shaped symbol with true and false evaluate to decide the branches. The conditional output box contains conditional output list. The conditional outputs depend on both the state of system and the inputs. Let's consider an ASM chart shown in Fig and use it to explain the flow of the sequence.

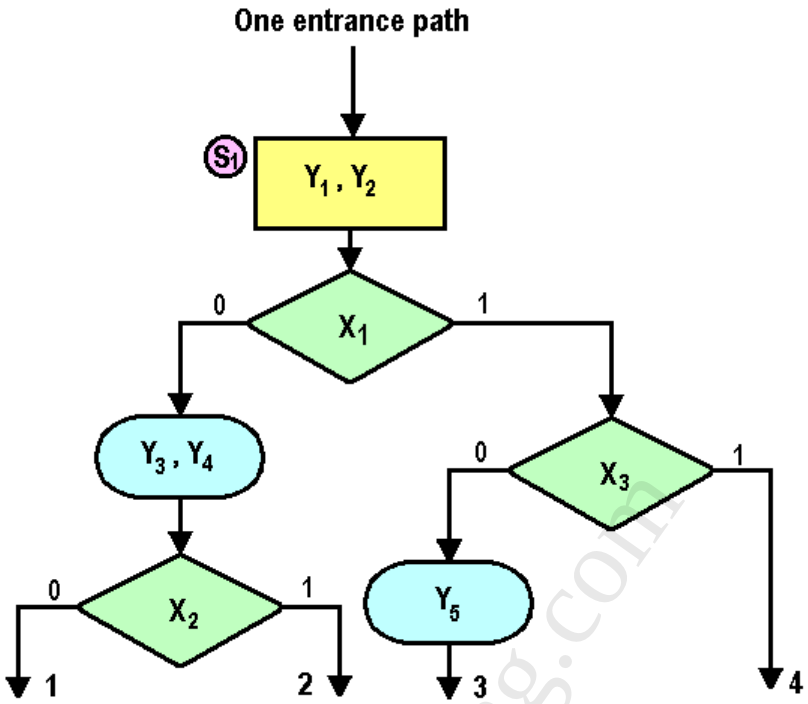



**Fig. An ASM chart showing the state box, conditional box, and decision box**

When state S1 entered, output Y1 and Y2 become "1". If inputs X1 and X2 are both equal to 0, Y3 and Y4 are also "1", and at the end of state time the machine goes to the next state via exit path 1. If inputs X1 = "0" and X2 = "1", Y3 and Y4 are "1", the exit to next state is via path 2.

On the other hand, if X1 = "1" and X3 = "0", the output Y5 is "1" and exit to the next state will occur via exit path 3. If X1 = "1" and X3 = "1", the exit to the next state will occur via path 4.

**Question Paper Code: 57281**

**B.E/B.Tech DEGREE EXAMINATIONS MAY/JUNE 2016**

**Third Semester**

**EC6302-DIGITAL ELECTRONICS**

**(Common to Mechatronics Engineering and Robotics and Automation Engineering)**

**Regulation 2013**

**Time: Three hours** **Maximum Marks: 100 marks**

**ANNSWER ALL QUESTIONS**

**PART-A (10 x 2 = 20 marks)**

1. Prove the Boolean theorems (a) x + x = x ;
   (b) x + xy = x [Pg- 1]
2. Define Noise Margin. [Pg- 1]
3. Write the design procedure of combinational circuits. [Pg- 24]
4. Draw the combinational circuit that converts 2 coded inputs into 4 coded outputs.
5. Differentiate synchronous and asynchronous sequential circuit. [Pg- 39]
6. Give the truth table of transparent latch.
7. Give the classification of programmable logic devices. [Pg- 52]
8. How the Bi-polar RAM cell is different from MOSFET RAM cell? [Pg- 52]
9. What are Hazards? [Pg- 57]
10. Define critical race and give the methods for critical-race free state assignment. [Pg- 68]

