Subject: DECO
Faculty: A. Swathi
Topic: Number system

Class Notes

Unit No: I
Lecture No:
Date Conducted:

Number system:

A system that is used for representing numbers is called the number system.

- In digital electronics, the numbers are used to represent the information.
- Hence, it is important to learn and understand different types of number systems so we can easily represent and interpret the information in the form of numbers.
- There are several types of number systems and the basis of this classification is the base or radix of the number of symbols used to denoted the numbers in the number system.

Types of Number Systems:

Depending on the base or radix, number systems can be classified into 4 major types.

They are
1. Decimal Number system
2. Binary Number system
3. Octal Number system
4. Hexadecimal Number system.

- The number system which we human beings commonly use is the decimal number system consisting of digits 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.
- But in computers, instructions as well as data are stored in binary number system consisting of digits '0' & '1'.

It is so because binary digits 0 and 1 can be easily represented by an electrical switch. If the switch is closed it could be '1' and if it is open it could be '0'.

Decimal Number system :

The system of numbers which has base or radix 10, i.e uses total 10 symbols to represent numbers of the system is called decimal number system.

* The symbols used in the decimal number system are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ; where each of these symbols assigned a specific value.
* It is a positional weighted system. i.e the value attached to a symbol depends on its location with respect to the decimal point.
* Each symbol in the number is called a digit.
* The left most digit in any number representation, which has the greatest positional weight out of all the digits represent in that number, is called the Most significant Digit (MSD).
* The right most digit, which has the least positional weight out of all the digits represent in that number, is called the least significant digit (LSD).
* The digits on the left side of the decimal point form the Integer part of a decimal number and

those on the right side form the fractional part.

* The digits to the right of the decimal point have weights which are negative powers of 10 and
* The digits to the left of the decimal point have weights which are positive powers of 10.
* The value of a decimal number is the sum of the products of the digits of that number with their respective column weights.
* The weight of each column is 10 times greater than the weight of the column to its right.

In general, the value of any decimal number

$d_n\, d_{n-1}\, d_{n-2} \cdots d_1\, d_0$.

Ex:

Let us consider a number 247.

The number is said to have 3 digits (2, 4 and 7) Each one known as decimal digit.

The digit "2" in view of its position has a value equal to 200

Illy 4 has a value equal to 40 and 7 has a value 7.

Hence the value of the given number can be given as

$$247 = 2\times100 + 4\times10 + 7\times1$$
$$= 2\times10^2 + 4\times10^1 + 7\times10^0$$

∴ The position value of 7 is 1, the positional value of 4 is 10, and 2 is 100.

In general, the value of any mixed decimal number is given by

$d_n d_{n-1} d_{n-2} \cdots d_2 d_1 d_0 \cdot d_{-1} d_{-2} d_{-3} \cdots d_{-k}$

$(d_n \times 10^n) + (d_{n-1} \times 10^{n-1}) + \cdots + (d_1 \times 10^1) + (d_0 \times 10^0) +$

$(d_{-1} \times 10^{-1}) + (d_{-2} \times 10^{-2}) + \cdots$

Ex:

Consider the mixed decimal number 9256.26 using digits 2, 5, 6, 9.

$9256.26 = 9 \times 1000 + 2 \times 100 + 5 \times 10 + 6 \times 1 + 2 \times (1/10) + 6 \times (1/100)$

$= 9 \times 10^3 + 2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 + 2 \times 10^{-1} + 6 \times 10^{-2}$

The positional value of 2 should be less than that of 6 by 10 times (and so equal to 10^{-1})

The digit 6 is the least significant digit

The digit 9 is the most significant digit.

* Consider another number 6592.69 using the same digits 2, 5, 6, 9

Here

$6592.69 = 6 \times 10^3 + 5 \times 10^2 + 9 \times 10^1 + 2 \times 10^0 + 6 \times 10^{-1} + 9 \times 10^{-2}$

The digit 9 is the most significant digit

The digit 6 is the least significant digit.

Note: The difference in positional values of the same digits, when placed in different positions.

Binary Nomber system:

The binary number system is a positional weighted system.

* The base or radix of this number system is 2. Hence, it has two independent symbols.
* The base itself cannot be a symbol
  The symbols used are '0' and '1'.
* A binary digit is called a bit.
* A binary point seperates the integer and fractional parts.
* The weight of each bit position is one power of 2 greater than the weight of the position to its immediate right.
* The first bit to the left of the binary point has a weight of 2^0 and that column is called the units column.
* The second bit to the left has a weight of 2^1 and it is in the 2's column.
* The third bit to the left has a weight of 2^2 and it is in the 4's column, and so on.
* The first bit to the right of the binary point has a weight of 2^{-1} and it is said to be in the $\frac{1}{2}$'s column, the next right bit with wt of 2^{-2} is in the $\frac{1}{4}$'s column, and so on.

Ex:

Let us consider a binary number 1010

- This number consists of 4 binary digits (BITS)
- The decimal value of the binary number is the sum of the products of all its multiplied by the weights of their respective positions.

$$1010 = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0$$

- The value of 1010.10 can be given as

$$1010.10 = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0 + 1\times 2^{-1} + 0\times 2^{-2}$$

- In this case, the point is known as the binary point.
- The binary digit '0' present after the binary point is now LSB.
- The binary number system is used in digital computers because the switching circuits used in these computers use two state devices such as transistors, diodes etc.
- A transistor can be OFF or ON, a switch can be OPEN or CLOSED, a diode can be OFF or ON etc.
- These devices have to exists in one of the two possible states.

So, these two states can be represented by the symbols 0 and 1 respectively.

Subject: DECO
Faculty: A. Swathi
Topic: octal Number system

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 4

Octal Number System:

As the number value increases, the no. of digits to be used in the binary number system increases.

* In order to make it easier for the user, the octal number system was used.
* It has a base or radix 8.
* It consists of digits 0, 1, 2, 3, 4, 5, 6 and 7.
* In this system, all the numbers are given as combination of these symbols.
* It is an important system which is often used in microcomputers.

Ex:

Let us consider an octal number 352.

This number consists of three octal digits.

* Each of these digits has a positional value, i.e weightage to each position will be powers of 8.
* The right most digit '2' has the least weightage. The left most digit '3' has the heighest weightage.
* The value of the octal number 352, is given by

$$352 = 3\times8^2 + 5\times8^1 + 2\times8^0.$$

Hexadecimal Number system :

The hexadecimal number system has a base of 16 this means this system has 16 (Hexa+decimal) symbols.

- Since the basic numbers we used to have only ten digits (from 0 to 9) The first six letters of the english alphabets are borrowed to make the remaining 6 symbols.
- Hence this system is made up of the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F.
- All the numbers are given as combination of these symbols.
- Each of these digits has a positional value.
- The weightage to each position will be a power of 16.

Ex :
Let us consider a hexadecimal number 5C.
This number consists of two hexa decimal digits.
The right most digit 'C' has the least weightage.
The left most digit '5' has the most weightage.
The value of the 5C should probably written as

$$5C = 5 \times 16^1 + C \times 16^0$$

'C' corresponds to the decimal no 12,

So, $5C = 5 \times 16^1 + 12 \times 16^0$.

Subject: DECO
Faculty: A. Swathi
Topic: Conversions

Class Notes

Unit No: I
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 5

Conversion of numbers from one radix to another radix.

Decimal to Binary:

There are many ways of converting decimal numbers to binary numbers the most commonly used is the dibble-dabble method.

* To convert decimal number system to binary number system, divide the given number successively by 2 till the quotient is zero and read the remainders from bottom to top for integer.
* To convert the given decimal fraction to binary, successively multiply the decimal fraction by 2 till the product is '0' or till the required accuracy is obtained.
  Keep the integer in the product as it is and read the integers in the product from top to bottom.

Ex:

1. Convert 52_{10} to binary using double-dabble method.

Sol: We divide the given decimal number successively by 2

		successive division	Remainders
2	52		
2	26	-	0
2	13	-	0
2	6	-	1
2	3	-	0
2	1	-	1
	0	-	1

read the remainders from bottom to top

read the remainders upwards (i.e from bottom to up) to get equivalent binary number.

$\therefore (52)_{10} = (110100)_2$

Ex: Convert $(105.15)_{10}$ to binary using dibble-dabble method.

Sol The given decimal number is a mixed number convert integer and fraction part seperately.

Conversion of 105_{10}

Successively divide by 2 till the quotient is zero and read the remainders from bottom to top.

successive division

2	105	
2	52	-1
2	26	-0
2	13	-0
2	6	-1
2	3	-0
2	1	-1
	0	-1

$(105)_{10} = (1101001)_2$

Conversion of $(0.15)_{10}$

Multiply the given fraction by 2

Keep the integer in the product as it is and multiply the new-fraction in the product by 2.

Continue this process till the product is zero and read the integers in the products from top to bottom.

Given fraction	0.15
0.15 by 2	0.30
0.30 by 2	0.60
0.60 by 2	1.20
0.20 by 2	0.40
0.40 by 2	0.80

$(0.15)_{10} = (0.001001)_2$

∴ The final result is

$(105.15)_{10} = (1101001.001001)_2$

Decimal to octal :

To convert a mixed decimal number to a mixed octal number, convert the integer and fraction parts seperately.

* To convert the given decimal integer number to octal successively divide the given number by 8 till the quotient is 0. The last remainder is the MSB.
* The remainders read upwards that gives the equivalent octal integer number.

Ex :

Convert 378_{10} to octal

Sol We divide the given decimal number by 8 successively and read the remainders from bottom to up.

Successive division		Remainders
8 | 378		
8 | 47		↑ 2
8 | 5		7
0		5

Read the remainders from bottom to up.

Therefore, $378_{10} = 572_8$

* To convert the given decimal fraction to octal, successively multiply the decimal fraction by 8 till the product is '0' or till the required accuracy is obtained.
The first integer from the top is the MSB
The integers to the left of the octal point read downwards that give the octal fraction.

Ex:
Convert $(0.93)_{10}$ to octal

0.93×8 → 7.44
0.44×8 → 3.52
0.52×8 → 4.16
0.16×8 → 1.28

Read the integers to the left of the octal point downwards.

$\therefore (0.93)_{10} = (0.7341)_8$

* Conversion of large decimal numbers to binary and large binary numbers to decimal can be conveniently ~~the octal number~~ to and quickly performed via octal.

Ex:
Convert $10111101000|_2$ to decimal

Sol Since the given binary number is large. we first convert this number to octal then convert octal number to decimal

$(101111010001)_2 = (5721)_8$

$(5721)_8 = 5\times8^3 + 7\times8^2 + 2\times8^1 + 1\times8^0$

$= 2560 + 448 + 16 + 1$

$(5721)_8 = (3025)_{10}$

Ex:

Convert $(5497)_{10}$ to binary

Sol: Since the given decimal number is large, we first convert this number to octal and then convert the octal number to binary.

Successive division		Remainders
8	5497	↑ 1
8	687	7
8	85	5
8	10	2
8	1	1
	0	

Therefore,

$(5497)_{10} = (12571)_8$

$(12578)_8 = (001010101111001)_2$

Decimal to Hexadecimal :

To convert decimal to hexadecimal, successively divide the given number by 16 till the quotient is 0. The last remainder is MSB.
The remainders read from bottom to top give the equivalent hexadecimal integer.

* To convert hexadecimal fraction to hexadecimal, successively multiply the given decimal fraction and subsequent decimal fraction by 16, till the product is '0' or the required accuracy is obtained, and collect all the integers to the left of the decimal point.
The first integer is the MSB and the read the integers from top to bottom give the hexadecimal fraction. This is known as the hexdabble method.

Ex :
Convert $(2598.675)_{10}$ to hex

Sol) The given decimal number is a mixed number. Convert the integer and fraction parts seperately

Conversion of 2598

16	2598
16	162
16	10
	0

Remainder

Decimal	Hex
6	6
2	2
10	A

Reading the remainders upwards,

$2598_{10} = A26_{16}$.

