DIGITAL LOGIC DESIGN

(REC-301)

B.Tech 2nd Year (EC & CS Branch)

Mr. Navneet Pal
Assistant Professor
Electronics & Communication Department

Syllabus
Digital Logic Design (REC-301)

Unit	Topic
I	Digital System And Binary Numbers: Number System and its arithmetic , Signed binary numbers, Binary codes, Cyclic codes, Hamming Code, the map method up to five variable, Don't care conditions, POS simplification, NAND and NOR implementation, Quine Mc-Clusky method (Tabularmethod).
II	Combinational Logic: Combinational Circuits: Analysis Procedure, Design procedure, Binary adder -subtractor, Decimal adder, Binary multiplier, Magnitude comparator, Multiplexers, Demultiplexers, Decoders, Encoders.
III	Sequential Logic And Its Applications : Storage elements: latches & flip flops, Characteristic Equations of Flip Flops, Flip Flop Conversion, Shift Registers, Ripple Counters, Synchronous Counters, Other Counters: Johnson & Ring Counter.
IV	Synchronous & Asynchronous Sequential Circuits Analysis of clocked sequential circuits with state machine designing, State reduction and assignments, Design procedure. Analysis procedure of Asynchronous sequential circuits, circuit with latches, design procedure, Reduction of state and flow table, Race-free state assignment, Hazards
V	Memory & Programmable Logic Devices: Digital Logic Families: DTL, DCTL, TTL, ECL & CMOS etc., Fan Out, Fan in, Noise Margin; RAM, ROM, PLA, PAL; Circuits of Logic Families, Interfacing of Digital Logic Families, Circuit Implementation using ROM, PLA and PAL; CPLD and FPGA.

UNIT-1

Content

- Digital System And Binary Numbers:
- Number System and its arithmetic
- Signed binary numbers
- Binary codes
- Cyclic codes,
- Hamming Code
- The map method up to five variable
- Don’t care conditions,
- POS simplification,
- NAND and NOR implementation,
- Quine Mc-Clusky method (Tabularmethod).

Very Short Questions and Answers

Q1. What are weighed codes? Give examples.

Ans: The weighed codes are the codes that have assigned weights or values for each bit position. A code in which each bit position has a certain numeric value assigned. Several such codes exist, such as 8-4-2-1, 7-4-2-1, 6-3-1 -1, Excess – 3 code.

Q2. What are cyclic codes?

Ans: An (n, k) linear code C is called a cyclic code if any cyclic shift of a codeword is another codeword. That is,

$$\textbf{if} \quad \bar{c} = (c_0, c_1, \cdots, c_{n-1}) \in C$$

$$\textbf{then} \quad \bar{c}^{(1)} = (c_{n-1}, c_0, c_1, \cdots, c_{n-2}) \in C$$

Q3. What are self complementing codes?

Ans: A self-complementing code is the one in which the members of the number system complement on themselves. This requires the following two conditions to be satisfied.

a). The complement of the number should be obtained from that number by replacing 1s with 0s and 0s with 1s.

b). The sum of the number and its complement should be equal to decimal 9. Example of a self-complementing code is 2-4-2-1 code, Excess-3 code. , BCD code

Q4. What are BCD codes?

Ans:A special binary code used to directly represent the decimal characters. Each four bit value in BCD represents a single decimal character.

Q5. What are alphanumeric codes?

Ans: A binary code used to represent the alphabets, numbers and punctuation marks as well as control characters for controlling a printer or display. Eg – ASCII, EBCD

Q6. What is an Excess – 3 code?

Ans: A self complementing code used to represent BCD numbers. It is self complementing because the 1's complement is also the 9's complement of the BCD number. It is widely used in BCD arithmetic circuits.
Examble: 1000 of 8421 = 1011 in Excess-3

Q7. What is a Gray code? *(or)* What is a unit distance code? Give an example

Ans:A reflective, unit distance code where only one bit position changes for each adjacent change in value.

Q8 What is a reflective code?

Ans: A code is said to be reflective when code for 9 is complement for the code for 0, and so is for 8 and 1 codes, 7 and 2, 6 and 3, 5 and 4. Codes 2421, 5211, gray and excess-3 are reflective, whereas the 8421 code is not.

Q9. What is meant by parity bit?

