MODULE - 3

Syllabus

Combinational Logic - 90

Combinational Circuits and design Procedure - 91

Binary adder and Subtractor - 92

Multilevel NAND and NOR circuits - 100

Exclusive - OR and Equivalence Functions.

Implementation of Combination Logic

Parallel adder

Carry look ahead adder

BCD adder

Code Converter

Magnitude Comparator

Decoder

Multiplexer

Demultiplexer

Parity Generator

I COMBINATIONAL CIRCUIT

Combinational circuits consists of logic gates whose outputs at any time are determined directly from the present combination of inputs without regard to previous inputs.

Sequential circuits employ memory elements (binary cells) in addition to logic gates. Their outputs are a function of the input and the state of the memory elements

Combinational circuit consists of input variables, logic gates and output variables. The logic gates accept signals from the inputs and generate signals to the outputs.

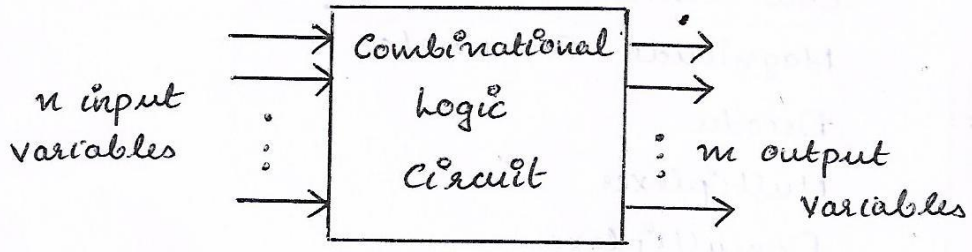


fig Block diagram of a combinational circuit

The n input binary variables come from an external source, the m output variables go to an external destination. For n input variables there are 2^n possible combination of binary input values. For each possible input combination there is one and only one possible output combination.

II DESIGN PROCEDURE

Design of combinational circuits start from the verbal outline of the problem and ends in a logic circuit diagram.

The procedure involves the following steps:

1. The problem is stated.
2. The number of available input variables and required output variables is determined.
3. The input and output variables are assigned letter symbols.
4. The truth table that defines the required relationship between inputs and output is derived
5. The simplified Boolean function for each output is obtained.
6. The logic diagram is drawn.

A truth table for a combinational circuit consists of input columns and output columns. The 1's & 0's in the input column are obtained from the 2^n binary combinations available for n input variables. The binary values for the outputs are determined from examination of the stated problem.

Some input combinations will not occur. These combinations become don't care. The output function specified in the truth table give the exact definition of the combinational circuit.

Constraints to be considered during design procedure are

1) Minimum number of gates
2) Minimum number of inputs to a gate.
3) Minimum propagation time of the signal through the circuit.

4) Minimum number of interconnections
5) Limitations of the driving capabilities of each gate.

III BINARY ADDER

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 10$ Sum = 0 Carry = 1

A combinational circuit that performs the addition of two bits is called a half-adder.
A combinational circuit that performs the addition of three bits is called a full-adder.

i) Half Adder

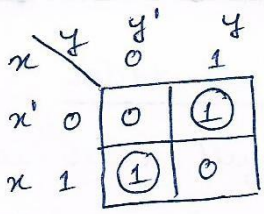
* Consist of two binary inputs (Augend and Addend) and two binary outputs (sum and carry)
* Two inputs x, y and two outputs S (for sum) and C (for carry)

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

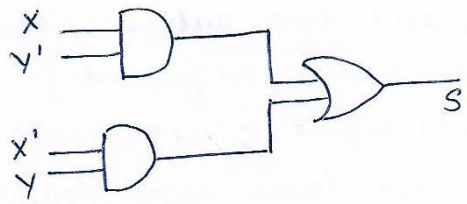
The simplified sum of products expressions for the output are:

