Logic Gates

- A logic gate is a device that acts as a building block for digital circuits. They perform basic logical functions that are fundamental to digital circuits. Most electronic devices we use today will have some form of logic gates in them. For example, logic gates can be used in digital electronics such as smartphones and tablets or in memory devices.

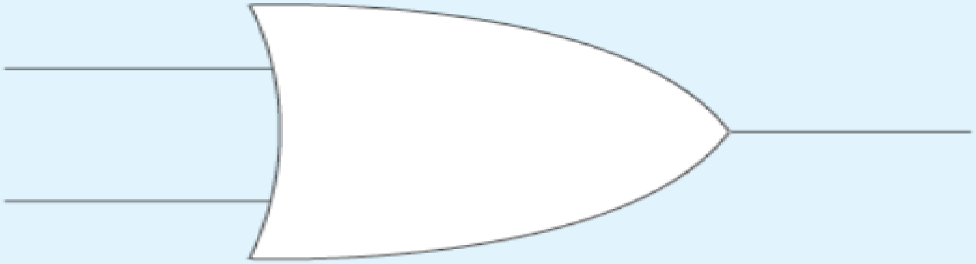
- In a circuit, logic gates work based on a combination of digital signals coming from its inputs. Most logic gates have two inputs and one output, and they are based on Boolean algebra. At any given moment, every terminal is in one of the two binary conditions: true or false. False represents 0, and true represents 1.
- Depending on the type of logic gate being used and the combination of inputs, the binary output will differ. A logic gate can be thought of like a light switch, where in one position the output is off (0), and in another, it is on (1). Logic gates are commonly used in integrated circuits (IC).

Basic logic gates

- There are seven basic logic gates: AND, OR, XOR, NOT, NAND, NOR and XNOR.
- **The AND gate (.)** is named so because, if 0 is false and 1 is true, the gate acts in the same way as the logical "and" operator. The following illustration and table show the circuit symbol and logic combinations for an AND gate. (In the symbol, the input terminals are on the left, and the output terminal is on the right.) The output is "true" when both inputs are "true." Otherwise, the output is "false." In other words, the output is 1 only when both inputs are 1.

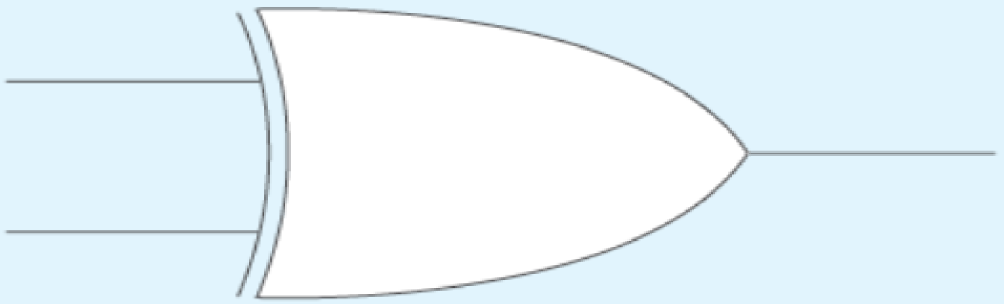
Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

- **The OR gate (+)** gets its name from behaving like the logical inclusive "or." The output is true if one or both of the inputs are true. If both inputs are false, then the output is false. In other words, for the output to be 1, at least one input must be 1.



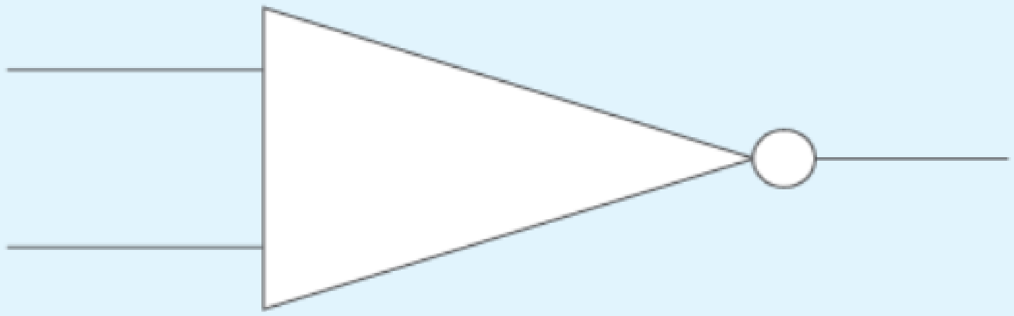
Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

- **The XOR (exclusive-OR) gate** acts in the same way as the logical "either/or." The output is true if either, but not both, of the inputs are true. The output is false if both inputs are "false" or if both inputs are true. Similarly, the output is 1 if the inputs are different but 0 if the inputs are the same.



Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

- **A logical inverter**, sometimes called a NOT gate to differentiate it from other types of electronic inverter devices, has only one input. A NOT gate reverses the logic state. If the input is 1, then the output is 0. If the input is 0, then the output is 1.

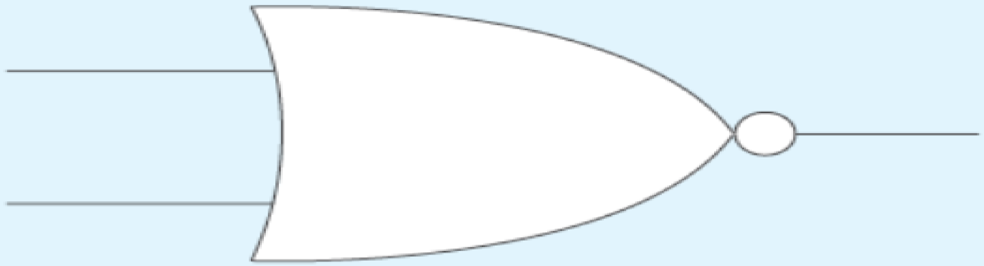


Input 1	Output
1	0
0	1

- **The NAND (Negated AND) gate** operates as an AND gate followed by a NOT gate. It acts in the manner of the logical operation "and" followed by negation. The output is false if both inputs are true. Otherwise, the output is true. Another way to visualize it is that a NAND gate inverts the output of an AND gate. The NAND gate symbol is an AND gate with the circle of a NOT gate at the output.

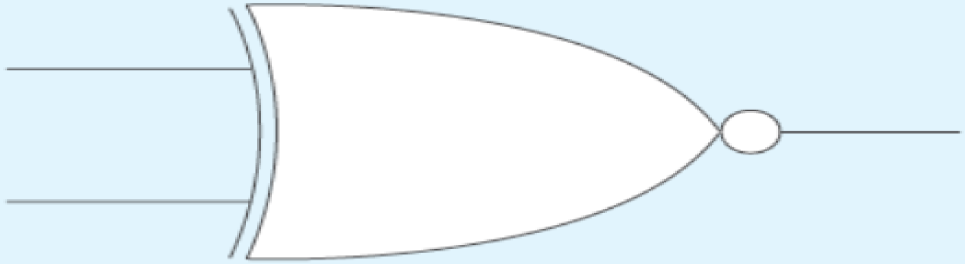
Input 1	Input 2	Output
0	0	1
0	1	1
1	0	1
1	1	0

- **The NOR (NOT OR) gate** is a combination OR gate followed by an inverter. Its output is true if both inputs are false. Otherwise, the output is false.