Conversion of 0.675

Given fraction is 0.675

0.675×16	$\downarrow$ 10.8
0.800×16	$\downarrow$ 12.8
0.800×16	$\downarrow$ 12.8
0.800×16	$\downarrow$ 12.8

Reading the integers to the left of hexadecimal point downwards, $0.675_{10} = 0.ACCC_{16}$

$\therefore 2598.675_{10} = A26.ACCC_{16}$

Ex:

Convert 49056_{10} to binary

Sol) The given number is very large. It is tedious to convert this to binary directly. So, convert this to hex first and then convert the hex to binary.

16 | 49056	
16 | 3066	
16 | 191	
16 | 11	
0	

Remainder

Decimal	Hex	Binary
0	$\uparrow$ 0	0000
10	| A	1010
15	| F	1111
11	| B	1011

$\therefore 49056_{10} = BFA0_{16} = (1011\ 1111\ 1010\ 0000)_2$

Binary to octal :

To convert a binary number to an octal number, starting from the binary point make groups of 3 bits each, on either side of the binary point, and replace each 3-bit binary group by the equivalent octal digit.

Ex :

1. Convert 110101.10101_2 to octal

Sol Groups of 3 bits are $\underline{110}\ \underline{101}\ .\ \underline{101}\ \underline{010}$

convert each group to octal $\quad 6\ \ 5\ .\ \ 5\ \ 2$

∴ The result is $(65.52)_8$

2. Convert 10101111001.0111_2 to octal

Sol Groups of 3 bits are $\underline{010}\ \underline{101}\ \underline{111}\ \underline{001}\ .\ \underline{011}\ \underline{100}$

convert each group to octal $\quad 2\ \ 5\ \ 7\ \ 1\ .\ \ 3\ \ 4$

∴ The result is $(2571.34)_8$

$\therefore (10101111001.0111)_2 = (2571.34)_8$

Subject: DECO
Faculty: A. Swathi
Topic: Conversions

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 9

Binary to Decimal :

Binary numbers may be converted to their decimal equivalents by the positional weights method.

* In this method, each binary digits of the number is multiplied by its position weight and the product terms are added to obtain the decimal number.

Ex :

Convert 10101_2 to decimal

Sol

positional weights $2^4\ 2^3\ 2^2\ 2^1\ 2^0$

Binary number $1\ 0\ 1\ 0\ 1$

$\therefore (10101)_2 = (1\times2^4) + (0\times2^3) + (1\times2^2) + (0\times2^1) + (1\times2^0)$

$= 16 + 0 + 4 + 0 + 1$

$= 21_{10}$

Ex :

Convert 11011.101_2 to decimal

Sol

$2^4\ 2^3\ 2^2\ 2^1\ 2^0\ 2^{-1}\ 2^{-2}\ 2^{-3}$

$1\ 1\ 0\ 1\ 1 . 1\ 0\ 1$

$= (1\times2^4) + (1\times2^3) + (0\times2^2) + (1\times2^1) + (1\times2^0) + (1\times2^{-1}) + (0\times2^{-2}) + (1\times2^{-3})$

$= 16 + 8 + 0 + 2 + 1 + 0.5 + 0 + 0.125$

$= 27.625_{10}$

$\therefore (11011.101)_2 = (27.625)_{10}$

Binary to Hexadecimal:

To convert a binary number to a hexadecimal number, starting from the binary point, make groups of 4 bits each, on either side of the binary point and replace each 4-bit group by the equivalent hexa decimal digit.

Ex:

Convert 10110111011_2 to hexadecimal

Sol Make groups of 4 bits, and replace each 4-bit group by a hex digit.

Given binary number is 10110111011

Groups of 4 bits are 0010 1101 1011

Convert each group to hex 2 D B

The result is $(2DB)_{16}$

Ex:

Convert 0101111011.011111_2 to hexadecimal

Sol

Given binary number is 0101111011.011111

Groups of 4 bits are 0010 1111 1011 . 0111 1100

Convert each group to hex 2 F B 7 C

∴ The result is $(2FB.7C)_{16}$

Subject: DECO
Faculty: A. Swathi
Topic: octal to Decimal

Class Notes

Unit No:
Lecture No:
Date Conducted:
Page No: 10

Octal to Decimal:

To convert an octal number to a decimal number, multiply each digit in the octal number by the weight of its position and add all the product terms.

The decimal value of the octal number $d_n d_{n-1} \cdots d_1 d_0 \cdot d_{-1} d_{-2} \cdots d_{-k}$ is $(d_n \times 8^n) + (d_{n-1} \times 8^{n-1}) + \cdots + (d_0 \times 8^0) + (d_{-1} \times 8^{-1}) + \cdots$

Ex:
Convert 4057.06_8 to decimal

Sol $4057.06_8 = 4\times8^3 + 0\times8^2 + 5\times8^1 + 7\times8^0 + 0\times8^{-1} + 6\times8^{-2}$

$= 2048 + 0 + 40 + 7 + 0 + 0.0937$

$= 2095.0937_{10}$

Octal to Binary:

To convert a given octal number to binary, just replace each octal digit by its 3-bit binary equivalent.

Ex:
Convert 367.52_8 to binary

Sol Given octal number is 3 6 7 . 5 2

Convert each octal digit to binary 011 110 111 . 101 010

The result is $(011110111 . 101010)_2$

Octal to Hexadecimal :

To convert an octal number to hexadecimal, the simplest way is to first convert the given octal number to binary and then the binary number to hexadecimal.

Ex :

Convert 756.603_8 to hex

Sol

Given octal number is 7 5 6 . 6 0 3

Convert each octal digit to binary 111 101 110 . 110 000 011

Group of four bits are 0001 1110 1110 . 1100 0001 1000

Convert each four-bit group to hex 1 E E . C 1 8

The result is $(1EE.C18)_{16}$

Hexadecimal to Binary :

To convert a hexadecimal number to binary, replace each hex digit by its 4-bit binary group.

Ex :

Convert $4BAC.B0_{16}$ to Binary

Sol

Given hex number is 4 B A C . B 0

Convert each hex digit to 4-bit binary. 0100 1011 1010 1100 . 1011 0000

The result is $(0100\ 1011\ 1010.1100 . 1011\ 0000)_2$

Subject: DECO
Faculty: A. Swathi
Topic: Hexadecimal to Decimal

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 11

Hexadecimal to Decimal:

To convert a hexadecimal number to decimal, multiply each digit in the hex number by its position weight and add all those product terms.

Ex:

Convert $5C7_{16}$ to decimal

Sol: Multiply each digit of 5C7 by its position weight and add the product terms.

$$5C7_{16} = (5\times16^2)+(12\times16^1)+(7\times16^0)$$
$$= 1280+192+7 = 1479_{10}$$

Hexadecimal to octal:

To convert a hexadecimal number to octal, the simplest way is to first convert the given octal no to binary and then the binary number to octal.

Ex:

Convert $B9F.AE_{16}$ to octal

Sol: Given hex number is B 9 F . A E

Convert each hex digit to binary 1011 1001 1111 . 1010 1110

Group of three bits are 101 110 011 111 . 101 011 100

Convert each 3 bit group to octal 5 6 2 7 . 5 3 4

The resut is $(5627.534)_8$.

Binary Arithmetic:

Arithmetic operations such as addition, subtraction, multiplication and division can be carried out in any number system.

* The method followed is similar to the method followed in the decimal number system.

Binary Addition:

The rules for binary addition are

$0+0=0$ $\quad 0+1=1$ $\quad 1+0=1$ $\quad 1+1=10$ i.e 0 with carry 1

Ex:
Add the binary numbers 1101.101 and 111.011

Sol

8	4	2	1	.	$\frac{1}{2}$	$\frac{2}{2}$	$\frac{3}{2}$
1	1	0	1	.	1	0	1
+	1	1	1	.	0	1	1
1	0	1	0	1	.	0	0	0

In the 2^{-3}'s column $1+1=0$, with a carry of 1 to the 2^{-2} column.
In the 2^{-2}'s column $0+1+1=0$, with a carry of 1 to the 2^{-1} column
In the 2^{-1}'s column $1+0+1=0$, with a carry of 1 to the 1's column
In the 1's column $1+1+1=1$, with a carry of 1 to the 2's column
In the 2's column $0+1+1=0$, with a carry of 1 to the 4's column
In the 4's column $1+1+1=1$, with a carry of 1 to the 8's column
In the 8's column $1+1=0$, with a Carry of 1 to the 16's Column

Subject: DECO
Faculty: A. Swathi
Topic: Binary subtraction

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 12

Binary subtraction:

The binary subtraction is performed in a manner similar to that in decimal subtraction.

The rules for binary subtraction are

$0-0=0$; $0-1=1$; with a borrow of 1

$1-0=1$; $1-1=0$.

Ex:

Subtract 111.111_2 from 1010.01_2

Sol

8 4 2 1	$2^{-1}\ 2^{-2}\ 2^{-3}$	(column numbers)
1 0 1 0	. 0 1 0	
1 1 1	. 1 1 1	
0 0 1 0	. 0 1 1	

In the 2^{-3} column, a '1' can't be subtracted from a '0'. So, borrow a '1' from the 2^{-2} column, making the 2^{-2} column. The '1' borrowed from the 2^{-2} column becomes 10 in the 2^{-3} column.

∴ In the 2^{-3} column $10-1=1$.

In the 2^{-2} column $1-1=0$

In the 2^{-1} column $1-1=0$

In the 1's column $1-1=0$

Now, in the 2's column, a '1' can't be subtracted from a '0'. So, borrow a '1' from the 4's column. But in the 4's column has a '0'.

In the 2's column $10-1=1$

In the 4's column $1-1=0$

In the 8's column $0-0=0$

Hence, the result is $(0010.011)_2$

Binary Multiplication:

There are two methods of binary multiplication- The paper method and the computer method. Both the methods obey the multiplication rules.

$0 \times 0 = 0$; $0 \times 1 = 0$; $1 \times 0 = 0$; $1 \times 1 = 1$

Ex:

Multiply 1101_2 by 110_2

Sol

$$\begin{array}{r} 1101 \\ \times\ 110 \\ \hline 0000 \\ 1101 \\ 1101 \\ \hline 1001110 \end{array}$$

Binary Division:

Like multiplication, division too can be performed by two methods. The paper method, and the computer method.

Ex:

Divide 101101_2 by 110_2

Sol Divisor 110 can't go in the first three bits of the dividend i.e 101. so, consider the 1st 4 bits 1011 of dividend.

$$\begin{array}{r|l} 110 \overline{)\ 101101} & (111. \\ 110 & \\ \hline 1010 & \\ 110 & \\ \hline 1001 & \\ 110 & \\ \hline 110 & \\ 110 & \\ \hline 000 & \end{array}$$

Therefore, $101101 \div 110 = 111.1$

Subject: DECO
Faculty: A. Swathi
Topic: Complements

Class Notes

Unit No: I
Lecture No:
Date Conducted:

Complements :

Complements are used in digital computers for simplifying the subtraction operation and for logical manipulation.

There are two types of complements for each base 'r' system

1. The r's complement and
2. The (r-1)'s complement

* when the value of the base 'r' is substituted in the name, the two types are reffered to as the 2's and 1's complement for binary numbers and 10's and 9's complement for decimal numbers.

(r-1)'s complement :

Given a number 'N' in base 'r' having 'n' digits, the (r-1)'s complement of N is defined as

$$(r^n - 1) - N$$

* For decimal numbers $r = 10$ and $r - 1 = 9$.

So, the 9's complement of N is $(10^n - 1) - N$

Now, 10^n represents a number that consists of a single 1 followed by numbers.

$10^n - 1$ is a number represented by n 9's.

* For example, with $n = 4$, we have $10^4 = 10000$

and $10^4 - 1 = 9999$.

It follows that the 9's complement of a decimal no is obtained by subtracting each digit from 9.

