Digital Techniques

Logic Gates

Experiment (1)

Asst. Lect. Leal Abdullah Hamza

leal.abdullah@mustaqbal-college.edu.iq

2021-2022

Logic Gates

Object:

To perform the functions of the gates

Theory:

A logical gate is an electronic device that performs a Boolean operation on one or more inputs to produce an output. There are four types of names that have been used for circuits, logic circuits and gates. Binary logic deals with variables that take on two discrete values and with operations that assume logical meaning. The two values the variables take may be called by different names (e.g. True and False, Yes and No, 1 and 0).

Logic Gates:

1) AND Gate: the AND gate is a circuit which gives a high output (logic1) if all its inputs are high. A dot (.) is used to indicate the AND operation. In practice, however, the dot is usually omitted.

2) OR gate: the OR gate is a circuit which gives a high output if one or more of its inputs are high. A plus sign (+) is used to indicate the OR operation.

3) NOT Gate: The Not gate is a circuit which produces at its output the negated (inverted) version of its input logic. The circuit is also known as an inverter. If the input is A, the inverted output is written as $\overline{A}$.

4) NAND Gate: the NAND gate is a NOT-AND circuit which is equivalent to an AND circuit followed by a NOT circuit. The output of the NAND-gate is high if any of its inputs is low.

5) NOR Gate: The NOR gate is a NOT – OR circuit which is equivalent to an OR circuit followed by a NOT circuit. The output of the NOR gate is low if any of its inputs is high.

6) EX-OR Gate: The Exclusive-OR gate is a circuit which gives a high output if either of its two inputs is high, but not both. An encircled plus sign (+) is used to indicate the EX-OR operation.

Table (1.1) shows all possible input/output combinations for two inputs. A truth table with n inputs (Table 1.1) has 2^n rows.

Table (1-1) Logic gates

Inputs		Outputs				
A	B	AND	OR	NAND	NOR	EX-OR
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	0	0

The symbols for the above – mentioned logic gates having two inputs, are summarized in Fig. (1-1)

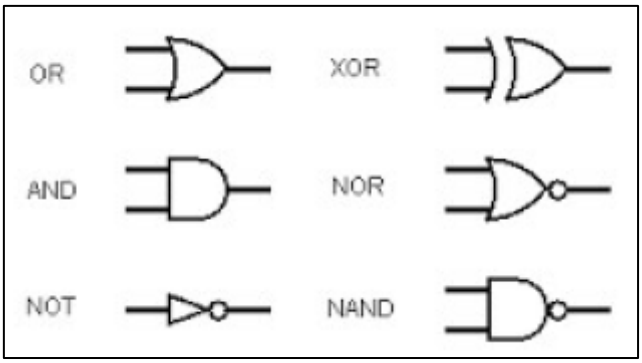

Fig. (1-1) Symbols for logic gates

The functions of the gates described so far can be summarized by means of the idealized waveform diagrams shown in Fig. (1-2)

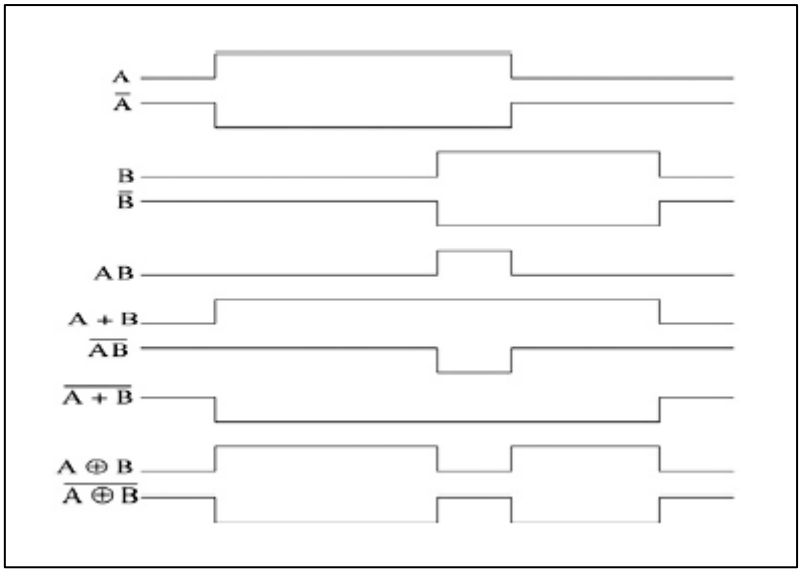

Fig. (1-2) idealized waveform diagram for two-input gates using positive logic.