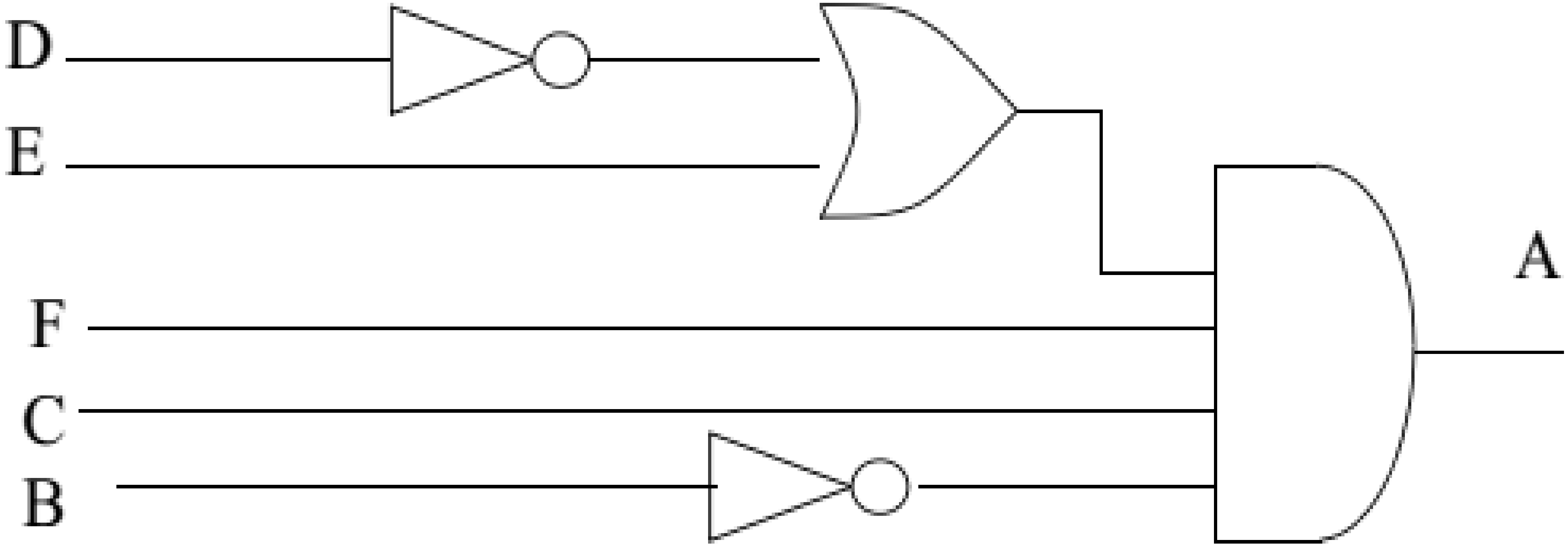
Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	0

- **The XNOR (exclusive-NOR) gate** is a combination of an XOR gate followed by an inverter. Its output is true if the inputs are the same and false if the inputs are different.