Ex: Find the 9's complement of 546700

Sol: The 9's complement of 546700 is,

$$\begin{array}{r} 999999 \\ -546700 \\ \hline 453299 \end{array}$$

For binary numbers:

$r=2$ and $r-1=1$

So, the 1's complement of N is $(2^n-1)-N$.

* 2^n is represented by a binary number that consists of a '1' followed by n 0's.
* 2^n-1 is a binary number represented by n 1's.

For example,

with $n=4$, we have $2^4=(10000)_2$

and $2^4-1=(1111)_2$

Thus, the 1's complement of a binary number is obtained by subtracting each digit from 1.

1's complement

1's complement of a binary number is formed by changing 1's into 0's and 0's into 1's.

* Binary complements are used to represent the negative numbers.

The subtraction of binary number is carried out by taking the complement of the number and adding.

Ex:

① 1's complement of 1011 is 0100.

② 1's complement of 01111 is 10000.

→ subtract $(1010)_2$ from $(1100)_2$ by using 1's complement

Sol) This is explained step by step

Step$_1$: The 1's complement of 1010 is 0101

Step$_2$: This is added to 1100

which gives

$$\begin{array}{r} 1100 \\ 0101 \\ \hline 10001 \end{array}$$

1's complement of 1010

Step$_3$: It can be noticed that the operands have 4 bits while the result of addition has 5 bits.

* The extra bit (1) is represent at the MSB is known as the end arround carry. (eac)
* When we get eac, it means that the difference is positive and the actual value is obtained by adding the end carry to the LSB
* In case of end carry is not there, there it means that the difference is negative and the actual value is obtained by taking the 1's complement of this result.

$$\begin{array}{r} 1100 \\ 0101 \\ \hline 10001 \\ \longrightarrow 1 \\ \hline 0010 \end{array}$$

Taking the end arround carry and adding to LSB '1'

→ Result

Ex: Subtract 1001.101 from 1100.001

Sol Step1: is complement of 1001.101 is 0110.010.

Step2: Adding

```
  1100.001
  0110.010      is complement of 1001.101
 [1]0010.011    Taking the end carry bit and
    ----->1     adding to LSB '1'.
```

Step3:

```
  0010.100      Result
```

Ex:

Subtract 101 from 1101

Sol The length of 1101 is 4 bits

The length of 101 is 3 bits.

101 is also write as 0101

Step1: is complement of 0101 is 1010

Step2: Adding 1101 and 1010

```
  1101
  1010      is complement of 0101
 [1]0111    Taking end around carry and
    ----->  adding to LSB '1'.
```

Step3:

```
  1000      Result.
```

Adding leading zeros for the number with less number of bits. so as to make the lengths of both the numbers same before taking Complement.

Subject: DECO
Faculty: A. Swathi
Topic: r's Complement

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 15

r's Complement:

The r's complement of an 'n' digit no 'N' in base 'r' is defined as $r^n - N$ for $N \neq 0$ and
0 for $N = 0$

* Comparing with the (r-1)'s complement, we note that the r's complement is obtained by adding '1' to (r-1)'s complement.

since, $\boxed{r^n - N = [(r^n - 1) - N] + 1}$

Ex: Find 10's complement for 2389.

Sol

step1: First find the 9's Complement of 2389

```
  9 9 9 9
- 2 3 8 9
---------
  7 6 1 0   9's complement of 2389
+       1   adding 1 to LSB
---------
  7 6 1 1   The Result
---------
```

step2:

step3:

Ex: The 2's complement of 101100 is,

step1: 1's complement of 101100
replace 0's by 1's and 1's by 0's ⇒ 010011

step2: Adding 1 to LSB

```
  010011
+      1
--------
  010100
--------
```

step3: we get the result.

Method ②

→ For 10's complement

The 10's complement of 246700 is

Step 1 : Leaving the two zeros unchanged

Step 2 : Subtracting 7 from 10 and subtracting other digits from 9.

```
  9 9 9 10
- 2 4 6 7  0 0
-----------------
  7 5 3 3  0 0
-----------------
```

Step 3 : The result is 753300.

→ For 2's complement

The 2's complement of 1101100 is

Step 1 : By leaving all least significant 0's and first '1' unchanged. And then replace all 1s by 0's and all 0s by 1's in all other higher, significant bits.

Step 2 :

```
1 1 0 1 1 0 0
      ↕ ↕ ↕
      1 0 0
0 0 1 0 1 0 0
```

Step 3 : The 2's complement of 1101100 is

0010100.

Subject: DECO
Faculty: A. Swathi
Topic: Complements

Class Notes

Unit No: I
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 16

9's complement method of subtraction

To perform decimal subtraction using the 9's complement method, ① obtain the 9's complement of subtrahend and ② add it to the minuend. Call this number the intermediate result.

If there is a carry, it indicates that the answer is positive

③ Add the carry to LSD of this result to get the answer. carry is called end around carry.

→ If there is no carry, it indicates that the answer is negative and the intermediate result is it's 9's complement Take the 9's complement of this result and place a negative sign in front to get the answer.

Ex: subtract 745.81 − 436.62 by using the 9's complement method.

Sol Step$_1$: Here Minuend is = 745.81 and
Subtrahend is = 436.62

obtain the 9's complement of subtrahend.

$$\begin{array}{r} 999.99 \\ -436.62 \\ \hline 563.37 \end{array}$$

Step$_2$: add it to the minuend.

745.81
+ 563.37 9's complement
[1]309.18
→ end around carry.

step③: Add end around carry to the LSD of this result

[1]309.18 Intermediate result
+ →1 End around carry
309.19 Result

The carry indicates that the result is positive.
So answer is **+309.19**

Ex:② 436.62 − 745.81 by using 9's complement method.

Sol Step$_1$: 9's complement of subtrahend is

999.99
745.81
254.18

Step$_2$: Add

436.62 Minuend
+ 254.18 9's complement of subtrahend
690.80

Step$_3$: There is no carry indicating that the result is negative. So take 9's complement of the intermediate result and put a minus sign

∴ The 9's complement of 690.80 is

999.99
− 690.80
309.19

∴ The answer is **−309.19**

Subject: DECO
Faculty: A. Swathi
Topic: Complements

Class Notes

Unit No: I
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 17

10's complement method of subtraction

To perform decimal subtraction using the 10's complement method,

1. obtain the 10's complement of the subtrahend and
2. add it to the minuend.
3. If there is a carry, ignore it.
   The presence of carry indicates that the answer is positive
   The result obtained is itself the answer.

→ If there is no carry, it indicates that the answer is negative and the result obtained is its 10's complement. obtain the 10's complement of the result and place a negative sign infront to get the answer.

Ex① Subtract 2928.54 − 416.73 using 10's complement method.

Sol Step1: obtain the 10's complement of subtrahend

```
  9999.99
  0416.73
  ------
  9583.26   → 9's complement
    +1      → 10's complement
```

Step2: add

```
  2928.54
  9583.27   10's complement of 416.73
  -------
 [1]2511.81   Ignore carry
```

Step3: There is a carry indicating that the answer is positive. Ignore the carry.

The answer is **2511.81**

Ex① Subtract 416.73 − 2928.54 by using 10's complement method.

Sol Step₁: obtain the 10's complement of subtrahend

$$\begin{array}{r} 9999.99 \\ -\ 2928.54 \\ \hline 7071.45 \quad \text{9's complement} \\ +\qquad\ 1 \\ \hline 7071.46 \quad \text{10's complement} \end{array}$$

Step₂: Add

$$\begin{array}{rl} 0416.73 & \text{Minuend} \\ +\ 7071.46 & \text{10's complement} \\ \hline 7488.19 & \end{array}$$

Step₃: There is no carry indicating that the result is negative.

So, take 10's complement of the intermediate result. and put a minus sign.

∴ The 10's complement of 7488.19 is

$$\begin{array}{r} 9999.99 \\ 7488.19 \\ \hline 2511.80 \\ +\qquad 1 \\ \hline 2511.81 \end{array}$$

∴ The answer is $\boxed{-2511.81}$

2's complement method of subtraction

1. obtain the 2's complement of the subtrahend
2. Add minuend and 2's complement of the subtrahend.
3. If there is carry, ignore it. result is positive. the result obtained is itself the answer.
4. If there is no carry, indicates that result is negative and result is obtained is 9's 10's complement.

Subject: DECO
Faculty: A. Swathi
Topic: Complements

Class Notes

Unit No: I
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 18

Subtraction by using 1's and 2's complement

→ Perform the subtraction with the following unsigned binary numbers a) by taking 2's complement of the subtrahend and b) by taking 1's complement of the subtrahend.

i) 11010 − 10000

Sol a) 1) 2's complement of 10000 is 10000

2) Add

11010		= 26
+ 10000	2's complement	= −16
101010	ignore the carry	+10

= +01010 Result is positive.

3) There is carry. Ignore it. The MSB is 0. Hence the answer is positive and is in true binary form

So it is **+01010 = +10**

Sol b) 1) 1's complement of 10000 is 01111

2) Add

11010	
+ 01111	1's complement
101001	end around carry

3) There is carry. It is added to LSB of intermediate result. The MSB is 0.

01001
→1
01010

Hence the result is positive and is in binary form

So it is **+1010 = +10**

ii) 1010100 – 1010100

Sol a) 2's complement method

1) obtain 2's complement of 1010100 is

0101011 1's complement

+ 1 Adding 1

1010100 2's complement

2)

3) Ignore the carry.

The MSB is 0.

Hence the answer is positive.

1010100

1010100

10000000

ii) 11010 – 1101

Sol 2's complement Method

i)

11010 = 26

+ 10011 2's complement = -13

101101 Ignore carry +13

= +01101 = +13.

There is carry. Ignore it. The MSB is 0. Hence the answer is positive and is in true binary form so it is **+01101 = +13**

1's complement Method

i)

11010 26

+ 10000 1s complement -13

110101 Add carry +13

01101 Result positive = +13.

So. it is **+1101 = +13**

Subject: DECO
Faculty: A. Swathi
Topic: Boolean Algebra

Class Notes

Unit No: I
Lecture No:
Date Conducted:

Boolean Algebra:

Boolean Algebra is the algebra of truth tables and operations performed on them.

* In the year 1938, Claude Elwood Shannon developed Boolean algebra for the analysis of two valued switching functions.
Shannon was the first to apply Boolean algebra to digital circuitry.
* Every circuit in a computer or any other electronic device can be designed by using the rules of boolean algebra.
* It is used in designing circuits which perform different tasks.
* Boolean algebra performs logical addition & multiplication. No subtraction and division operations are available here.
* There are also no negative or fractional numbers in boolean algebra.
* Therefore, boolean algebra reduces complex logic circuits into simpler ones.
* Binary '0' represents low voltage level while binary '1' represents high voltage level.
* Symbols '0' and '1' are used to represent open and close contacts respectively.
Hence two element boolean algebra also known as switching algebra can be constructed.

Boolean variable :

In algebra, we define a variable as the one that can take different values at different instants of time.

* A boolean variable is a variable that is capable of taking only two states or values. These states are represented by 1 or 0. (True or False)

Boolean Expression :

Boolean expressions are similar to algebraic expressions containing the relationship among several terms. These terms will be have boolean variables and logical operators.

Truthtable :

A truthtable is the pictorial representation of a boolean expression containing boolean variables for showing results of all possible combinations of input.

* The input boolean variables will have one of two states '0' or '1', True or False.
* The combinations of all such input variable states will result into an output value to be zero or one for that Boolean expression.
* The study of truthtable is very useful in simplification of boolean expressions and in designing of electronic circuit.

Subject: DECO
Faculty: A. Swathi
Topic: properties of Boolean algebra

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 20

Laws of Boolean Algebra: properties

Axioms or postulates of Boolean algebra are a set of logical expressions that we accept without proof and upon which we can build a set of useful theorems.