Though there is a large variety of a gate, it is often desirable to convert a logic expression into a form suitable for whatever type of gates is available; it is desirable to use one type of gate rather than mix different gates. So it is easy to design logic circuits using NAND or NOR gates only. NAND gates can be used to produce any logic function. For this reason, they are referred to as universal gates.

The NAND gate can be used to generate the NOT function, the AND function, the OR function, and the NOR function. An inverter can be made from a NAND gate by connecting all the inputs together and creating, in effect, a single common input, as shown in Fig. (1.3) for two- input gate. An AND function can be generated using only NAND gates, as shown in part (b). Also, an OR function can be produced with NAND gates, as illustrated in part (c). Finally, a NOR function is produced, as shown in part (d).

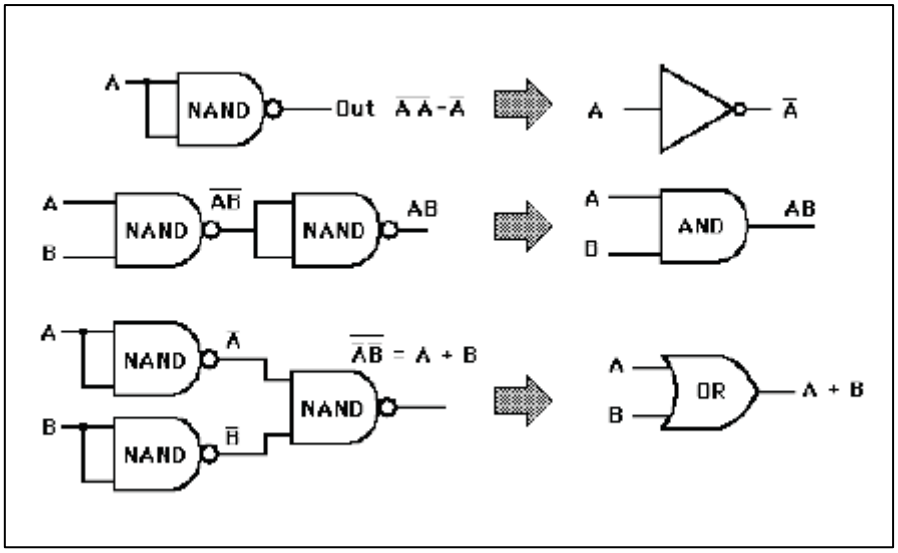

Fig. (1-3) Universal application of NAND gates.

Procedure:

1- By means of NAND gates, find the truth table of all possible gates shown in Fig. (1-1).

Discussion:

1) Implement the following functions using |:

   a) Mixed gates.

   b) NAND gates only.

      i) F_1 =A. B. C + C. D

      ii) F_2 =A. B. C +D

      iii) F_3 =A. B +B. C. D + E. F. G. H

2) Implement the following functions using:

   a) Mixed gates.

   b) NOR gates only

      i) F_1=A+B

      ii) F_2= (A+B). (A+C)

      iii) F_3=A.B+A. B

3) Use only NOR gates to produce the logic function of:

a) NOT gate.

b) OR gate.

c) AND gate.

d) NAND gate.

4) Determine the output waveform for the circuit shown in Fig. (1-4.a), with the inputs as shown.

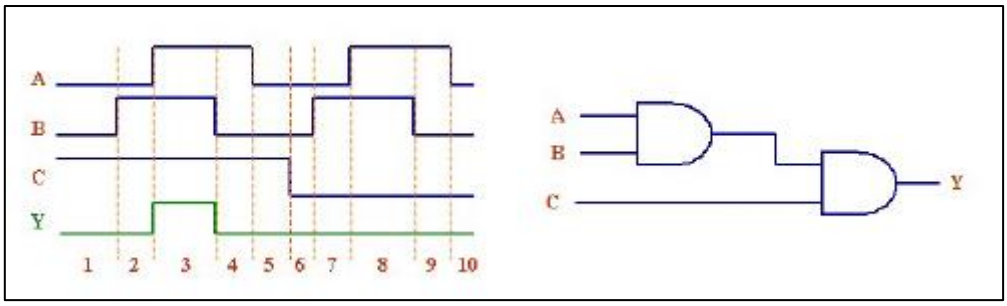

Fig. (1.4.a) output waveform

5) Show how the following expressions can be implemented using NAND gates only:

a) $F_1=A.B.C$

b) $F_2=\overline{A.B.C}$

c) $F_3=\overline{A.B}+\overline{C.D}$

d) $F_4= (A+B).(C+D)$