Ans: Parity bit is an extra bit included with a binary message to make the number of 1's either odd or even. The message, including the parity bit is transmitted and then checked at the receiving and for errors.

Q.10 Define duality property

Ans :Duality property states that every algebraic expression deducible from the postulates of Boolean algebra remains valid if the operators and identity elements are interchanged. If the dual of an algebraic expression is desired, we simply interchange OR and AND operators and replace 1's by 0's and 0's by 1's.

Q11. State De Morgan's theorem.

Ans: De Morgan suggested two theorems that form important part of Boolean algebra.
1) The complement of a product is equal to the sum of the complements.
(AB)' = A' + B'
2) The complement of a sum term is equal to the product of the complements.
(A + B)' = A'B'

Q12. What are called don't care conditions?

Ans: In some logic circuits certain input conditions never occur, therefore the corresponding output never appears. In such cases the output level is not defined, it can be either high or low. These output levels are indicated by 'X' or'd' in the truth tables and are called don't care conditions or incompletely specified functions

Q13. Convert the given expression in canonical SOP form Y = AC + AB + BC

Ans: Y = AC + AB + BC
=AC(B + B') + AB(C + C') + (A + A')BC
=ABC + ABC' + AB'C + AB'C' + ABC + ABC' + ABC
=ABC + ABC' +AB'C + AB'C'

Q14 Simplify F = ABC +AB'C + A'C + AB'

Ans:
F = ABC +AB'C + A'C + AB'
= AC(B+B') + A'C + AB'
= AC + A'C + AB'
= C + AB'

Q15. Simplify the expression Z= AB + AB'(A'C')'

Ans:
Z= AB + AB'(A'C')'
= AB + AB'(A+C)
= AB + AB' + AB'C
= A (B + B' +B'C)
= A (1+B'C) = A

Q16. What are the different classification of binary codes?

Ans • Weighted codes
• Non - weighted codes
• Reflective codes

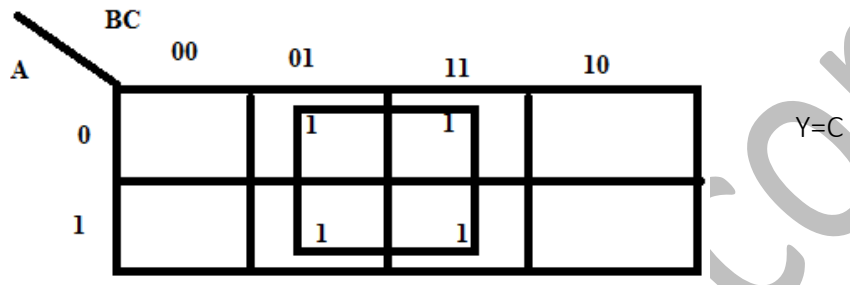
- Sequential codes
- Alphanumeric codes
- Error Detecting and correcting codes.

Q17. Why are NAND and NOR gates known as universal gates?

Ans: The NAND and NOR gates are known as universal gates, since any logic function can be implemented using NAND or NOR gates.

Q18 Simplify the function *Y = m(1,3,5,7)*

Ans:



Short and Long Type Questions

Q19.Explain or list out the advantages and disadvantages of K-map method?

Ans: The ***advantages*** of the K-map method are

i. It is a fast method for simplifying expression up to four variables.
ii. It gives a visual method of logic simplification.
iii. Prime implicants and essential prime implicants are identified fast.
iv. Suitable for both SOP and POS forms of reduction.
v. It is more suitable for class room teachings on logic simplification.

The ***disadvantages*** of the K-map method are

i. It is not suitable for computer reduction.
ii. K-maps are not suitable when the number of variables involved exceed four.
iii. Care must be taken to fill in every cell with the relevant entry, such as a 0, 1 (or) don't care terms

Q20:What is Hamming code?

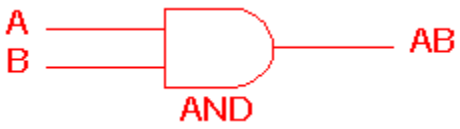
Ans: One of the most common error-correcting codes used in random-access memories was devised by R. W. Hamming. In the Hamming code, *k* parity bits are added to an *n-bit* data word, forming a new word of *n* + *k* bits. The bit positions are numbered in sequence from I to *n* + *k.* Those positions numbered as a power of 2 are reserved for the parity bits. The remaining bits are the data bits. The code can be used with words of any length.