AND operation

$0 \cdot 0 = 0$ $\quad 0 \cdot 1 = 0$ $\quad 1 \cdot 0 = 0$ $\quad 1 \cdot 1 = 1$

OR operation

$0 + 0 = 0$ $\quad 0 + 1 = 1$ $\quad 1 + 0 = \phi$ $\quad 1 + 1 = 1$

NOT operation

$\overline{1} = 0$ $\quad \overline{0} = 1$

Complementation Laws:

Law1: If $A = 0$, then $\overline{A} = 1$

Law2: If $A = 1$, then $\overline{A} = 0$

Law3: $\overline{\overline{A}} = A$

Commutative Laws:

Law1: $A + B = B + A$

Law2: $A \cdot B = B \cdot A$

Associative Laws:

Law1: $(A + B) + C = A + (B + C)$

Law2: $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

Distributive laws :

Law$_1$: $A \cdot (B+C) = AB + AC$

Law$_2$: $A + BC = (A+B)(A+C)$

Law$_3$: $A \cdot \overline{A}B = A+B$

Idempotent laws :

Law$_1$: $A \cdot A = A$

Law$_2$: $A + A = A$

Identity laws :

Law$_1$: $A \cdot 1 = A$

Law$_2$: $A + 1 = 1$

Null laws :

Law$_1$: $A \cdot 0 = 0$

Law$_2$: $A + 0 = A$

Absorption laws :

Law$_1$: $A + A \cdot B = A$

Law$_2$: $A(A+B) = A$

Law$_3$: $A(\overline{A}+B) = AB$

Law$_4$: $A \cdot B + \overline{B} = A + \overline{B}$

Law$_5$: $A \cdot \overline{B} + B = A + B$

Demorgan's laws :

Law$_1$: $\overline{A+B} = \overline{A}\,\overline{B}$

Law$_2$: $\overline{A \cdot B} = \overline{A} + \overline{B}$

Subject: DECO
Faculty: A. Swathi
Topic: Basic theorems of Boolean algebra

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 21

Basic Theorems of Boolean algebra

1. Consensus Theorem (Included factor Theorem)

Theorem 1 :

$$AB + \bar{A}C + BC = AB + \bar{A}C$$

- proof :

$$L.H.S = AB + \bar{A}C + BC$$
$$= AB + \bar{A}C + BC(A + \bar{A}) \quad [\because A + \bar{A} = 1]$$
$$= AB + \bar{A}C + ABC + \bar{A}BC \quad \text{Identity law}$$
$$= AB(1 + C) + \bar{A}C(1 + B) \quad [\because 1 + C = 1]$$
$$= AB + \bar{A}C \quad [1 + B = 1]$$
$$= R.H.S$$

∴ Hence proved

* This theorem can be extended to any number of variables.

$$AB + \bar{A}C + BCD = AB + \bar{A}C$$
$$L.H.S = AB + \bar{A}C + BCD$$
$$= AB + \bar{A}C + BC + BCD$$
$$= AB + \bar{A}C + BC(1 + D)$$
$$= AB + \bar{A}C + BC$$
$$= AB + \bar{A}C$$
$$= R.H.S$$

∴ $AB + \bar{A}C + BCD = AB + \bar{A}C$

Theorem 2 :

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

proof :

$L.H.S = (A+B)(\bar{A}+C)(B+C)$

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

$= A\bar{A}B + ABC + \bar{A}BB + BBC + A\bar{A}C + ACC + \bar{A}BC + BCC$

$[\because A\cdot\bar{A} = 0 \quad BB = B \quad CC = C]$

$= 0 + ABC + \bar{A}B + BC + 0 + AC + \bar{A}BC + BC$

$= ABC + \bar{A}B + BC + AC + \bar{A}BC \quad [\because BC + BC = BC]$

$= BC(A+\bar{A}) + BC + \bar{A}B + AC$

$= BC + BC + \bar{A}B + AC \quad [\because A + \bar{A} = 1]$

$= BC + \bar{A}B + AC \quad [\because BC + BC = BC]$

$R.H.S = (A+B)(\bar{A}+C)$

$= A\bar{A} + AC + \bar{A}B + BC$

$= 0 + AC + \bar{A}B + BC$

$= BC + \bar{A}B + AC$

$\therefore L.H.S = R.H.S$

This theorem can be extended to any no. of variables

For example,

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

Transposition Theorem :

Theorem :

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

proof :

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

$= A\bar{A} + AB + \bar{A}C + BC$

$= 0 + AB + \bar{A}C + BC(A+\bar{A})$ [$\because A\bar{A} = 0$, $A+\bar{A} = 1$]

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

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

$= AB(1+C) + \bar{A}C(1+B)$ [$\because 1+C = 1$, $1+B = 1$]

$= AB + \bar{A}C$ Identity Law

$= L.H.S$

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

Duality :

In Boolean Algebra we can produce dual by changing all "+" signs to "." signs
all "." signs to "+" signs and compl complementing all 0's and 1's.

The variables are not complemented.

Simplifying the following Boolean Expressions.

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

Sol $A\bar{B}C + B + B\bar{D} + AB\bar{D} + \bar{A}C$

$\Rightarrow A\bar{B}C + B(1 + \bar{D} + A\bar{D}) + \bar{A}C$ $\quad [\because 1 + \bar{D} + A\bar{D} = 1]$

$1 + \bar{D} + A\bar{D}$
$1 + \bar{D}(1 + A)$
$1 + \bar{D} = 1$

$\Rightarrow A\bar{B}C + B + \bar{A}C$

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

$\Rightarrow C(\bar{A} + A)(\bar{A} + B) + B$ $\quad [\because A\bar{B} + \bar{A} = \bar{A} + \bar{B}$ Distributive law$]$

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

$\Rightarrow \bar{A}C + \bar{B}C + B$

$\bar{A}\bar{A} + \bar{A}\bar{B} + A\bar{A} + AB$
$\bar{A} + \bar{A}\bar{B} + 0 + AB$
$\bar{A}(1 + \bar{B}) + AB$
$\bar{A} + A\bar{B} = \bar{A} + \bar{B}$

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

$\Rightarrow B + C + C\bar{A}$

$\Rightarrow B + C(1 + \bar{A})$

$\Rightarrow B + C$

Reduce the expression $AB + A\bar{B}C + B\bar{C}$

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

$= A(B + \bar{B})(B + C) + B\bar{C}$ $\quad [B + \bar{B}C = (B + \bar{B})(B + C)]$

$= AB + AC + B\bar{C}$

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

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

$= AC(1 + B) + B\bar{C}(1 + A)$

$= AC + B\bar{C}$

Subject: DECO
Faculty: A. Swathi
Topic: Logic Gates

Class Notes

Unit No: I
Lecture No:
Date Conducted:

Logic Gates:

Logic gates are the fundamental blocks of digital system.

* It consists of several inputs and only one input.
* The output level (logic HIGH or logic LOW) depends upon the combinations of input levels.
* There are just three basic types of gates -
    1. AND Gate
    2. OR Gate
    3. NOT Gate
* The interconnection of gates to perform a variety of logic operations is called "logic design".
* THE UNIVERSAL GATES

Though logic circuits of any complexity can be realized by using only the three basic gates, there are two universal gates

1. NAND Gate
2. NOR Gate

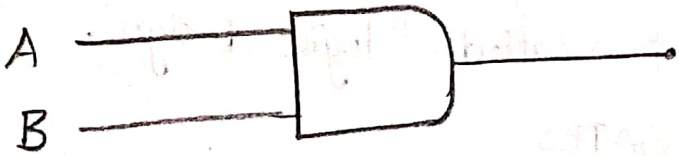
- each of which can also realize logic circuits single handedly.
- The NAND and NOR gates are called Universal Building blocks.
- Both NAND and NOR gates perform all the 3 basic logic functions (AND, OR and NOT).

The AND Gate:

An AND gate has two or more inputs but only one output.

* The output assumes the logic 1 state, only when each one of its inputs at logic 1 state.
* The output assumes the logic 0 state even if one of its inputs is at logic 0 state.
* The AND Gate is defined as a device whose output is 1, if and only if all its inputs are 1. Hence, the AND gate is also called as all or nothing gate.
* AND logic gate is denoted by the symbol . (dot)

The logic diagram for AND gate is,



Logic symbol of n input AND gate

Input		output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Truthtable of AND gate

The OR Gate:

An OR gate has two or more inputs but only one output.

* The output assumes the logic 1 state, even if one of its inputs is in logic 1 state.
* Its output assumes the logic 0 state, only when each of its inputs is in logic 0 state.
* An OR gate can be defined as a device whose output is 1, even if one of its inputs is 1. Hence an OR gate is also called an any or all gate.
* It can also be called an inclusive OR gate.
* Logic symbol for the OR operation is '+'.

Logic symbol of OR gate

The truth table for OR gate is,

Input		output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

The NOT Gate : (Inverter)

A NOT gate, also called an inverter, has only one input and only one output.

* It is a device whose output is always the complement of its input.
  i.e, the output of a NOT gate assumes the logic 1 state when its input is in logic 0 state and the logic 0 state when its input is in logic 1 state.
* The logic symbol of Inverter is,

Logic symbol of NOT gate

* Truthtable for NOT gate is,

Input	output
A	Y
0	1
1	0

Truthtable of NOT gate

* The symbol for NOT operation is '—' bar.
* When the input variable to the NOT gate is represented by A and the output variable by X, the output expression for the output is $X = \bar{A}$.
  This is read as X is equal to A bar.
* A descrete NOT gate may be realized using a transistor.

Subject: DECO
Faculty: A. Swathi
Topic: Universal Gates

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 25

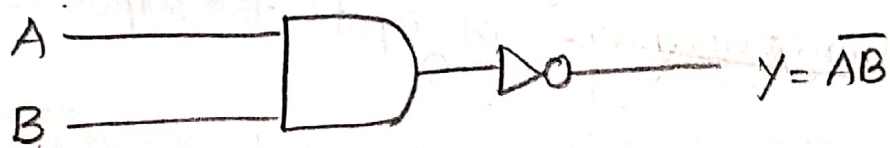
The NAND Gate :

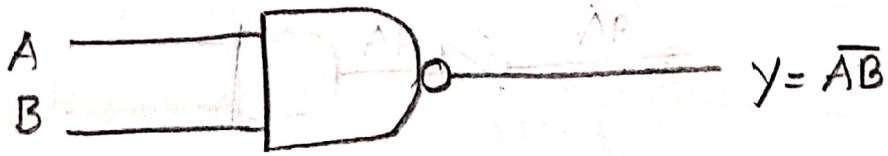
NAND means NOT AND, i.e the AND output is NOTed. so, a NAND gate is a combination of an AND gate and a NOT gate.

* NAND is a contraction of the word NOT-AND.

NAND operation = AND operation + NOT operation

* The output is logic 0 level, only when each of the inputs assumes a logic 1 level.
* NAND indicated by the graphic symbol, which consists of an AND graphic symbol followed by a small circle.
* Logic diagram for NAND gate is,





* Truth table for NAND gate is,

Inputs		output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NOR Gate :

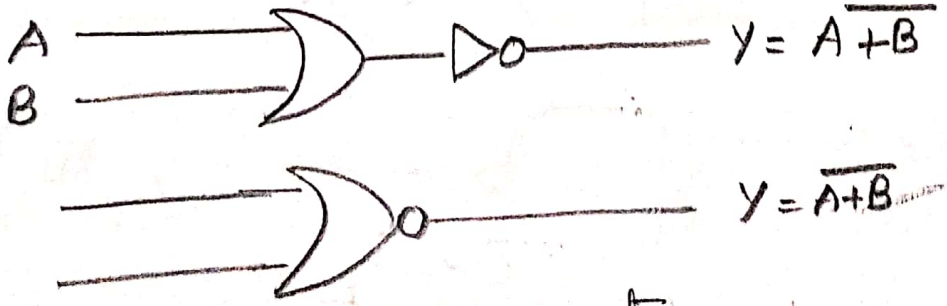
NOR means NOT OR, i.e the OR output is NOTed.