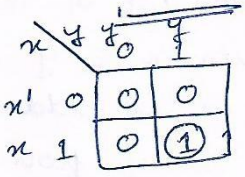
$S = x'y + xy'$

$C = xy$ SOP



$S = x'y + xy'$



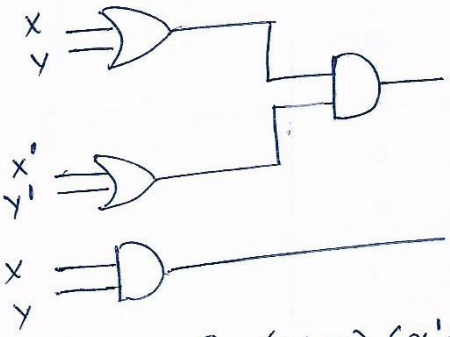


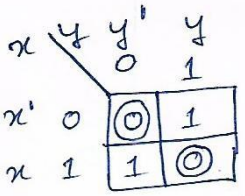
$C = xy$

$S = xy' + x'y$

$C = xy$

POS



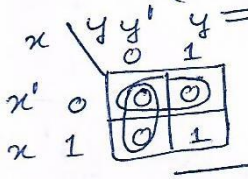


$S = x'y' + xy$

$= (x'+y')(x+y)$

$S = (x+y)(x'+y')$

$C = xy$

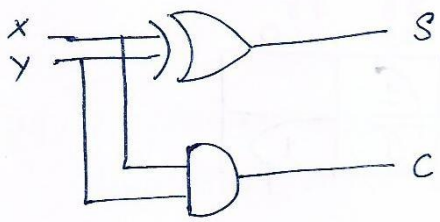


$C = y' + x'$

$= xy$

$x'y + xy' = x \oplus y$

So S is the exclusive OR of x & y



$S = x \oplus y$

$C = xy$

fig. Various implementation of a half adder.

ii) Full Adder

A full adder is a combinational circuit that forms the arithmetic sum of three input bits. It consist of three inputs and two outputs. Two inputs x & y represent the two significant bits to be added. The third input, z represents the carry from the previous lower significant position

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

x \ yz	y'z' (00)	y'z (01)	yz (11)	yz' (10)
x' (0)	0	(1)	0	(1)
x (1)	(1)	0	(1)	0

SOP

$$S = xy'z' + x'y'z + xyz + x'yz'$$

x \ yz	y'z' (00)	y'z (01)	yz (11)	yz' (10)
x' (0)			1	
x (1)		1	1	1

$$c = yz + xz + xy$$

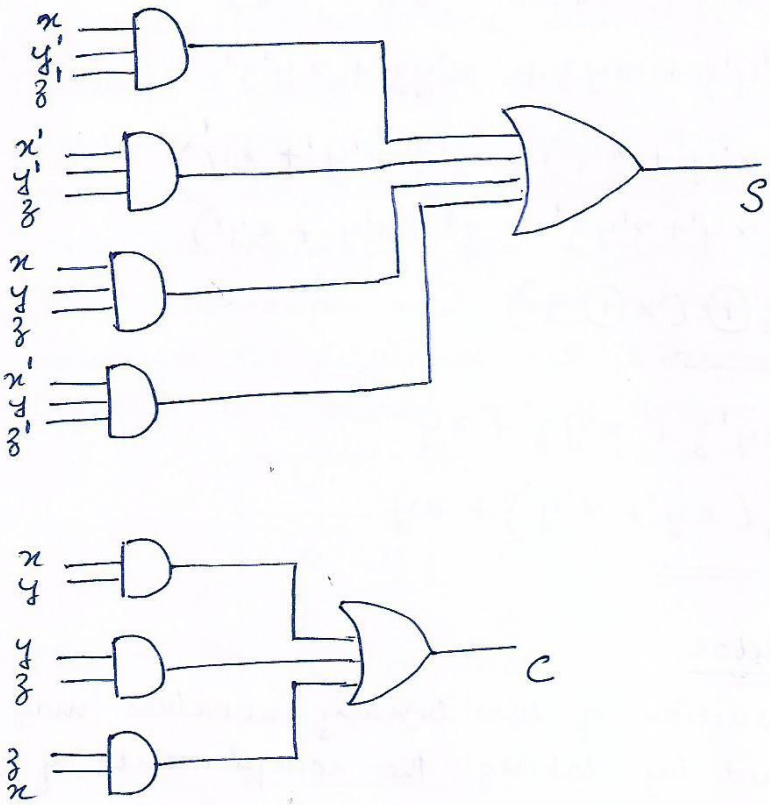


fig. Implementation of full-adder in sum of products.