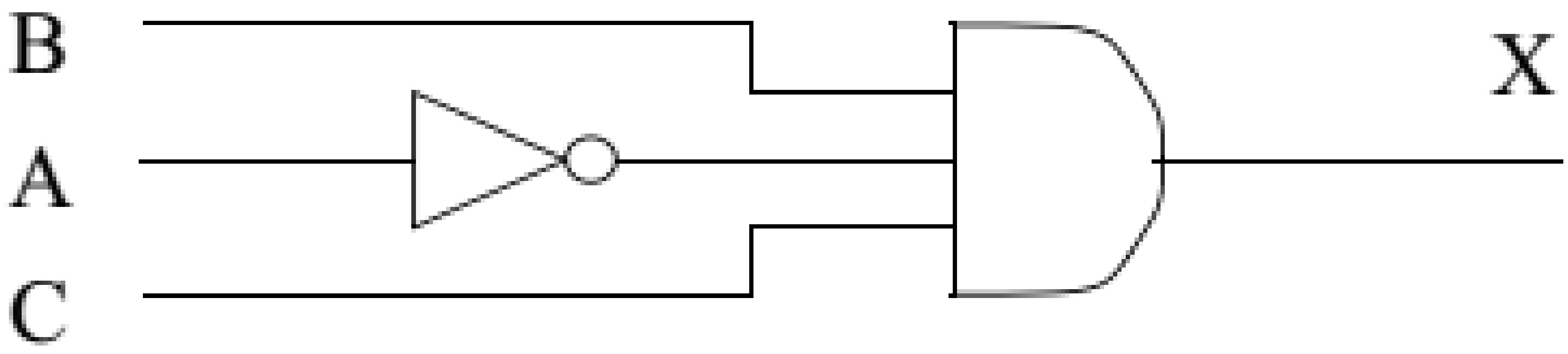


Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	1

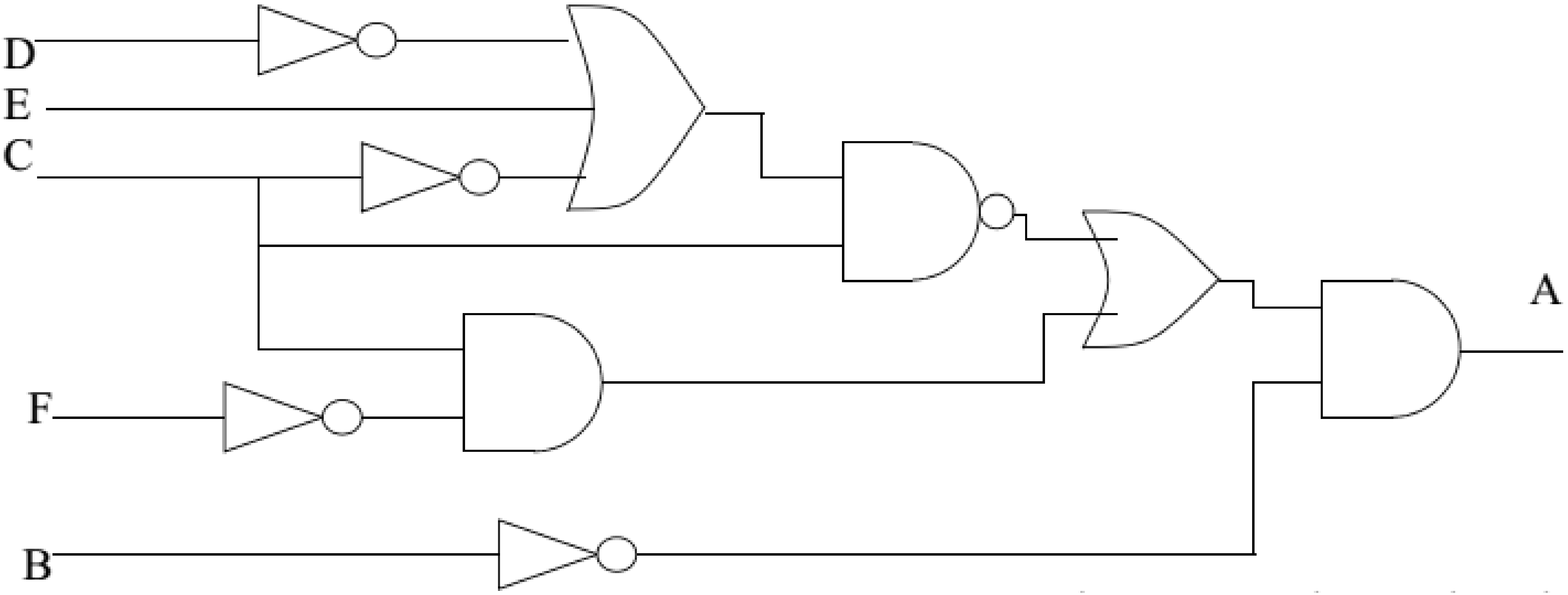
$$A = \bar{B} \cdot C \cdot F \cdot (\bar{D} + E)$$