* So, NOR gate is a combination of an OR gate and a NOT gate.
* Infact NOT is a contraction of the word NOT-OR.

NOR gate = OR Gate + NOT gate

* A NOR gate can also be used as an inverter, by typing all its input terminals together and applying the signal to be converted to the common terminal
* NOR gate uses an OR graphic symbol followed by a small circle.
* The output is logic 1 level, only when each one of its inputs assumes a logic 0 level.

For other combinations of inputs, the output is logic 0 level.



Logic symbol of NOR gate

Truthtable for NOR gate is,

Inputs		output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Subject: DECO
Faculty: A. Swathi
Topic: Exclusive-OR gate

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 26

Exclusive-OR Gate:

The exclusive-OR gate has a graphic symbol similar to the OR gate except for the additional curved line on the input side.

* The x-OR gate has two inputs but one output.
* The output is logic high when the inputs are not equal.
* The output is logic low when the inputs are equal to either Logic HIGH or Logic LOW.
* x-OR gate is also called as inequality detector or anti-coincidence gate.
* x-OR gate is a logic gate that performs modulo addition of its inputs.
* x-OR operator is denoted by the symbol '⊕'.

Logic symbol of 2-input x-OR gate

Inputs		output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Truth table of x-OR gate

Exclusive-NOR Gate (X-NOR)

· XNOR is a logic gate that performs the negation of X-OR operation.

XNOR operator = X-OR operator + NOT operator

* The output of the X-NOR gate is logic 1 (i.e high) when both the inputs have the same values.
  i.e either both are high or both are low
* The output of X-NOR logic gate is low (i.e logic 0) if any one of the inputs is either low(0) or high(1)
* X-NOR is also called as equality detector.
* X-NOR is indicated by an X-OR graphic symbol followed by a small circle.
* X-NOR operator is denoted by the symbol '⊙'.

Logic symbol

$Y = A \odot B$

$Y = \overline{A \oplus B}$

$Y = \bar{A}\bar{B} + AB$

Logic Symbol of X-NOR gate

Inputs A	Inputs B	output Y
0	0	1
0	1	0
1	0	0
1	1	1

Truthtable of X-NOR gate.

Subject: DECO
Faculty: A. Swathi
Topic: Canonical form

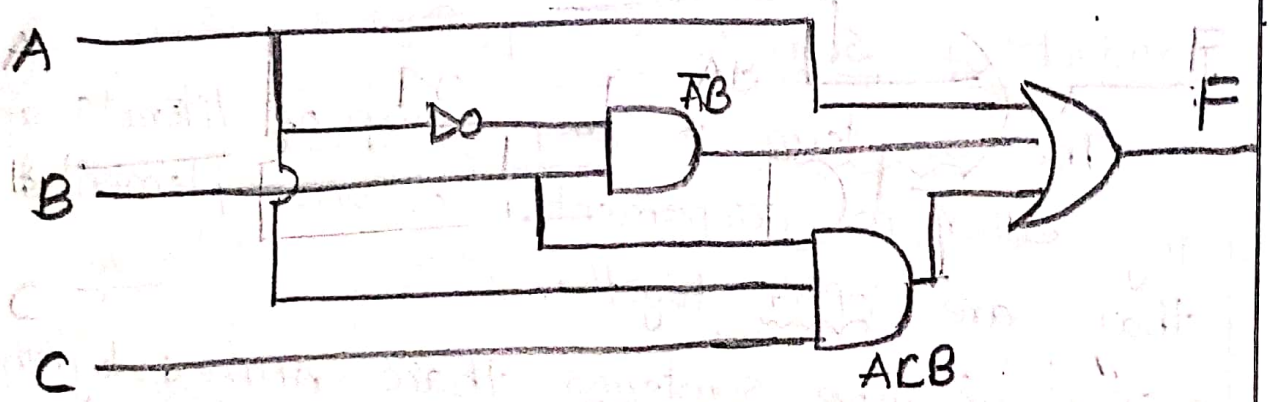
Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 27

Canonical and Standard Forms :

$F = A + \bar{A}B + ACB$, in this A, B, C are known as Literals.

* A Literal is complemented or uncomplemented variable
* To complement any Boolean expression using logic gates the literals are designated as inputs to the logic gate.
* The implementation of $F = A + \bar{A}B + ACB$ using logic gates as,



* To implement a logic function with less number of gates we have to minimize literals and the number of terms.
Usually, literals and terms are arranged in one of the two standard forms.

1. Sum of products form
2. product of sum form.

Sum of products form (SOP):

The product term in any group of literals appearing either in complemented or uncomplemented form, that are ANDed together.

* Two or more product terms are ORed together, to form a sum of products expression.

Ex:

1. $F = A\bar{B}C + \bar{B}C\bar{D} + \bar{A}CD + A\bar{B}\bar{D}$
2. $F = A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{D} + ABC + BCD$
3. $F = AB + \bar{A}\bar{C}B + A\bar{C} + B\bar{D}$

where AB, $\bar{A}CB$, $A\bar{C}$, $B\bar{D}$ are product terms.

Product of Sum form (POS):

The sum term is any group of literals appearing either in complemented or uncomplemented form that are ORed together.

* Two or more sumterms thare ANDed together to form a product of sums expression.

Ex:

1. $F = (A+\bar{B}+\bar{C})\ (\bar{A}+B+\bar{C}+\bar{D})\ (\bar{C}+\bar{D})$
2. $F = (B+C)\ (\bar{A}+\bar{C}+D)\ (A+\bar{B}+D)$
3. $F = (\bar{A}+B)\ (A+B+C)\ (\bar{C}+B)\ (\bar{B})$

where, $\bar{A}+B$, $A+B+C$, $\bar{C}+B$, $\bar{B}$ are Sum terms.

Subject: DECO
Faculty: A. Swathi
Topic: Canonical forms

Class Notes

Unit No: I
Lecture No:
Date Conducted:

Canonical sum of products:

A sum of products expression is referred to as Canonical sum of products or standard sum of products expression if every product term involves every literal or its complement.

Ex:

1. $F = \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}BC\bar{D}$
2. $F = \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}CD + A\bar{B}\bar{C}D$
3. $F = \bar{A}BC + \bar{A}\bar{B}C + AC\bar{B} + ABC$ where $\bar{A}BC, \bar{A}\bar{B}C, AC\bar{B}, ABC$ are product terms involves all the literals A, B and C.

Canonical product of Sum:

The product of sum expression is referred to as canonical product of sums or standard product of sums expression. If every sum term involves every literal or its complement.

Ex:

1. $F = (\bar{A}+\bar{B}+\bar{C}+\bar{D})(\bar{A}+B+C+\bar{D})(A+B+\bar{C}+D)$
2. $F = (A+\bar{B}+\bar{C}+\bar{D})(\bar{A}+B+\bar{C}+\bar{D})(\bar{A}+B+C+D)$
3. $F = (A+\bar{B}+C+D)(\bar{A}+B+\bar{C}+\bar{D})(A+\bar{B}+\bar{C}+D)(A+B+C+D)$

Where, $A+\bar{B}+C+D$, $\bar{A}+B+\bar{C}+\bar{D}$, $A+\bar{B}+\bar{C}+D$, $A+B+C+D$ are sum terms involving all literals A, B, C and D.

Minterm:

A product term which contains each of 'n' variables as factors in either complemented or uncomplemented form is called a Minterm.

* It is represented by 'm'.

Maxterm:

A sumterm which contains each of 'n' variables as factors in either complemented or uncomplemented form is called a Maxterm.

* It is represented by 'M'.

Conversion from pos to standard pos:

1. Examine each term
2. In eachterm, check the variables that donot occur, then add the "variables and to Complement".

   Ex: In 'x+y', "z" is not present

   Add $z\bar{z}$ to term and multiply it

   $\Rightarrow (x+y) + z\bar{z}$

   $\Rightarrow x+y+z\bar{z}$

   $\Rightarrow (x+y+z)\ (x+y+\bar{z})$.

3. Eliminate redundant terms.

Subject: DECO
Faculty: A. Swathi
Topic: Conversions

Class Notes

Unit No: I
Lecture No:
Date Conducted:

Convert pos into canonical pos. form

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

Sol Given that,

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

We know that

$A\bar{A} = 0$, $B\bar{B} = 0$, $C\bar{C} = 0$

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

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

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

2. $f = (\bar{A}+\bar{B})(A+\bar{B})(A+B)$

Sol Given that,

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

We know that

$A\bar{A} = 0$, $B\bar{B} = 0$, $C\bar{C} = 0$

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

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

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

Conversion from SOP to standard SOP form:

To convert SOP to canonical SOP form the following steps are required.

1. Examine each term, if it is a minterm, retain it and continue to the next term.
2. In each product which is not a minterm, check the variables that do not occur, for each x_i that does not occur, multiply by product by $(x_i + \bar{x}_i)$. For ex, for a 3-variable function (x, y, z) if a term contains only x, y which is not a minterm multiply xy with $(z+\bar{z}) = xyz + xy\bar{z}$.
3. Multiply out all products and eliminate redundant terms.

Minimization of Switching functions:

A switching function can usually be represented by a no. of expressions.

* our intension is to develop a procedure for obtaining minimal expression.
* we did simplifications, by using algebraic manipulations. even for 4 or 5 variables this method is ineffective.
* The presented map method is very effective for hand simplifications of expressions upto 5 or 6 variables, while the tabulation procedure is suitable for machine computation and yields minimal expressions.

Subject: DECO
Faculty: A. Swathi
Topic: Sop into canonical sop form

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 30

→ Convert sop into canonical sop-form:

1. $f = \bar{A}B + \bar{C} + ABC$.

Sol) We know that $A+\bar{A}=1$; $B+\bar{B}=1$; $C+\bar{C}=1$

In $\bar{A}B$ term 'c' is not present,
In $\bar{C}$ term A, B are not present
so, add these literals.

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

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

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

Eliminate redundant term $\bar{A}B\bar{C}$.

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

2. $f = A\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}$.

Sol) We know that $A+\bar{A}=1$, $B+\bar{B}=1$, $C+\bar{C}=1$

In $A\bar{C}$ term, B literal is not present
In $\bar{A}\bar{B}$ term, C literal is not present
So, add missing literals

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

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

$f = AB\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C$.

Difference between sop and pos

Sum of product	product of sum
1. A way of representing boolean expression as sum of product terms	1. A way of representing boolean expressions as product of sum terms
2. sop uses minterms. Minterm is a product of boolean variables either in normal form or complemented form.	2. pos uses maxterms. Maxterm is a sum of boolean variables either in normal form or complemented form.
3. It is sum of minterms Minterms are represented as 'm'	3. It is product of maxterms Maxterms are represented as 'M'.
4. sop is formed by considering all the minterms, whose output is HIGH(1) while writing minterms for sop,	4. pos is formed by considering all the maxterms, whose output is Low(0).
5. while writing minterms for sop, input with value 1 is considered as the variable itself and input with value 0 considered as complement of the input.	5. While writing maxterms for pos, input value 1 is considered as the complement and input with value 0 is considered as the variable itself.

Karnaugh Mapping:

Karnaugh Mapping is a method of plotting a boolean function and is used to simplify the expression making use of a truth table.

* To draw a karnaugh map, a Boolean function must be written in a sum of products form or in the Minterm form
* If it is not in this form, then we should use the truth table to convert it to the sum of the products form.

How to plot a karnaugh Mapping:

A karnaugh mapping is a diagram consists of squares. Each square of the map respesents a minterm. i.e a term in the sum of the product form.

* Any logic expression can be written as a sum of products, i.e sum of minterms.

Therefore, a logic expression can easily be represented on a karnaugh map.

A karnaugh map for n variables is made up of 2^n squares.

Each square designates a product term of a Boolean Expression.

For product terms which are present in the Expression 1's are written in the corresponding squares.

0's are written in those squares which correspond to product terms not present in the expression. For clarity of the map, writing of 0's can be omitted.
So, blank squares indicate that they contain 0's.