**PART-B (5 x 16 =80 marks)**

11.(a) Simplify the following Boolean function F, using Quine Mccluskey method and verify the result using K-map $F(A,B,C,D) = \Sigma (0,2,3,5,7,9,11,13,14)$ (16) [MODEL Pg- 4]

Or

(b) (i) Draw and explain Tri-state TTL inverter circuit diagram with its operation. (10) [Pg- 13]

(ii) Implement the following function using NAND and inverter gates. (6)

F = AB + A'B' + B'C

12. (a) (i) Design a 4-bit magnitude comparator with 33 outputs: A > B, A=B, A<B. [Pg- 30] (8)

(ii) Design a 4 bit binary to gray code converter. (8)

Or

(b) (i) Implement the following Boolean function using 8 x 1 Multiplexers.

F (A, B, C, D) = Σ (1,3,4,11,12,13,14,15) [Pg- 38] (8)

(ii) Explain the concept of carry look ahead adder with neat logic diagram. [Pg- 28] (8)

13. (a) Design a 3-bit synchronous counter using D-flip flop. (16)

Or

(b) (i) Draw and explain the 4-bit SISO,PISO and PIPO shift register with its waveforms. [Pg- 44] (12)

(ii) Realize D flip-flop using SR flip-flop. (4)

14.(a) (i) Implement the following function with PLA. [MODEL Pg- 56]

$F_1(x,y,z) = \Sigma m(1,2,4,6)$

$F_2(x,y,z) = \Sigma m(0,1,6,7)$

$F_3(x,y,z) = \Sigma m(2,6)$ (12)

(ii) Write Short notes on FPGA. [Pg- 59] (4)

Or

(b) (i) Explain memory READ and WRITE operation with neat timing diagram [Pg- 63] (8)

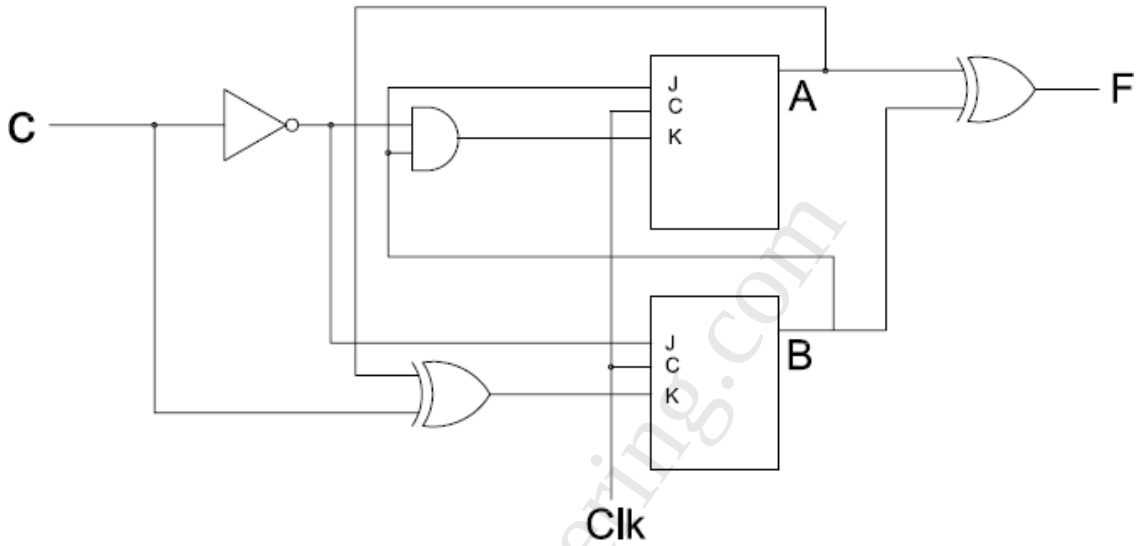
(ii) Explain the organization of ROM with relevant diagrams. (8)

