The example below illustrates the use of Boolean identities to simplify a logic expression:

```
F(x,y,z) = xy + x'z + yz
```

$$
\begin{aligned}
F(x,y,z) &= xy + x'z + yz \\
&= xy + x'z + yz(1) && \text{(Identity)} \\
&= xy + x'z + yz(x + x') && \text{(Inverse)} \\
&= xy + x'z + (yz)x + (yz)x' && \text{(Distributive)} \\
&= xy + x'z + x(yz) + x'(zy) && \text{(Commutative)} \\
&= xy + x'z + (xy)z + (x'z)y && \text{(Associative twice)} \\
&= xy + (xy)z + x'z + (x'z)y && \text{(Commutative)} \\
&= xy(1 + z) + x'z(1 + y) && \text{(Distributive)} \\
&= xy(1) + x'z(1) && \text{(Null)} \\
&= xy + x'z && \text{(Identity)}
\end{aligned}
$$

- A Boolean expression may have multiple logically equivalent (or “synonymous”) forms

- There are two canonical (i.e. standardized) forms for Boolean expressions:
  - sum-of-products and product-of-sums.

- The Boolean product is the AND operation and the Boolean sum is the OR operation.

In the sum-of-products form, ANDed variables are ORed together.

For example:

$$F(x,y,z) = xy + xz + yz$$

In the product-of-sums form, ORed variables are ANDed together:

For example:

$$F(x,y,z) = (x+y)\,(x+z)\,(y+z)$$

To obtain the sum-of-products form using its truth table, we list the values of the variables that result in a true function value (=1).

Each group of variables is then ORed together.

$$F(x,y,z) = (x'yz') + (x'yz) + (xy'z') + (xyz') + (xyz)$$

Use the complement of the 0 inputs. The result is the logical sum of the non-zero "minterms".

$$F(x,y,z) = xz' + y$$

x	y	z	xz'+ y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

To obtain the product-of-sums form using its truth table, we list the values of the variables that result in a false function value (=0).

Each group of variables is then ANDed together.

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

Use the complement of the 1 inputs. The result is the logical product of the Zero “maxterms”.

$$F(x,y,z) = xz' + y$$

x	y	z	xz'+ y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

It can be more economical to build a circuit using the complement of a function (and complementing its result) than to implement the function directly.

DeMorgan's law provides an easy way of finding the complement of a Boolean function.

Recall DeMorgan's law states:

$$(xy)' = x' + y' \quad \text{and} \quad (x + y)' = x'y'$$

DeMorgan's law can be extended to any number of variables.

Replace each variable by its complement and change all ANDs to ORs and all ORs to ANDs.

Thus, we find that the complement of:

$$F(x,y,z) = (xy) + (x'y) + (xz')$$

is:

$$\begin{aligned} F'(x,y,z) &= ((xy) + (x'y) + (xz'))' \\ &= (xy)'(x'y)'(xz')' \\ &= (x' + y')(x + y')(x' + z) \end{aligned}$$

This concludes our examination of Boolean Algebra as a basis for describing the behavior of logic circuits.

Next we will see how logic gates are used to implement various functions required within the computer.