TWO-Variable K-map :

A two-variable expression can have $2^2 = 4$ possible combinations of the input variables A and B.

- Each of these combinations, $\bar{A}\bar{B}$, $\bar{A}B$, $A\bar{B}$ and AB (in the sop form) is called a minterm.
- Instead of these, representing the minterms in terms of input variables, using the notation the minterms may be represented in terms of their decimal designations.

m_0 for $\bar{A}\bar{B}$
m_1 for $\bar{A}B$
m_2 for $A\bar{B}$
m_3 for AB

Assuming that A represents the MSB.
The letter 'm' stands for minterm and the subscript represents the decimal designation of minterm.

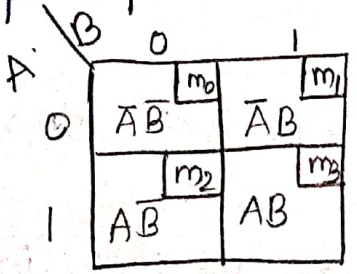


Fig: The minterms of a two variable K-map.

Subject: DECO
Faculty: A. Swathi
Topic: Two - variable K-map

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 32

- Each sum term in the standard pos expression is called a maxterm.

A function in two variables (A, B) has $2^2 = 4$ possible maxterms, $A+B$, $A+\overline{B}$, $\overline{A}+B$ and $\overline{A}+\overline{B}$

They are represented

M_0 for $A+B$
M_1 for $A+\overline{B}$
M_2 for $\overline{A}+B$
M_3 for $\overline{A}+\overline{B}$.

The upper case letter M stands for maxterm and its subscript denotes the decimal designation of that maxterm.

- For mapping a pos expression on to the K-map, 0's are placed in the squares corresponding to the maxterms which are present in the expression.
- 1's are placed (or no entries are made) in the squares corresponding to the maxterms which are not present in the expression.
- The decimal designation of the squares for maxterms is the same as that for the minterms.

A \ B	0	1
0	m_0 $A+B$	m_1 $A+\overline{B}$
1	m_2 $\overline{A}+B$	m_3 $\overline{A}+\overline{B}$

Maxterms of a 2-variable K-map.

Ex: Reduce the expression $f = \bar{A}\bar{B} + \bar{A}B + AB$ using mapping

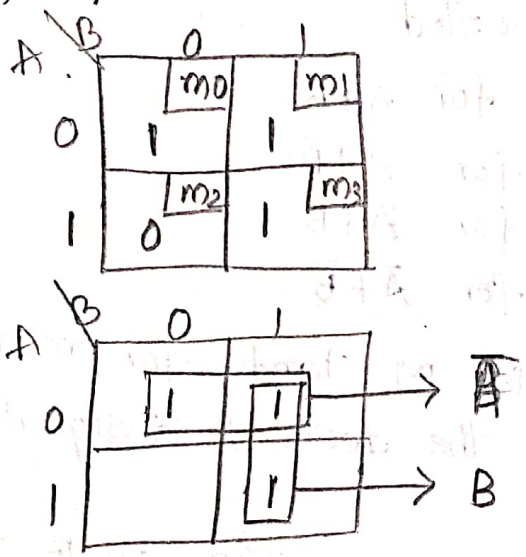
Sol: The Given expression is

$$f = \bar{A}\bar{B} + \bar{A}B + AB$$

Expressed in terms of minterms is

$$f = \Sigma m(0, 1, 3)$$

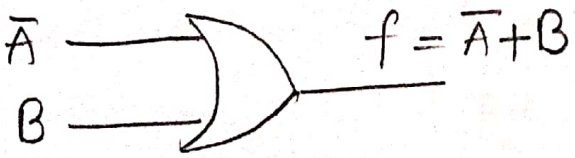
K-map for this function is



In one square, A is constant as a '0'. but B varies from a '0' to a '1', and in the other 2-square, B is constant as a '1' but A varies from a '0' to a '1'.

So, the required expression is $\bar{A} + B$.

It requires two gate inputs for realization.



Logic diagram for $f = \bar{A} + B$

Subject: OECO
Faculty: A. Swathi
Topic: 2-variable K-map

Class Notes

Unit No: I
Lecture No:
Date Conducted:

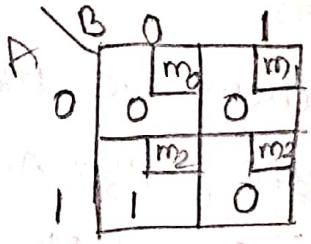
→ Reduce the expression $f = (A+B)(A+\bar{B})(\bar{A}+\bar{B})$ using mapping

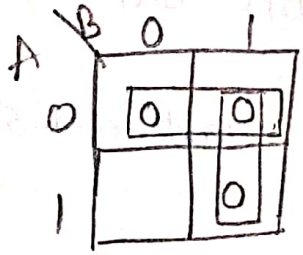
Sol: Given expression is $f = (A+B)(A+\bar{B})(\bar{A}+\bar{B})$

Expressed interms of maxterms is

$$f = \Pi M(0,1,3)$$

The K-map for f and its reduction is.





In the given expression, the maxterm M_2 is absent. This is indicated by a '1' on the K-map.

So, the required Expression is $A\bar{B}$.

The corresponding Sop expression is Σm_2 or $A\bar{B}$.

This realization is the same as that of the pos form.

It requires two gate inputs, for re

Three-Variable K-map:

A function in three variables (A, B, C) expressed in the standard sop form have $2^3 = 8$ possible combinations $\bar{A}\bar{B}\bar{C}$, $\bar{A}\bar{B}C$, $\bar{A}B\bar{C}$, $\bar{A}BC$, $A\bar{B}\bar{C}$, $A\bar{B}C$, $AB\bar{C}$ and ABC. Each of these combinations designated by $m_0, m_1, m_2, m_3, m_4, m_5, m_6$ and m_7 respectively, is called a minterm.

- A is the MSB of the minterm designator and C is the LSB.

In the standard pos form, the 8 possible combinations are $M_0 = A+B+C$, $M_1 = A+B+\bar{C}$, $M_2 = A+\bar{B}+C$, $M_3 = A+\bar{B}+\bar{C}$, $M_4 = \bar{A}+B+C$, $M_5 = \bar{A}+B+\bar{C}$, $M_6 = \bar{A}+\bar{B}+C$ and $M_7 = \bar{A}+\bar{B}+\bar{C}$. is called a maxterm.

- A is the MSB of the maxterm designator and C is the LSB.

- A 3-variable K-map, has $8 = 2^3$ squares or cells, each square on the map represents a minterm or maxterm. A small number on the top right corner of each cell indicates the minterm or maxterm designation.

A \ BC	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$ 0	$\bar{A}\bar{B}C$ 1	$\bar{A}BC$ 3	$\bar{A}B\bar{C}$ 2
1	$A\bar{B}\bar{C}$ 4	$A\bar{B}C$ 5	ABC 7	$AB\bar{C}$ 6

Minterms

A \ BC	00	01	11	10
0	$A+B+C$ 0	$A+B+\bar{C}$ 1	$A+\bar{B}+\bar{C}$ 3	$A+\bar{B}+C$ 2
1	$\bar{A}+B+C$ 4	$\bar{A}+B+\bar{C}$ 5	$\bar{A}+\bar{B}+\bar{C}$ 7	$\bar{A}+\bar{B}+C$ 6

Maxterms

Subject: DECO
Faculty: A. Swathi
Topic: Three-variable K-map

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 34

Ex: Map the expression

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

Sol Given expression is.

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

In the given expression, the minterms are

$\bar{A}\bar{B}C = 001 = m_1$
$A\bar{B}C = 101 = m_5$
$\bar{A}B\bar{C} = 010 = m_2$
$AB\bar{C} = 110 = m_6$
$ABC = 111 = m_7$

So, the expression $f = \Sigma m(1,2,5,6,7)$

The corresponding K-map is.

A \ BC	00	01	11	10
0	0 (m0)	1 (m1)	0 (m3)	1 (m2)
1	0 (m4)	1 (m5)	1 (m7)	1 (m6)

A \ BC	00	01	11	10
0		1		1
1		1	1	1

K-map in SOP form.

→ Map the expression.

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

Sol The Given expression is,

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

Maxterms are,

$A+B+C = 000 = M_0$

$\bar{A}+B+\bar{C} = 101 = M_5$

$\bar{A}+\bar{B}+\bar{C} = 111 = M_7$

$A+\bar{B}+\bar{C} = 011 = M_3$

$\bar{A}+\bar{B}+C = 110 = M_6$

So, the expression $f = \Pi M(0,3,5,6,7)$

The mapping of the expression is,

A \ BC	00	01	11	10
0	M_0	M_1	M_3	M_2
1	M_4	M_5	M_7	M_6

A \ BC	00	01	11	10
0	0		0	
1		0	0	0

K-map in pos form

Subject: DECO
Faculty: A. Swathi
Topic: Four-variable K-map

Class Notes

Unit No: I
Lecture No:
Date Conducted:

Four Variable K-map :

A four variable (A, B, C, D) expression can have $2^4 = 16$ possible combinations of input variables. such as $A\bar{B}\bar{C}\bar{D}$, $\bar{A}\bar{B}\bar{C}D$, --- --- $ABCD$ with minterm designations m_0, m_1 --- m_{15} respectively, in the sop form and $A+B+C+D$, $A+B+C+\bar{D}$, --- --- $\bar{A}+\bar{B}+\bar{C}+D$ with maxterm designations M_0, M_1, --- M_{15} respectively, in the pos form.

- A 4-variable K-map has $2^4 = 16$ squares or cells and each square on the map represents either a minterm or a maxterm.

AB \ CD	00	01	11	10
00	$\bar{A}\bar{B}\bar{C}\bar{D}$ (m0) 0	$\bar{A}\bar{B}\bar{C}D$ (m1) 1	$\bar{A}\bar{B}CD$ (m3) 3	$\bar{A}\bar{B}C\bar{D}$ (m2) 2
01	$\bar{A}B\bar{C}\bar{D}$ (m4) 4	$\bar{A}B\bar{C}D$ (m5) 5	$\bar{A}BCD$ (m7) 7	$\bar{A}BC\bar{D}$ (m6) 6
11	$AB\bar{C}\bar{D}$ (m12) 12	$AB\bar{C}D$ (m13) 13	$ABCD$ (m15) 15	$ABC\bar{D}$ (m14) 14
10	$A\bar{B}\bar{C}\bar{D}$ (m8) 8	$A\bar{B}\bar{C}D$ (m9) 9	$A\bar{B}CD$ (m11) 11	$A\bar{B}C\bar{D}$ (m10) 10

Minterms of a 4-variable K-map.

The binary number designations of the rows and columns are in the Gray code.

Here 11 follows 01 and 01 and 10 follows 11. This is called adjacency ordering.

- The binary numbers along the top of the map indicate the conditions of C and D along any column and binary numbers along the left side indicate the conditions of A and B along any row.
- The numbers in the top right corners of the squares indicate the minterm or maxterm designations as usual.

AB \ CD	00	01	11	10
00	A+B+C+D	A+B+C+D	A+B+C+D	A+B+C+D
01	A+B+C+D	A+B+C+D	A+B+C+D	A+B+C+D
11	A+B+C+D	A+B+C+D	A+B+C+D	A+B+C+D
10	A+B+C+D	A+B+C+D	A+B+C+D	A+B+C+D

maxterms of a four-variable K-map.

- squares which are physically adjacent to each other or which can be made adjacent by wrapping the map around from left to right or top to bottom can be combined to form bigger squares.
- The bigger squares (2 squares, 4 squares, 8 squares, etc.) must from either a geometric square or rectangle.