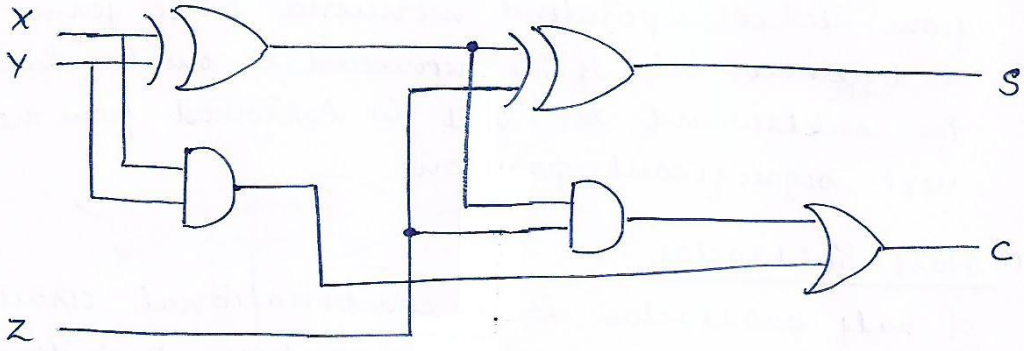


fig. Implementation of full adder with two half adder and an OR gate

$$S = x'y'z + x'yz' + xy'z' + xyz$$
$$= x'y'z + xyz + x'yz' + xy'z'$$
$$= z(x'y' + xy) + z'(x'y + xy')$$
$$= z(xy' + x'y)' + z'(x'y + xy')$$
$$= z \oplus (x \oplus y)$$

$$C = xy'z + x'yz + xy$$
$$= z(xy' + x'y) + xy$$

IV SUBTRACTOR

The subtraction of two binary numbers may be accomplished by taking the complement of the Subtrahend and adding it to the minuend Subtraction can be implemented using logic circuits in a direct manner also. But in this method each subtrahend bit of the number is subtracted from its correspondend minuend bit to form a difference bit. If the minuend is smaller than the subtrahend bit, a 1 is borrowed from the next significant position.

i) Half Subtractor

A half subtractor is a combinational circuit that subtracts two bits and produces their difference It also has an output to specify if a 1 has been borrowed. Minuend bit is designated by x and subtrahend bit by y. If $x > y$ we have three possibilities

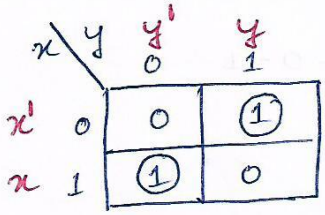
$$0 - 0 = 0$$
$$1 - 0 = 1$$
$$1 - 1 = 0$$

The result is called difference bit.

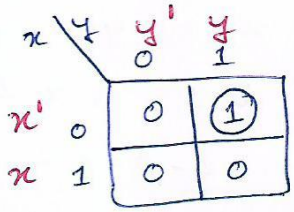
If $x < y$, we have 0-1 and it is necessary to borrow a 1 from the next higher stage. The one borrowed from the next higher stage adds 2 to the minuend bit, (10 in decimal system). With the minuend equal to 2, the difference becomes 2-1 = 1

Half-subtractor needs two outputs. One output generates the difference and is designated by the symbol D & second output borrow designated by B

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0



$D = xy' + x'y$



$B = x'y$

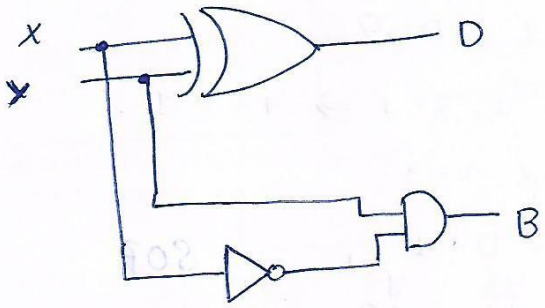


fig Half Subtractor

ii) Full Subtractor

A full subtractor is a combinational circuit that performs a subtraction between two bits taking into account that a 1 may have been borrowed by a lower significant stage. This circuit has three input and two outputs.