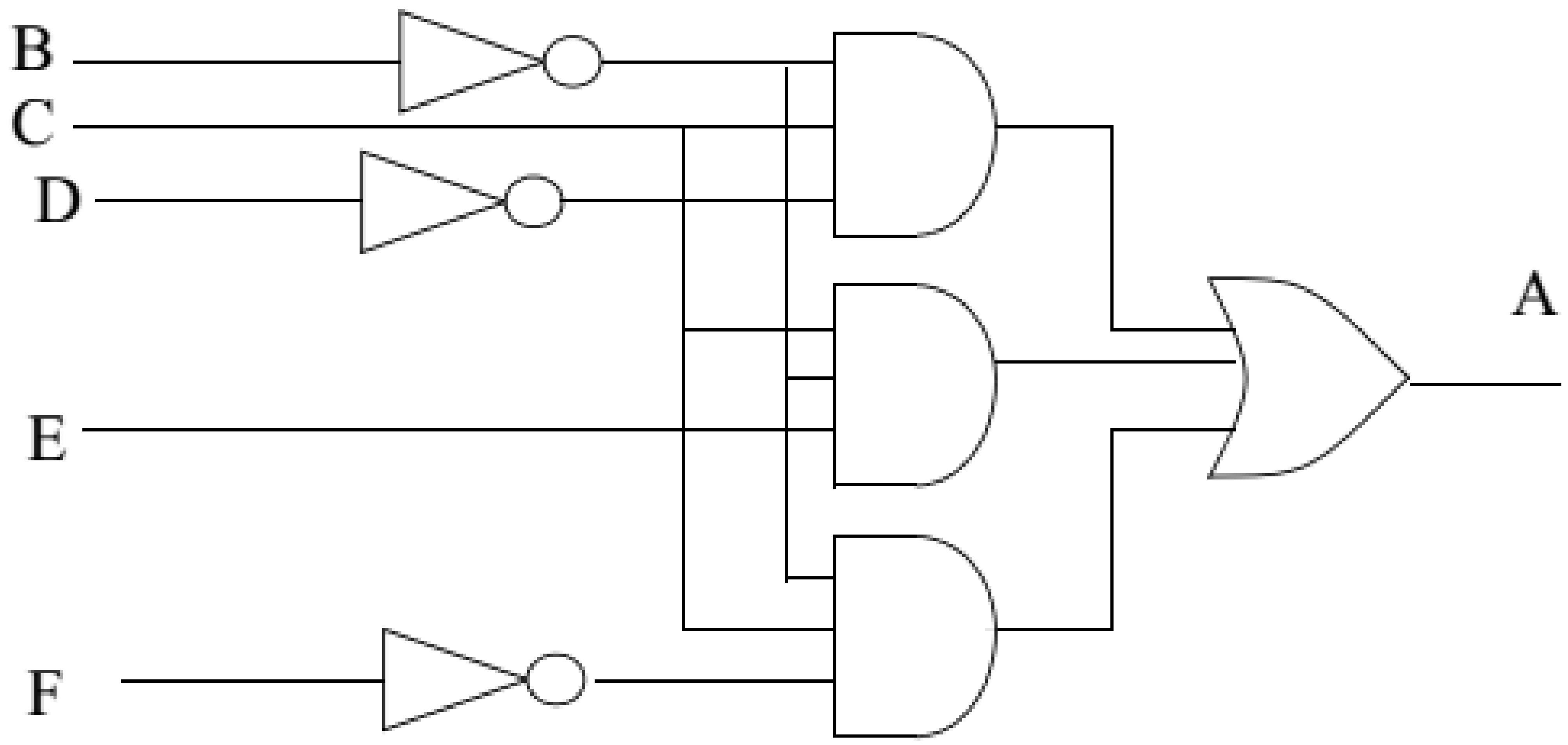
$$X = B \cdot \bar{A} \cdot C$$



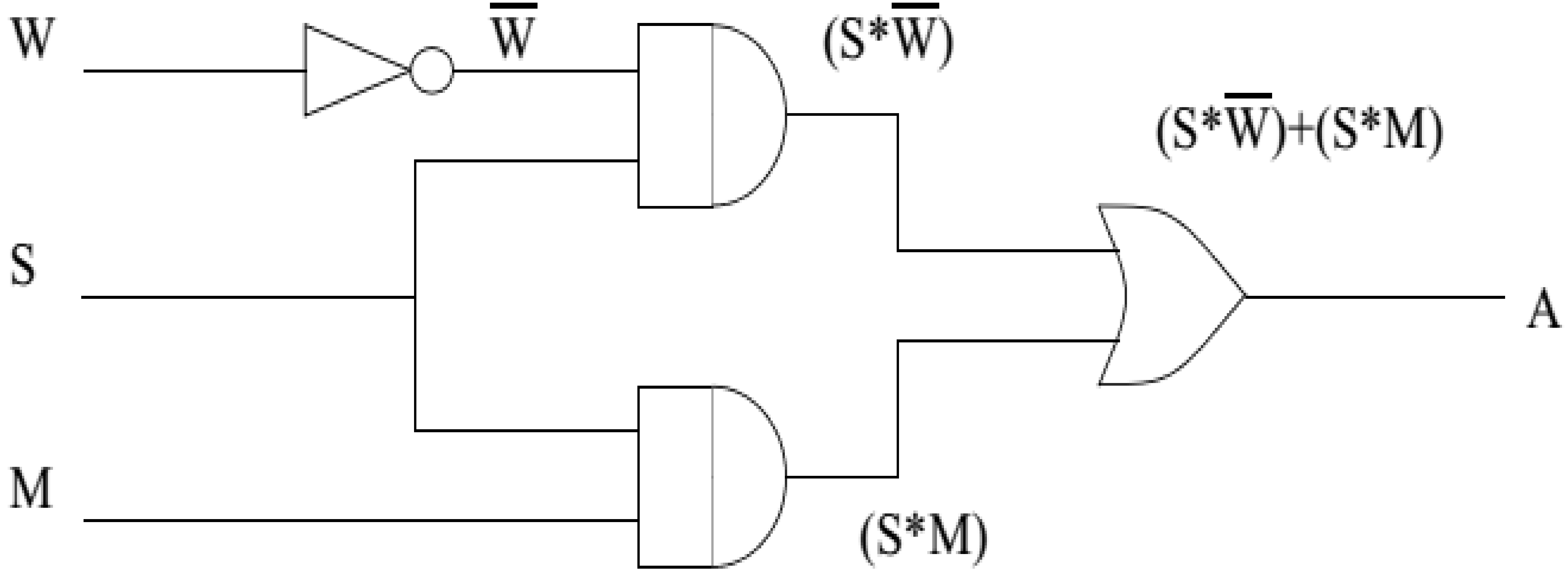
$$A = \overline{B} \cdot (\overline{C \cdot (\overline{D} + E + \overline{C})} + \overline{F} \cdot C)$$