Subject: DECO
Faculty: A. Swathi
Topic: K-map

Class Notes

Unit No: I
Lecture No:
Date Conducted:
Page No: 36

→ Using K-map, find the minimal sum of products and product of sums that are equivalent to

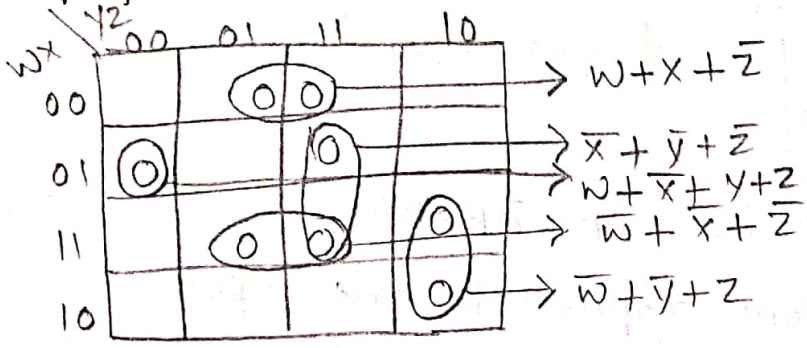
$E(w,x,y,z) = \Pi M(1,3,4,7,10,13,14,15)$

Sol Given that,

$E(w,x,y,z) = \Pi M(1,3,4,7,10,13,14,15)$

Given function is in pos-form (product of sum)

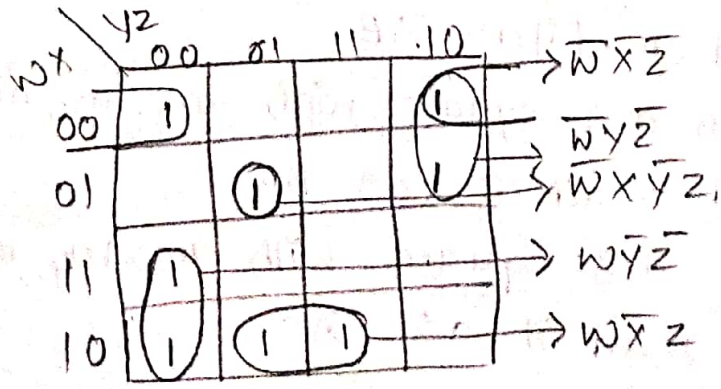
K-map for this function is,



$f = (w+x+\bar{z})(w+\bar{x}+y+z)(\bar{x}+\bar{y}+\bar{z})(\bar{w}+\bar{x}+\bar{z})(\bar{w}+\bar{y}+z)$

Sop form is:

$E(w,x,y,z) = \Sigma m(0,2,5,6,8,9,11,12)$



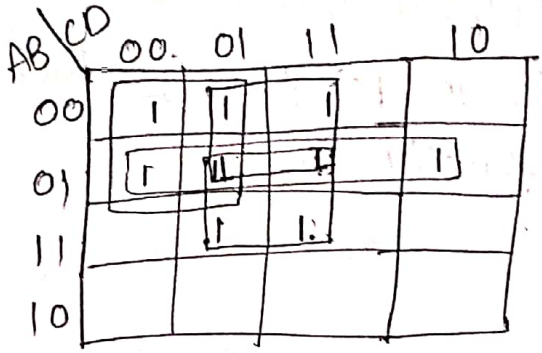
$\therefore f = \bar{w}\bar{x}\bar{z} + \bar{w}x\bar{y}z + \bar{w}y\bar{z} + w\bar{y}\bar{z} + w\bar{x}z$

Reduce using mapping the expression $f = \Pi M(2, 8, 9, 10, 11, 12, 14)$ and implement the real minimal expression in Universal logic.

Sol The given expression is in the form of sop

$$f = \Sigma m(0, 1, 3, 4, 5, 6, 7, 13, 15)$$

k-map for sop form is,



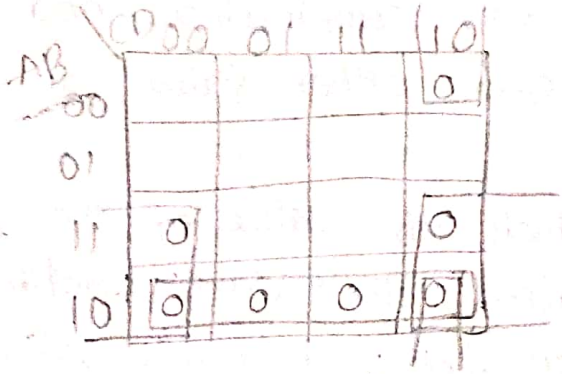
$$f_{min} = \bar{A}\bar{C} + \bar{A}D + \bar{A}B + BD$$

f_{min} sop form requires 12 gate inputs.

1. There are no isolated 1's
2. There are no 1's which can be Combined only into 2-squares. So make no 2-squares.
3. m_6 can form a 4-square with m_4, m_5, m_7. Make it and read it as $\bar{A}B$
4. m_{15} can form a 4-square with m_5, m_7, m_{13}. Make it and read it as BD.
5. m_0 Can form a 4-square with m_1, m_4, m_5. Make it and read it as $\bar{A}\bar{C}$.
6. Only m_3 is left. It can make a 4-square with m_1, m_5, m_7. Make it and read it as $\bar{A}D$.
7. Write all the product terms in sop form.

So, the minimal sop expression is

$f_{min} = \bar{A}B + BD + \bar{A}\bar{C} + \bar{A}D$.



pos k-map

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

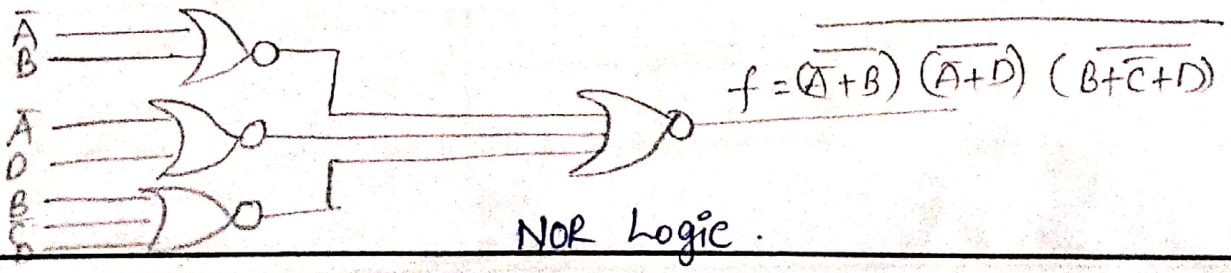
In the pos k-map,

1. There are no isolated 0's.
2. M_2 can form a 2-square with only M_{10}.
   Make it and read it as $(B+\bar{C}+D)$
3. M_{12} and M_{14} can form a 4-square with M_8 M_{10}
   Make it and read it as $(\bar{A}+D)$
4. M_9 and M_{11} can form a 4-square with M_8 M_{10}
   Make it and read it as $(\bar{A}+B)$
5. Write all the sum terms in pos form

The implementation of the minimal expression using NOR gates is

$f_{min} = \overline{(\bar{A}+B)(\bar{A}+D)(B+\bar{C}+D)}$

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



NOR Logic.

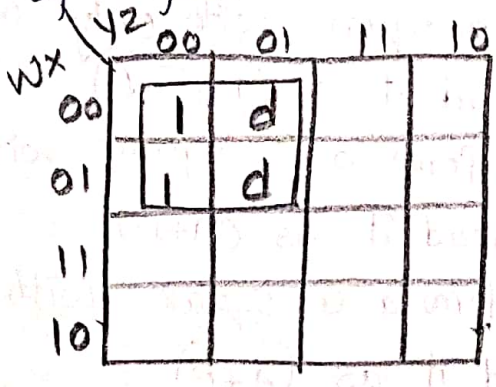
Don't Care Combinations :

So far the functions considered have been completely specified for every combination of the variables. There exist situations, however, where, while a function is to assume a value 1 for some combinations and the value '0' for others, it may assume either value for a number of combinations.

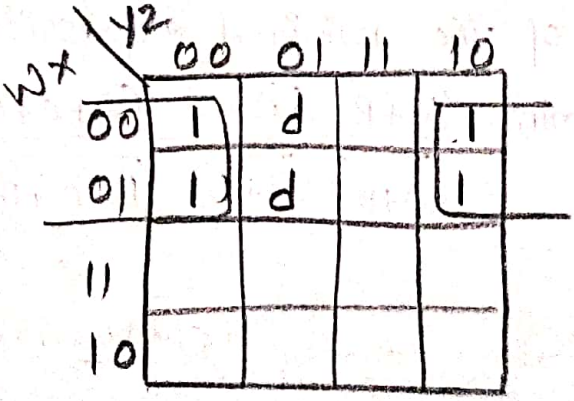
* Combinations for which the value of the function is not specified are called don't care combinations.
* The value of function for such combinations is denoted by a 'ϕ' or 'd'.

For example,

The d's can be used as 1's to form a subcube consisting of four 1's.



→ whereas the d's need not to used



Subject: DECO
Faculty: A.Swathi
Topic: K-map

Class Notes

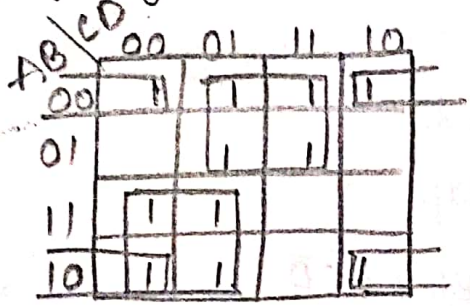
Unit No: I
Lecture No:
Date Conducted:
Page No: 38

Reduce using mapping the expression $f = \Sigma m(0,1,2,3,5,7,8,9,10,12,13)$ and implement the real minimal expression in universal logic.

Sol The given expression function is

$f = \Sigma m(0,1,2,3,5,7,8,9,10,12,13)$

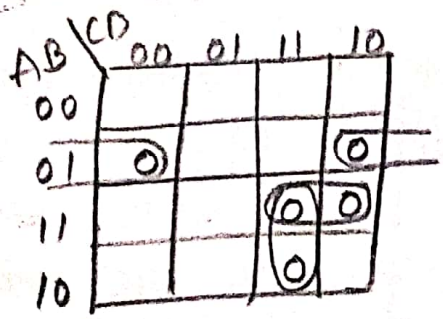
K-map for given function is (Sop K-map)



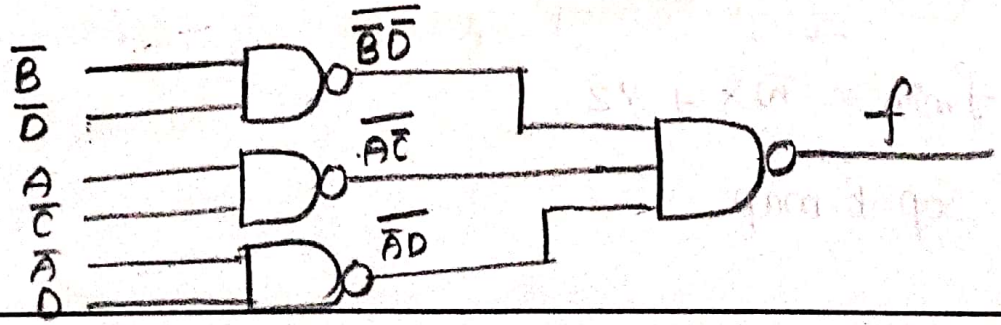
$f_{min} = \bar{A}D + A\bar{C} + \bar{B}\bar{D} = \overline{\bar{A}D + A\bar{C} + \bar{B}\bar{D}}$

K-map for pos form is,

$f = \Pi M(4,6,11,14,15)$



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

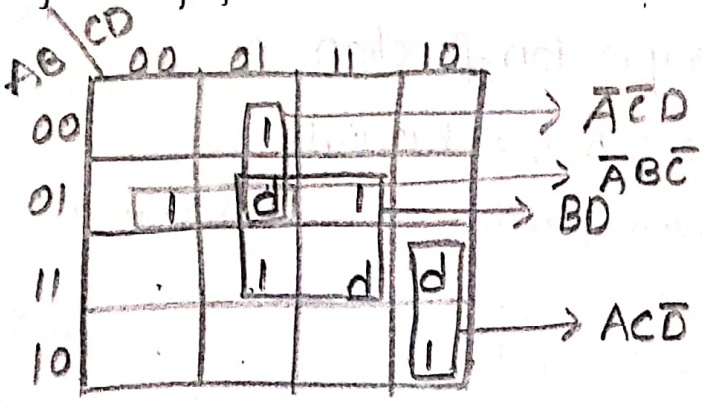


Minimize the following expressions using K-map:

1) $F(A,B,C,D) = \Sigma m(1,4,7,10,13) + \Sigma d(5,14,15)$.

