

Designing Combinational Circuits

Bob Brown
Computer Science Department
Southern Polytechnic State University

Combinational Circuits

A combinational circuit is one for which the output value is determined solely by the values of the inputs. Such a circuit can be represented by a truth table and computes a Boolean function. Desired characteristics of such a circuit are a minimum number of gates, simple rather than complex gates, and a minimum number of levels of gates. The first two characteristics can be stated as minimizing the number of gate inputs, which has the effect of minimizing cost. Minimizing the number of levels minimizes the number of gate delays and makes the circuit faster. For simple functions, it is possible to design a circuit by inspection of the function. For more complex functions, the appropriate circuit design may not be immediately obvious. Fortunately, there exists a completely mechanical way of designing a correct circuit for any Boolean function, that is, any function which can be represented by a truth table. A circuit designed in this way will be a two-level circuit, plus any inverters needed to form complement terms. Unless special steps are taken to simplify the expression which drives the circuit design, such a circuit will *not* necessarily have the minimum number of gate inputs.

Sum-Of-Products

Consider the truth table shown at the right. This represents a function F of three variables, A , B , and C . This truth table can be converted to a Boolean expression in standard form by identifying the rows which result in a value of one for F , writing the product (AND) of the variables for each such row, then writing the Boolean sum (OR) of the product terms. Such an expression is called a sum-of-products.* For example, the first row with a result of one is the second row. A and B are zeroes, and C is a one, so the product term is $\overline{A}\overline{B}C$. The sum-of-products expression for this truth table is:

$$F = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C}$$

This says that F is true when A and B are false and C is true OR when A and C are false and B is true OR when A is true and B and C are false. It also implies that F is false for all other combinations. It is possible to construct a circuit diagram directly from such an expression.

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

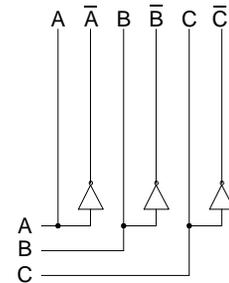
* Strictly speaking, this is called the *minterm form* because each product term contains every variable or its complement. Such an expression can be simplified to remove variables from product terms and still be a sum-of-products expression.

Circuit Design Using Sum-Of-Products

A correct two-level circuit can be generated for any sum-of-products expression using the following steps:

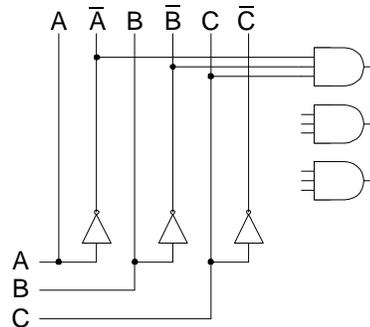
- Provide an inverter for each variable.
- Provide an AND gate for each product term; the number of inputs required is determined by the number of variables in the product term
- Connect the AND gates according to the product term.
- Provide an OR gate for the output; the number of inputs required is determined by the number of AND gates.

The figure at the right shows the result of the first step. At the lower left are inputs for each of the three variables, A, B, and C. For each variable, an inverter computes its complement. When A is true, or one, \bar{A} is zero, and vice-versa. The vertical wires provide for easy access to each variable and its complement.

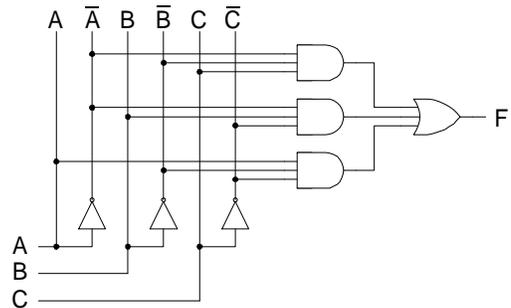


For the next step, we provide one AND gate for each of the product terms in the sum-of-products expression. There are three product terms in the example expression, so we will need three AND gates. The AND gates are placed at the right of the vertical wires; horizontal wires connect the AND gates to the proper variables or their complements. The first product term is $\bar{A}\bar{B}C$, so we wire the inputs of the first AND gate to the vertical wires for \bar{A} ,

\bar{B} , and C as shown in the figure. Connections are shown as heavy dots. Wires which cross without dots are not connected. The other two and gates are wired similarly, with wires from the gates connecting to the vertical wires which represent the variables of the product terms to be computed by each gate.



The final step is to provide an OR gate to compute the Boolean sum of the product terms computed by the AND gates. Only one OR gate is necessary, but it must have one input for each of the AND gates, *i.e.* for each of the product terms in the original expression. The final circuit is shown here. Such a circuit is always a two-level circuit, and it always correctly computes the function described by the truth table from which it is derived. However, it is not guaranteed to be the simplest circuit that will compute the function.



A minterm form expression such as we have used can be simplified using Karnaugh maps, manipulation with Boolean algebra, or some other method. So long as the result is a sum-of-products expression, this method can be used to produce a correct circuit for it.