15.(a) (i) Design an asynchronous sequential circuit with two inputs $X_1$ and $X_2$ and with one output Z. When $X_1$ is 0,the output Z is 0.The first change in $X_2$ that occurs while $X_1$ is 1 will cause output Z to be 1.The output Z will remain 1 until $X_1$ returns to 0. (16)

Or

(b)Construct the transition table, state table and state diagram for the Moore sequential circuit given below. (16)



**Question Paper Code: 27189**

**B.E/B.Tech DEGREE EXAMINATIONS NOV/DEC 2015**

**Third Semester**

**EC6302-DIGITAL ELECTRONICS**

**(Common to Mechatronics Engineering and Robotics and Automation Engineering)**

**REGULATION 2013**

**Maximum Marks : 100 marks**

**ANNSWER ALL QUESTIONS**

**PART-A (10 x 2 = 20 marks)**

1. State De –Morgan's Theorem. [Pg- 1]
2. Express the function $Y = A + \bar{B}C$ in canonical POS.
3. Define Half adder & Full adder [Pg- 24]
4. What is priority encoder? [Pg- 25]
5. What are the classifications of Sequential circuits? [Pg- 39]
6. What is edge triggered flip-flop? [Pg- 39]
7. What is Read and Write operations? [Pg- 52]
8. What is PLA? How it differs from ROM? [Pg- 52]
9. Draw the general model of ASM. [Pg- 67]
10. What is Hazard? Define static 1 hazard. [Pg- 67]

**PART-B (5 x 16 =80 marks)**

11.(a) (i) Minimize the following logic function using K-maps and realize using NAND and NOR gates. $F(A,B,C,D) = \Sigma (1,3,5,8,9,11,15)+ d(2,13)$ [Model Pg-7] (10)

(ii) Show that if all the gate in a two-level OR-AND gate network are replaced by NOR gate, the output function does not change. (6)

Or

(b) (i) Realize NOT,OR,AND gates using universal gates. [Pg- 10] (8)

(ii) Discuss about the basic operation of TTL NAND gates. [Pg- 13] (8)

12. (a) Explain with neat diagram the function of Binary Multiplier [Pg- 34]

i. Using Shift method (8)

ii. Parallel multiplier (8)

Or

(b)Design BCD to excess 3 code converter using minimum number of NAND gate. (16)

[Pg- 36]

13. (a)(i)Explain the operation of J.K Flip flop with neat diagram (10)

(ii)Explain the operation of master slave Flip flop and show how the race around condition is eliminated. (6)

Or

(b) Explain the operation of synchronous MOD 6 counter (16)

14. (a) Write the difference between static and dynamic RAM. Draw the circuits of one cell each and explain its working. [Pg-61] (16)

Or

(b)Write notes on [Pg- 59]

(i)PAL (8)

(ii)FPGA (8)

15. (a) Design a asynchronous sequential circuit with two inputs T and C. The outputs attains a value of 1 when T=1 and C moves from 1 to 0.Otherwise the output is 0.

[MODEL Pg- 73] (16)

Or

(b)Explain the difference methods of RACE Free state assignment. (16)

**Question Paper Code: 71444**

**B.E./B.Tech DEGREE EXAMINATION, APRIL /MAY 2015**

**Third Semester**

**Electronics and Communication Engineering**

**EC 2203/EC 34/080290010/10144 EC 304- DIGITAL ELECTRONICS**

**(REGULATION 2008/2010)**

**(Common to PTEC 2203- Digital Electronics for B.E (Part Time) Third Semester- Electronics and communication Engineering Regulation 2009)**

**Time: Three hours** **Maximum:100 Marks**

**Answer ALL questions**

**PART A – (10 *2 =20 marks)**

1. Define 'min term' and 'max term'. [Pg- 3]
2. Write a note on tristate gates. [Pg- 3]
3. Give the logic expressions for sum and carry in full adder circuit. [Pg- 25]
4. Give examples for combinational circuit (Any four). [Pg- 25]
5. Realize T FF and JK FF. [Pg- 40]
6. Draw the circuit diagram of a 3 bit Ring counter. [Pg- 42]
7. Compare static and dynamic RAM cell (any two) [Pg- 53]
8. Y=AB'+A'. Implement using ROM.
9. Differentiate flow chart and ASM chart. [Pg- 67]
10. List the problems that arise in asynchronous circuits. [Pg- 68]