Q21:c Explain Logic Gates.

Ans: Logic gates

Digital systems are said to be constructed by using logic gates. The basic gates are the AND, OR, NOT gates. The basic operations are described below with the aid of tables in the following, called truth tables.

AND gate



2 Input AND gate		
A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

The AND gate is an electronic circuit that gives a **high** output (1) only if **all** its inputs are high. A dot (.) is used to show the AND operation i.e. A.B. Bear in mind that this dot is sometimes omitted i.e. AB

OR gate

2 Input OR gate		
A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

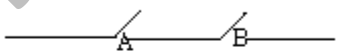
The OR gate is an electronic circuit that gives a high output (1) if **one or more** of its inputs are high. A plus (+) is used to show the OR operation.

Truth table for AND,

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

Z=A.B

Switch realisation of AND:



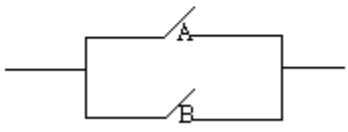
Continuity occurs only when both A AND B are closed.

Truth table for OR,

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

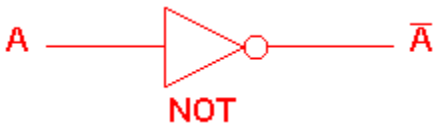
Z=A+B

Switch realisation of OR:



Continuity if A or B or both are closed.

NOT gate



NOT gate	
A	$\overline{A}$
0	1
1	0

The NOT gate is an electronic circuit that produces an inverted version of the input at its output. It is also known as an *inverter*. If the input variable is A, the inverted output is known as NOT A. This is also shown as A', or A with a bar over the top, as shown at the outputs.

Q22 Explain Principle of DUALITY.

Ans: PRINCIPLE OF DUALITY

According to this principle, *For every valid expression in Boolean Algebra there exists an equally valid dual expression.* The dual expression can be obtained by following the following three steps:

i) Complement each 0 and 1 (Change the 0's to 1's and 1's to 0's)
ii) Replace each OR (+) sign by AND (.) and each AND (.) sign by OR (+)
iii) Leave NOTs unchanged.

The dual of i) X.1 = X is X + 1= X
ii) X + (YZ) is X . (Y+ Z)
iii) X.(Y+Z) is X+ (Y.Z)

Q23 Write Postulates of Boolean Algebra.

Ans: Basic Postulates of Boolean Algebra

a) A+ 0 = A

b) A+1 = 1, In fact 1 OR anything will be 1 as , as long as one of the inputs for the OR operation is true the output will be true.

c) A. 0 = 0 , 0 AND anything will be 0 as all the inputs for the AND operation must be true for the output to be true.

d) A.1 = A

e) A+ A'= 1, As if A is 0 , A' will be 1 and if A is 1 ,A' will be 0. 1 OR anything is 1 A+A' , is *Tautology* as its value is always true.

f) A. A' = 0, As either a or A' will be 0 as 0 and anything is 0.
A.A' is *Fallacy* as its value is always false.

g) A+A= A ↕ Proof: A+A= (A+A).1=(A+A)(A+A')=A+A.A'=A
Idempotent Law

h) A. A = A Proof: A.A=A.A+0=A.A+A.A'=A(A+A')= A.1=A

i) A.B=B.A

j) A+B=B+A

Proof : By making the truth table and showing LHS=RHS

Commutative Law

k) (A+B)+C=A+(B+C)

l) (A.B).C=A.(B.C)

Proof: By truth table

Associative Law

m) A+ AB = A ,

n) A.(A+B)=A

Proof: A+AB= A(1+ B) = A. 1= A

Absorption Law

o) A.(B+C)=AB+AC

p) A+BC=(A+B).(A+C)

Proof: (A+B).(A+C)=AA+AC+AB+BC

Distributive Law =A(1+A+C+B)+BC

=A+BC

q) A'' = A

Involution

r) A+ A'B = A+B

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

= A(B+B')+B(A+A') =A+B

Q24 Write De' Morgen's Law

Ans: De'Morgan's Theorems

De'Morgan's Theorems' are useful in changing the forms of Boolean expressions . The two theorems are:

i) The complement of the sum is equivalent to the product of individual complements.

(A+B)' = A' . B'

ii) The complement of the product is equal to the sum of individual complements.