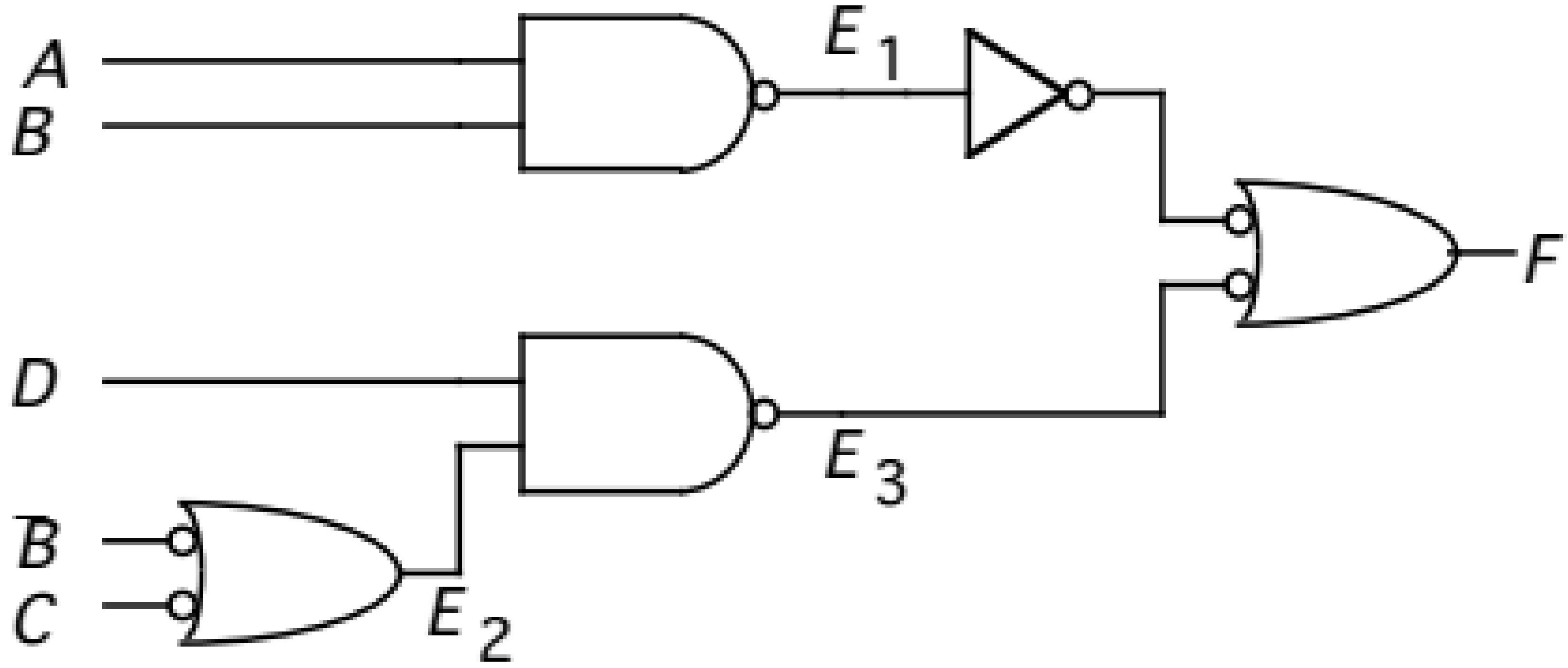
$$A = (\bar{B} \cdot C \cdot \bar{D}) + (\bar{B} \cdot C \cdot E) + (\bar{B} \cdot C \cdot \bar{F})$$