**PART B- (5*16=80 Marks)**

11.(a) (i) Simplify T(x,y,z)=(x+y)[x'(y'+z')]'+x'y'+x'z' (6)

(ii) Simplify the Boolean function and draw the logic diagram

f(w,x,y,z)=∑(0,1,2,4,5,6,8,9,12,13,14). [MODEL Pg- 4] (10)

Or

(b) (i) Realize AND, OR and NOT gate using NAND gate. [Pg- 10] (6)

(ii) Using tabulation method simplify

F=(A,B,C,D,E)= ∑(0,1,4,5,16,17,21,25,29). [Pg- 7] (10)

12. (a) Design a combinational circuit that converts 4 bit Gray code to a 4 bit binary number. Implement the circuit. (16)

Or

(b) Detail the following:

(i) BCD adder [Pg- 32] (8)

(ii) Magnitude Comparator.[Pg- 30] (8)

13.(a) (i) Describe a JK FF with its characteristic table and characteristic equation (6)

(ii) With a neat sketch describe a 3 bit synchronous up/down counter. Draw the timing waveform [Pg- 47] (10)

Or

(b)Design a sequential circuit with two D FF s A and B and one input x. When x=0, the state of the circuit remains the same. When x=1,the circuit goes through the state transitions from 00-01-11-10-00-01… (16)

14. (a) (i) List the steps involved in memory read and memory write operations. [Pg- 63] (10)

(ii) Give an account for classification of memories. (6)

Or

(b) Explain the structure of PAL and PLA .How a combinational logic function is implemented in PAL and PLA? Explain with an example for each. [Pg-56-59] (16)

15. (a) (i) Write the VERILOG code for full adder and JK FF. (8)

(ii) Explain the different types of hazards. Design a hazard free circuit for $y=x_1x_2+x_2'x_3$. (8)

Or

(b) With ASM chart design a binary multiplier. [MODEL Pg- 79] (16)

**Question Paper Code: 97059**

**B.E./B.Tech DEGREE EXAMINATION, NOVEMBER/DECEMBER 2014**

**Third Semester**

**Electronics and Communication Engineering**

**EC 6302- DIGITAL ELECTRONICS**

**(Common to Mechatronics Engineering and Robotics and Automation Engineering)**

**(Regulation 2013)**

Time: Three hours Maximum:100 Marks

Answer ALL questions

**PART A – (10 *2 =20 marks)**

1. Simplify the following Boolean expression into one literal
   W'X(Z'+YZ)+X(W+Y'Z) [Pg- 2]
2. Draw the CMOS inverter circuit. [Pg- 2]
3. Construct 4 bit parallel adder/subtractor using Full adders and XOR gates. [Pg- 25]
4. Convert a two-to- four line decoder with enable input 1*4 demultiplexer. [Pg- 25]
5. Realize JK flip flops. [Pg- 40]
6. How does ripple counter differ from synchronous counter?
7. Compare and contrast EEPROM and flash memory. [Pg- 53]
8. What is a Field Programmable Gate Arrays(FPGA) device? [Pg- 53]
9. Define ASM chart. List its Three basic elements. [Pg- 68]
10. What is critical race condition in asynchronous sequential circuits? Give an example. [Pg- 68]

**PART B- (5*16=80 Marks)**

11. (a)(i) Convert the following function into product of Max-terms.
    F(A,B,C)=(A+B')(B+C)(A+C'). (4)

(ii) Using Quine Mccluskey method, simplify the given function
F(A,B,C,D)=∑m(0,2,3,5,7,9,11,13,14). [Pg- 4] (12)

Or

(b) (i) Draw the multi-level two input NAND circuit for the following expression: F=(AB'+CD')E+BC(A+B).

