The symbol used for the OR gate is shown in Figure 13

Figure 13

As we can see, in figure 13, there are two inputs, shown by two arrows to the left and labelled by x and y, and its output is shown by an arrow to its right and labelled by the Boolean expression $x+y$. The input-output table corresponding to the OR gate is the following:

OR gate	Input x	Input y	Output x+y
	1	1	1
	1	0	1
	0	1	1
	0	0	0

In Circuitry theory, NOT, AND and OR gates are the basic gates. As we will see, we can design any circuit using these gates. The circuits that we will design depends only on the inputs not on the output. In other words these circuits have no memory. These circuits are called combinatorial circuits or logic circuits.

We extend these symbolic representation from basic Boolean expressions to arbitrary Boolean expressions.

Consider the Boolean expression $x'y$, $x'+y$, $x'y'$, $x'+y'$

The circuits for the expressions $x'y$ and xy' are shown in Figure 14(a) and 14(b) respectively.

Figure 14

The circuits for the expressions $x'+y$ and $x+y'$ are shown in Figure 15(a) and 15(b) respectively

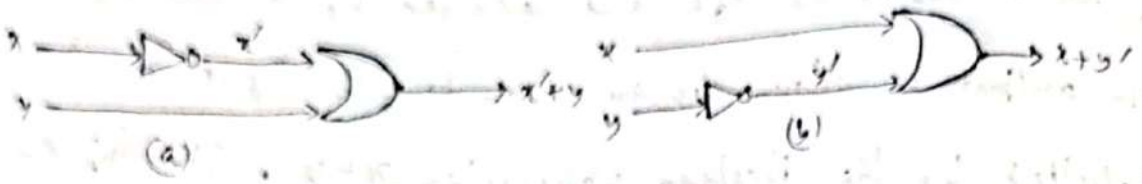


Figure 15

The circuit for the expression $x'y'$ is shown in Figure 16

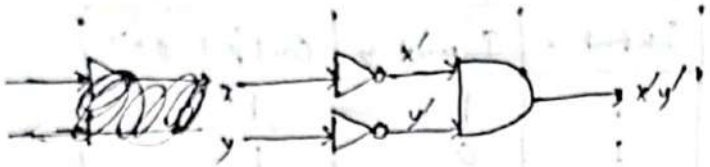
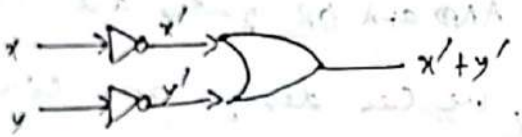


Figure 16

The circuit for $x'+y'$ is shown in Figure 17.



Understanding the the construction of the above circuits, we may now construct a logic circuit for the Boolean expression $xy+yz'$. To do this we proceed as follows: First we feed the inputs x and y through an AND gate. We also negate the input z and feed the output z' and the input y through an AND gate. Finally we take the outputs of the two AND gates and feed them through and OR gate to get the required logic circuit. The circuit is

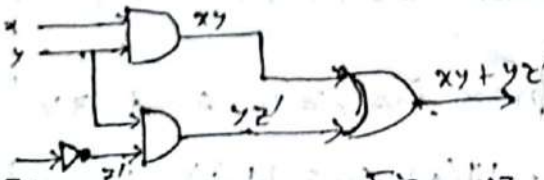


Figure 17

Note: The Boolean expression $xy+yz'$ can be simplified using distributive law as $y(x+z')$. Hence a simpler equivalent logic circuit is shown in figure 18.

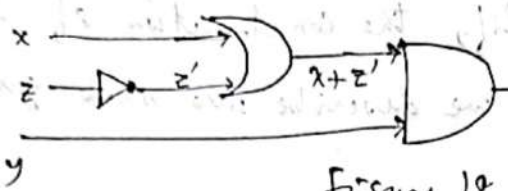


figure 18.

To find the logic circuit for the Boolean expression $xyz+xyz'+xy'z+x'yz$, we first simplify it.

$xyz+xyz'+xy'z+x'yz$

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

$= xy+xy'z+x'yz$ because $z+z'=1$

$= xy+xyz+xy'z+x'yz$ because $xy+xyz = xy(1+z) = xy.1 = xy$

$= xy+xz(y+y')+x'yz$

$= xy+xz+xyz+x'yz$ because $xz+xyz = xz(1+y) = xz.1 = xz$ and $y+y'=1$

$= xy+xz+(x+x')yz$

$= xy+xz+yz$ (because $x+x'=1$)

$= x(y+z)+yz$ (by distributive law)

Hence, the circuit corresponding to the expression $xyz+xyz'+xy'z+x'yz$ is equivalent to the circuit given by the expression $x(y+z)+yz$. Thus the required circuit is shown in figure 19.

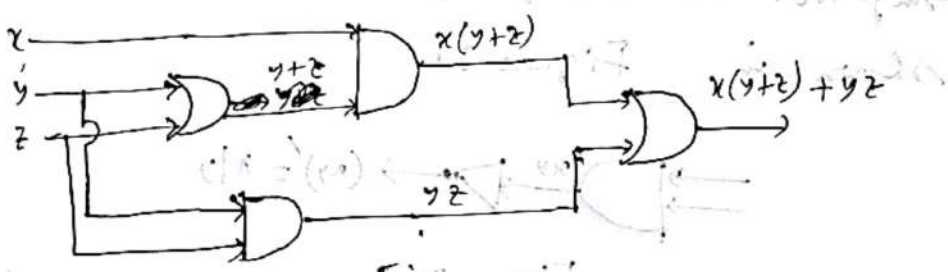


figure 19

We have presented various examples illustrating the design of circuits using the gates NOT, AND and OR. However there are other gates available that can be used to simply simplify the construction of a combinatorial circuit. Next we describe two such gates called NAND and NOR.

The symbol used for the Boolean expression $(xy)'$ is called a NAND gate.

The diagram in Figure 20 represents the NAND gate.

Figure 20

The input-output table of a NAND gate is as follows:

NAND gate	input x	input y	output of NAND gate
	1	1	0
	1	0	1
	0	1	1
	0	0	1

In Boolean algebra, the expression $(xy)'$ is also denoted by $x|y$. The binary operation symbol $|$ used in $x|y$ is called Scheffer stroke.

If we draw the circuit for the Boolean expression $(xy)'$ using NOT and AND gates, we obtain the circuit as shown in Figure 21.

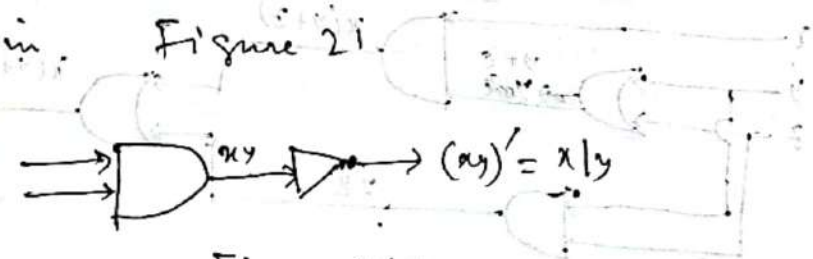


Figure 21