Sol The given expressions are in SOP form.

K-map for SOP form is,



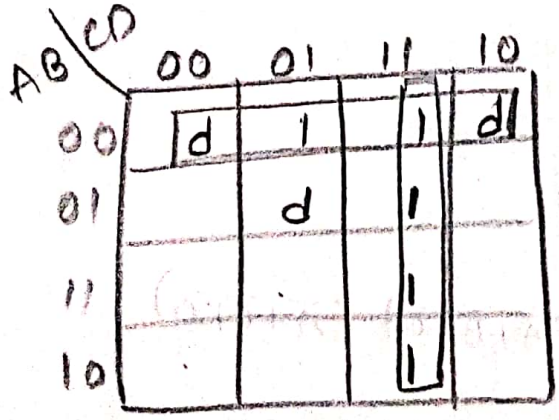
(K-map) SOP K-map

$F_{min} = \bar{A}\bar{C}D + \bar{A}B\bar{C} + BD + AC\bar{D}$

2) $F(W,X,Y,Z) = \Sigma m(1,3,7,11,15) + \Sigma d(0,2,5)$

Sol The given expressions are in SOP form

K-map for this SOP form is,



$f_{min} = \bar{W}\bar{X} + YZ$

SOP K-map

Subject: DECO
Faculty: A. Swathi
Topic: Five-variable K-map

Class Notes

Unit No: I
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:
Page No: 39

Five-variable K-map :

A five-variable (A, B, C, D, E) expression can have $2^5 = 32$ possible combinations of input variables such as $\bar{A}\bar{B}\bar{C}\bar{D}\bar{E}$, $\bar{A}\bar{B}\bar{C}\bar{D}E$, ---, ABCDE, with minterm designations $m_0, m_1, \ldots, m_{31}$ respectively, in SOP form.

- $A+B+C+D+E$, $A+B+C+D+\bar{E}$, ---- $\bar{A}+\bar{B}+\bar{C}+\bar{D}+\bar{E}$, with maxterms designations $M_0, M_1, \ldots M_{31}$ respectively, in POS form.
- The 32 squares of the K-map are divided into 2 blocks of 16 squares each.
- Left block represents minterms from m_0 to m_{15} in which A is a '0' and right block represents minterms from m_{16} to m_{31} in which A is 1.
- The five-variable K-map may contain 2 squares, 4-squares, 8-squares, 16-squares or 32-square involving these two blocks.
- squares are also considered adjacent in these two blocks, if when superimposing one block on top of another, the squares coinside with one another.

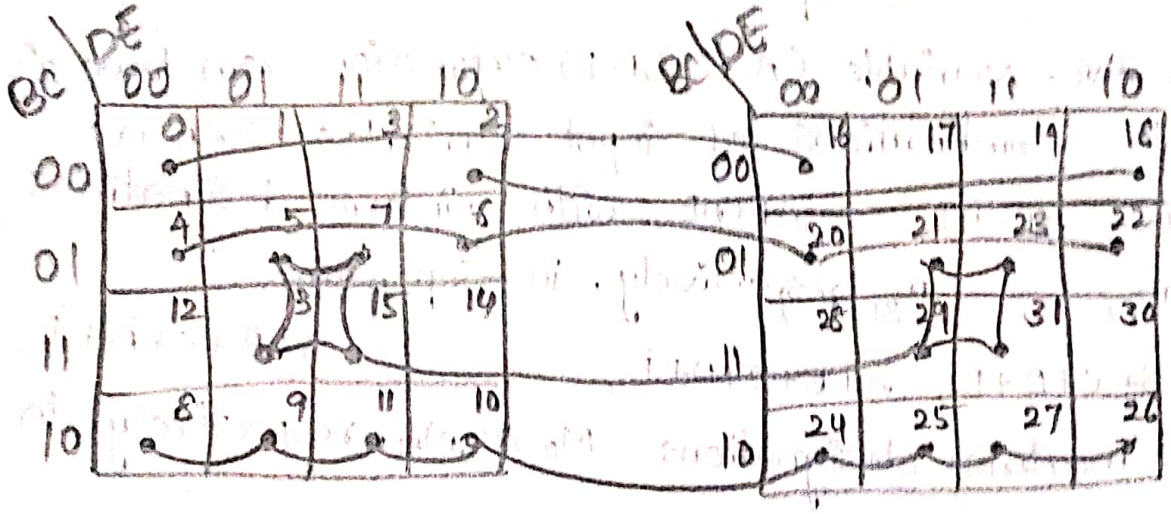
a) $m_0, m_{16} = \bar{B}\bar{C}\bar{D}\bar{E}$

b) $m_2, m_{18} = \bar{B}\bar{C}D\bar{E}$

c) $m_4, m_6, m_{20}, m_{22} = \bar{B}C\bar{E}$

d) $m_5, m_7, m_{13}, m_{15}, m_{21}, m_{23}, m_{29}, m_{31} = CE$

e) $m_8, m_9, m_{10}, m_{11}, m_{24}, m_{25}, m_{26}, m_{27} = B\bar{C}$



Some possible groupings in a five variable K-map.

Ex:

Reduce the following expression in sop and pos forms using mapping:

$f = \Sigma m(0,2,3,10,11,12,13,16,17,18,19,20,21,26,27)$.

Sol The given expression in pos form is

$f = \Pi M(1,4,5,6,7,8,9,14,15,22,23,24,25,28,29,30,31)$

The real minimal expression is the minimal of the sop and pos forms.

In the sop, form

1. There are no isolated 1's.
2. m_{12} can go only with m_{13}, Form a square which is read as $\bar{A}BC\bar{D}$.

Subject:
Faculty:
Topic:

Class Notes

Unit No:
Lecture No:
Link to Session
Planner (SP): S.No.... of SP
Book Reference:
Date Conducted:

3) m_0 can go with m_2, m_{16} and m_{18}. so. form 4-square which is read as $\bar{B}\bar{C}\bar{E}$

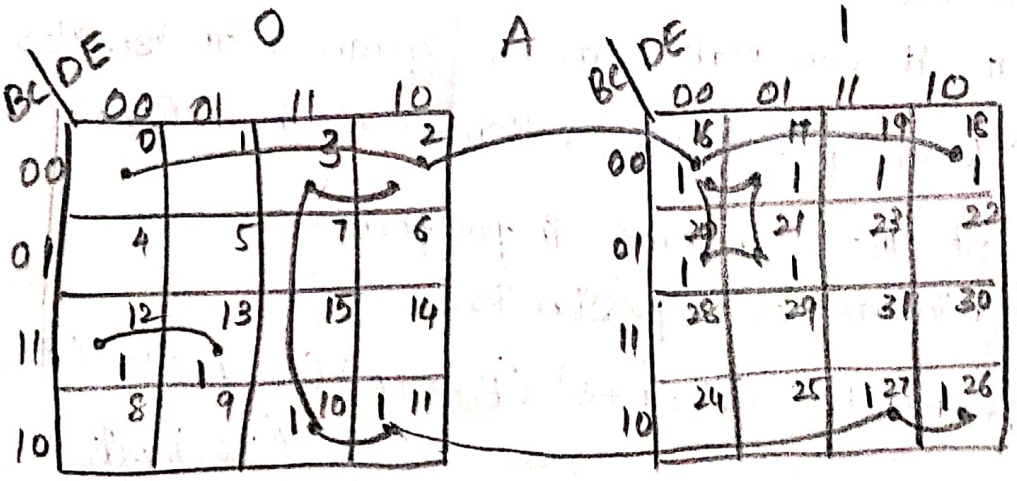
4) m_{20}, m_{21}, m_{17} and m_{16} form a square which is read as $A\bar{B}\bar{D}$

5) $m_2, m_3, m_{18}, m_{19}, m_{10}, m_{11}, m_{26}$ and m_{27} form an 8-square which is read as $\bar{C}D$.

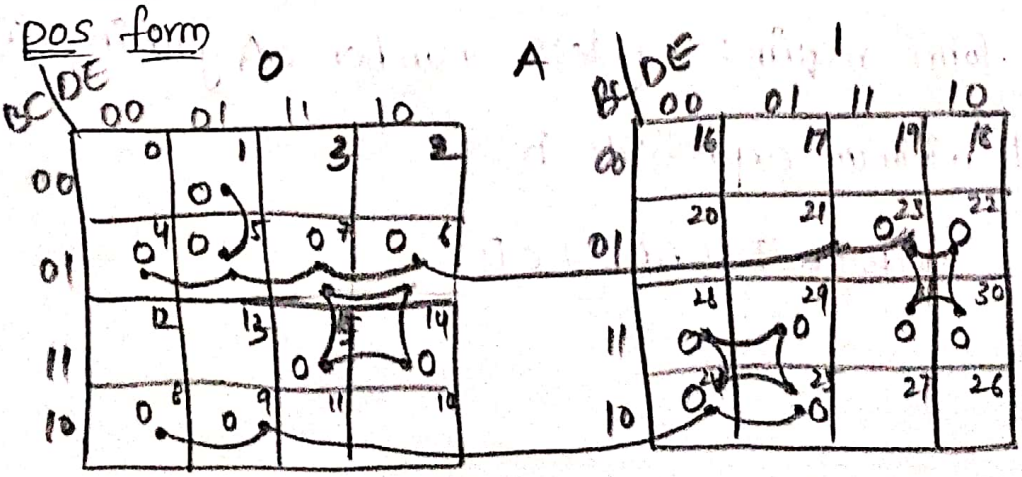
6) write all the productterms in sopform.

So, the minimal Sop expression is

$f_{min} = \bar{A}BC\bar{D} + \bar{B}\bar{C}\bar{E} + A\bar{B}\bar{D} + \bar{C}D$ (16 inputs).



POS form



In the pos K-map, the reduction is done as per these steps.

1. There are no isolated 0's.
2. M_1 can go only with M_5 or M_9. Make a 2-square with M_5, which is read as $(A+B+C+D+E)$
3. M_4 can go with M_5, M_7 and M_6 to form a 4-square, which is read as $(A+B+\bar{C})$.
4. M_8 can go with M_9, M_{24} and M_{25} to form a 4-square, which is read as $(\bar{A}+\bar{B}+D)$ (or) $(\bar{B}+C+D)$
5. M_{28} can go with M_{29}, M_{24} and M_{25} to form a 4s square which is read as $(\bar{A}+\bar{B}+D)$.
6. M_{30} can make a 4-square with M_{31}, M_{29} and M_{28} or with M_{31}, M_{14} and M_{15} or with M_{30}, M_{22} and M_{23}. Don't do that.
   Note that it can make an 8-square with M_{31}, M_{23}, M_{22}, M_6, M_7, M_{14} and M_{15}, which is read as $(\bar{C}+\bar{D})$.
7. Write all the sumterms in pos form.
   So. the minimal pos expression is

$$F_{min} = (A+B+D+\bar{E})(A+B+\bar{C})(\bar{B}+C+D)(\bar{A}+\bar{B}+D)(\bar{C}+\bar{D})$$

(20 inputs)

The sop form requires a less number of gate inputs.

The real minimum expression is.

$$f_{min} = \bar{A}BC\bar{D} + \bar{B}\bar{C}\bar{E} + A\bar{B}\bar{D} + \bar{C}D.$$