(A.B)' = A' + B'

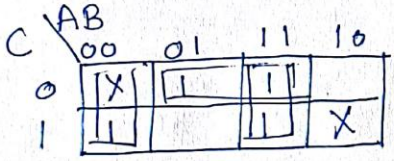
Q 25 Convert the following Boolean Function into Standard SOP & express it in terms of minterms. $Y(A,B,C) = AB + AC + B\bar{C}$

Sol

$$Y(A,B,C) = AB + Ac + B\bar{C}$$
$$= ABC(C+\bar{C}) + AC(B+\bar{B}) + B\bar{C}(A+\bar{A})$$
$$= \underline{ABC} + \underline{AB\bar{C}} + \underline{ABC} + A\bar{B}C + \underline{AB\bar{C}} + \bar{A}B\bar{C} \quad [SOP$$
$$Y(A,B,C) = ABC + AB\bar{C} + A\bar{B}C + \bar{A}B\bar{C}$$
$$Y(A,B,C) = \Sigma m(7,6,5,2)$$

Q 26 Minimize the following Expression using K-Map $Y(A,B,C) = \Sigma(1,2,6,7) + d(0,5)$

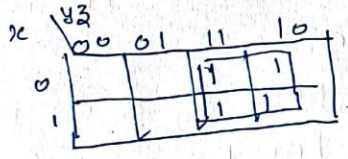
Sol $Y(A,B,C) = AB + B\bar{C} + \bar{A}\bar{B}$



Q 27 Simplify the Boolean function is (i) SOP (ii) POS $f(x,y,z) = \Sigma(2,3,6,7)$

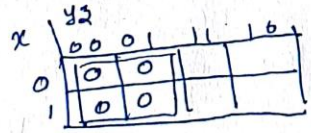
Sol

SOP Form



$\Rightarrow F = Y$

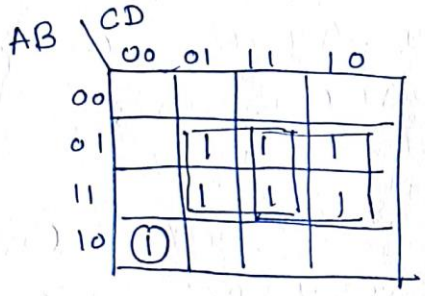
POS Form

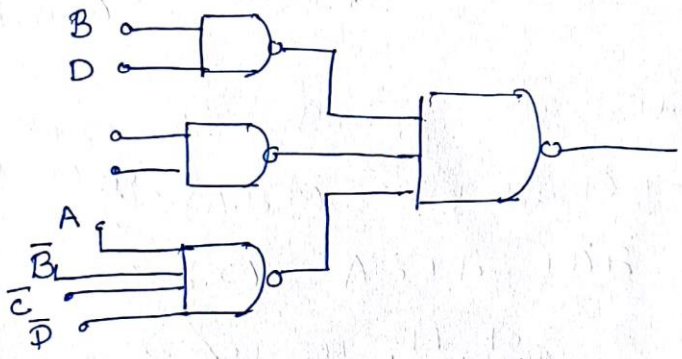


$\Rightarrow F = Y$

Q 28 Simplify the following Expression & implement it with two-level NAND gate circuit

$$Y = BD + BC\bar{D} + A\bar{B}\bar{C}\bar{D}$$



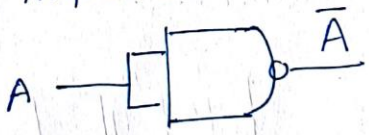


Q 29 Explain Universal Gates

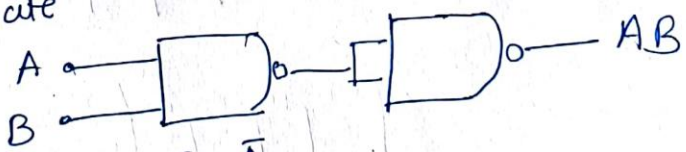
Ans A Gate that Can Perform all the logical operation (i) NAND (ii) NOR