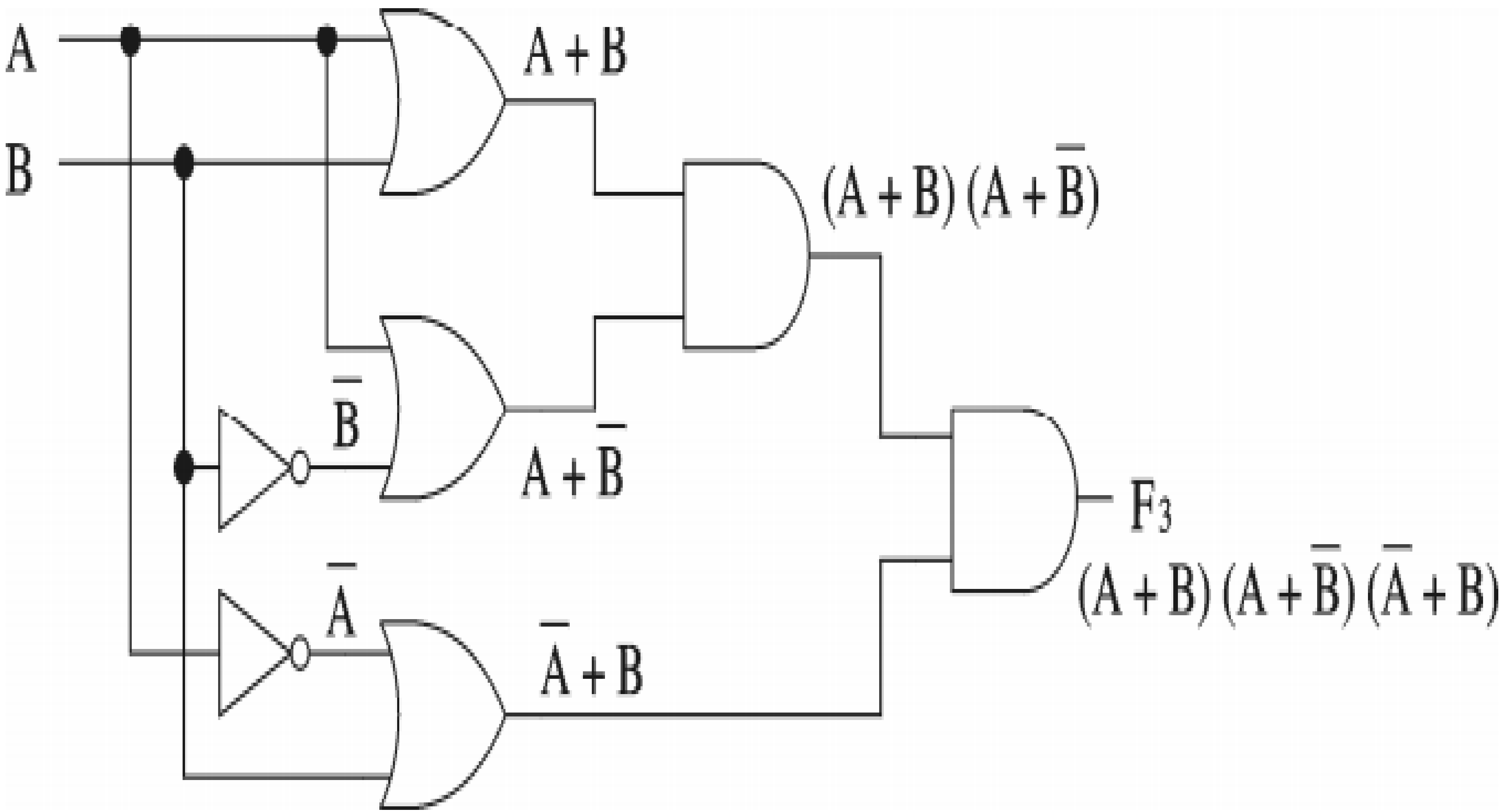


$$A = (S \cdot \overline{W}) + (S \cdot M) = S \cdot (\overline{W} + M)$$



Implement $F = \overline{AB} + D(B + \overline{C})$




A
B
A + B
(A + B) (A + B̄)
B̄
A + B̄
Ā
Ā + B
F3
(A + B) (A + B̄) (Ā + B)