(4)

(ii) Draw and explain Tri-State TTL inverter circuit diagram and explain its operation. [Pg- 13]

(12) 12. (a) (i) Design a 4 –bit decimal adder using 4 bit binary adders .

(10)

(ii) Implement the following Boolean functions using Multiplexers

$F(A,B,C,D)=\sum m(0,1,3,4,8,9,15)$ [MODEL Pg- 38] (6)

Or

(b) (i) Design a 4-bit magnitude comparator with three outputs: A> B,

A = B and A < B [Pg-30] (12)

(ii) Construct a 4-bit even parity generator circuit using gates. (4)

13.(a) (i) Design a 3-bit synchronous counter using JK flip-flops [Pg-47] (12)

(ii) Explain the differences between a state table, a characteristic table and an excitation table. (4)

Or

(b) Design the sequential circuit specified by the following state diagram using T flipflops. Check whether your design is self-correctable. (16)

14. (a) (i) Write short notes on EAPROM and static RAM cell using MOSFET

(6) (ii) Using eight 64* 8 ROM chips with an enable input and a decoder, construct a 512*8 ROM. (10)

Or

(b) (i) Use PLA with 3 inputs, 4 AND terms and two outputs to implement the following two Boolean functions [Pg- 56]

$F1(A,B,C)=\sum m(3,5,6,7)$ and $F2(A,B,C)=\sum m(1,2,3,4)$ (12)

(ii) Compare and contruct PLA and PAL. (4)

15.(a) (i) What is a hazard in an asynchronous sequential circuits? Define static hazard, dynamic hazard and essential hazard.[Pg- 77] (6)

(ii) Write and verify the HDL structural description of the four-bit register with parallel load. Use a 2*1 multiplexer for the flip-flop inputs. Include an asynchronous clear input. (10)

Or

(b) Design an asynchronous sequential circuit with inputs A and B and an output Y. Initially and at any time if both the inputs are 0,the output, Y is equal to 0.When A or B becomes 1,Y becomes 1.When the other input also becomes 1, Y becomes 0.The output stays at 0 until circuit goes back to initial state (16)

**ANNA UNIVERSITY**

B.E/B.Tech DEGREE EXAMINATION, NOVEMBER/DECEMBER 2016

Third Semester

Electronics and Communication Engineering

EC6302-DIGITAL ELECTRONICS

(Regulation 2013)

PART A - (10*2=20 marks)

1.Simplify the following expression X.Y+X(Y+Z)+Y(Y+Z).

2.Why totem pole outputs cannot be connected together.

3.Write about the design procedure for combinational circuits.

4.Draw the logic diagram and truth table of Full adder.

5.Define race round condition in flip flop.

6.Draw D-latch with truth table.

7. Briefly explain about EEPROM.

8. What is programmable logic array? How it differs from ROM?

9. Define Critical race and Non Critical race.

10. What is hazard and give it types?

PART B – (5*13=65 Marks)

11.(a) (i) Find the MSOP representation for F(A,B,C,D,E) = m(1,4,6,10,20,22,24,26)+d(0,11,16,27) using K-Map methods. Draw the circuits of the minimal expression using only NAND gates.

(ii) With neat circuit diagram, Explain the function of 3- inputs TTL NAND gate.

OR

(b) What are the advantages of using tabulation methods? Determine the Minimal sum of products for the Boolean expression F=$\sum$ ( 1,2,3,7,8,9,10,11,14,15) using tabulation method.

12. (a) (i) Design and explain 1 of 8 demultiplexer. (8)

(ii) What is parity checker? (5)

OR

(b) Design the operation of 3-Bit magnitude comparator.

13.(a) (i) Explain the operation of JK flip-flop with neat diagram. (6)

(ii) Explain the operation of Serial-in-serial-out Shift register. (7)

OR

(b) Design synchronous MOD-6 Counter. (13)

14.(a) Differentiate static and dynamic RAM. Draw the circuits of one cell of each and explain its working principle. (13)

OR

(b) Write short notes on.