(i) NAND Gate implementation

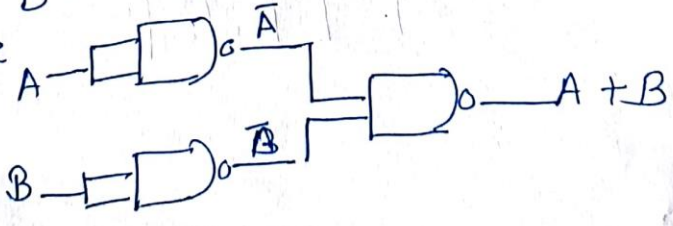
NOT Gate



AND Gate

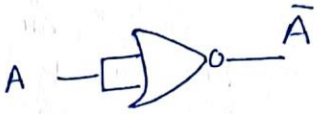


OR Gate

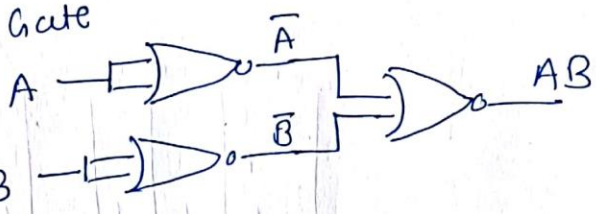


(ii) NOR Gate implementation

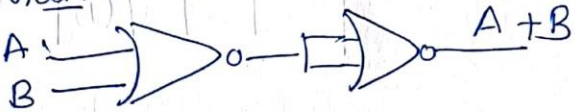
NOT Gate



AND Gate



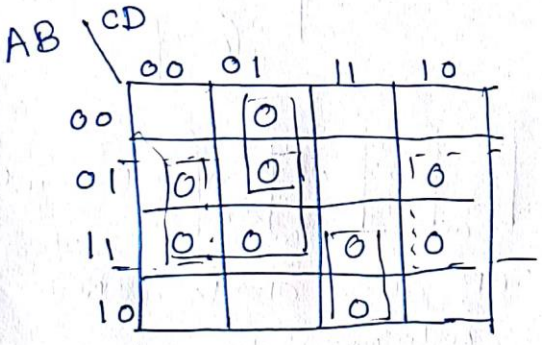
OR Gate



Q. 30 Minimize using K-Map

$f = \pi(1, 4, 5, 6, 11, 12, 13, 14, 15)$

Sol



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

Q 31 Convert the Base 5 Number $(4433214)_5$ to Base 12 ,

Sol Step 1 Base 5 to Base 10

$$\begin{array}{ccccccc} 4 & 4 & 3 & 3 & 2 & 1 & 4 \\ 5^6 & 5^5 & 5^4 & 5^3 & 5^2 & 5^1 & 5^0 \end{array}$$

$$= 4\times5^6 + 4\times5^5 + 3\times5^4 + 3\times5^3 + 2\times5^2 + 1\times5^1 + 4\times5^0$$

$$= 62500 + 12500 + 1875 + 375 + 50 + 5 + 4$$

$$= (77309)_{10}$$

Step 2 Base 10 to Base 12

12 | 77309		
12 | 6442	5	
12 | 536	10 (A)	↑
12 | 44	8	
3	8	

So $(4433214)_5 = (77309)_{10} = (38A85)_{12}$

Practice Questions

NUMBER SYSTEM & BOOLEAN ALGEBRA

1. Determine the decimal value of the fractional binary number 0.1011.
2. Perform 2's complement subtraction of 010110-100101
3. Convert (53) 10 to EX-3 code.
4. Why digital circuits are more frequently constructed with NAND or NOR gates than with AND & OR gates .
5. Convert 110011 into hexadecimal through octal.
6. What is variable mapping?
7. What is the feature of gray code?
8. Name the two canonical forms for Boolean algebra.
9. What is the BCD equivalent for the gray code 1110?

10. Obtain the minimum sop using QUINE- McCLUSKY method and verify using K-map
    F=m0+m2+m4+m8+m9+m10+m11+m12+m13.

11. Reduce the following using tabulation method.
    F=m2+m3+m4+m6+m7+m9+m11+m13.

12. Reduce the Boolean function using k-map technique and implement using gates
    f (w, x, y,z)= ∑m (0,1,4,8,9,10) which has the don't cares condition
    d (w, x, y, z)= ∑ m (2,11).

13. Find the minimum SOP expression using K-map for the function
    f= ∑m (7, 9, 10, 11, 12, 13, 14, 15) and realize the minimized function using only NAND gates.

15. Expand the following Boolean expression to minterms and maxterms
    A+BC'+ABD'+ABCD

16 .Prove the following (A+B) ((AC)'+C) (B'+AC)'=A'B.