The three inputs x, y and z denote the minuend, subtrahend and previous borrow. The two outputs D and B represent the difference and output borrow respectively

The eight rows under the input variables designate all possible combinations of 1's and 0's that the binary variables may take. The 1's and 0's for the output variables are determined from the subtraction of $x-y-z$.

① $x=0, y=0, z=1 \Rightarrow 0-0-1$

$B=1$, & x becomes 2

$2-0-1 = D=1$

② $x=0, y=1, z=1 \Rightarrow 0-1-1$

$B=1$, & x becomes 2

$2-1-1 = D=0$

③ $x=1, y=1, z=1 \Rightarrow 1-1-1$

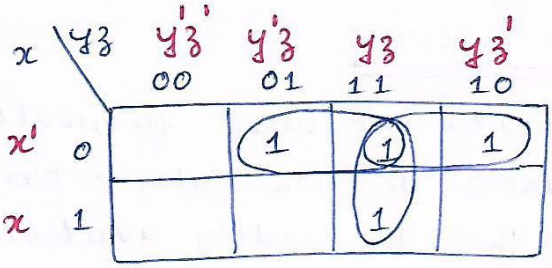
$B=1$ and $x=3$

$3-1-1 = D=1$

SOP

x \ yz	$y'z'$ 00	$y'z$ 01	yz 11	yz' 10
x' 0		1		1
x 1	1		1	

$$D = xy'z' + x'y'z + xyz + x'yz'$$



$B = yz + x'z + x'y$

x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

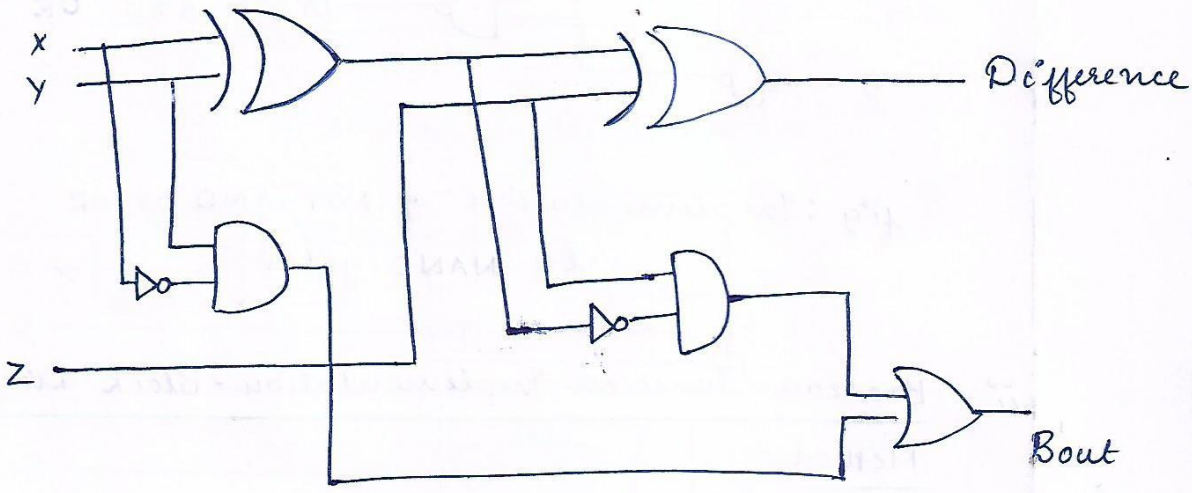


fig. Implementation of a full subtractor with two half subtractor