(i) PAL

(ii) FPGA

15. (a) Explain the steps involved in the design of asynchronous sequential circuits.

OR

(b) Design an asynchronous circuits that will output only the second pulse received and ignore any other pulse.

PART C – (1*15 =15 Marks)

16. (a) Design a synchronous up/down Counter.

OR

(b) Design an even parity generator that generate an even parity bit for every input string of 3- bits.

**ANNA UNIVERSITY**

B.E/B.Tech DEGREE EXAMINATION, MAY/JUNE 2017

Third Semester

Electronics and Communication Engineering

EC6302-DIGITAL ELECTRONICS

(Regulation 2013)

PART A - (10*2=20 marks)

1. Convert the given decimal number to their binary equivalent 108.364, 268.025.(pg.no.6)
2. Show how to connect NAND gate to get an AND gate and OR gate. (pg.no.100)
3. Draw the truth table and logic circuit of Half adder. (pg.no.161)
4. Compare the function of Decoder and Encoder. (pg.no.205,220)
5. Derive the characteristic equation of D-FF. (pg.no.263)
6. What is the binary disadvantage of Asynchronous counter?. (pg.no.309)
7. How does RAM retain information? (pg.no.140)
8. Differentiate between PAL and PLA(pg.no.435,440)
9. What are the steps for analysis of Asynchronous Sequential Circuit?. (pg.no.469)
10. What is the significance of State Assignment? (pg.no.477)

PART B – (5*13=65 Marks)

11.(a) A using of K Map Find the MSB of F= m(0,2,3,12,13,14,15)+d(7,11) (13) (pg.no.57)

OR

(b) (i)State and Prove De Morgans Theorem (pg.no.44) (3)

(ii) Find the SOP and POS of F=B'C'D+BCD+ACD'+A'B'C+A'BC' (10) (pg.no.55,56)

12. (a) Implement Y=(A+C)(A+B')(A+B+C') using NOR gate (13) (pg.no.101)

OR

(b)i)Why does a good logic designer minimize the use of NOR gate (3) (pg.no.107)

ii)Show that is all the gates in two level AND-OR gate networks replaced by NAND gate.The output function does not change (10) (pg.no.103)

13.(a) Design and Explain the working of a Synchronous MOD 3 Counter (13)

(pg.no.312)

OR

(b) Using SR Flip flop design parallel counter which count will sequence 000, 111, 101, 110,001,010,000 (pg.no.260) (13)

14.(a) Compare static and dynamic RAM (pg.no.414,418) (3)

ii)Implement the Switching function (pg.no.440)

Z1=AB'D'E+A'B'C'E'+BC+BD

Z2=A'C'E

Z3=BC+DE+C'D'E'+BD

Z4=A'C'E+CE using 5*8*4 PLA

OR

(b) i)Difference between Boolean Addition and Binary Addition (3) (pg.no.387)

ii)Design a Combinational circuit using a ROM when accepts a 3 bit number and generate an output binary number equal to the square of the given number (10) (pg.no.187)

15. (a) i)Summarize the design procedure for a synchronous sequential circuit(10)

(pg.no.456)

16. ii)Derive the state table of serial binary Adder (pg.no.525) (3)

OR

(b) What is the objective of State Assignment in a Synchronous Counter? Give the Hazard free realization for the Boolean function F(A,B,C,D,E)=m(0,2,6,7,8,10,12) (pg.no.311) (13)

PART C – (1*15 =15 Marks)

17. (a) A sequential machine has one input line where 0's and 1's are being initiant.The machine has to produce the output of 1 when exactly two 1's are followed by a 1 or exactly 2 1's are followed by a 0 using any state assignment and JK flipflop. (pg.no.496) (15)

OR

(b) Find the expression for following function using Quines Mc Cluskey method.F=(0,2,3,5,9,11,13,14,16,18,24,26,28,30) (pg.no.67) 15)