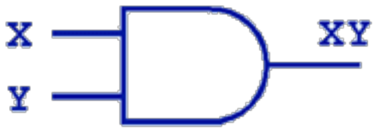
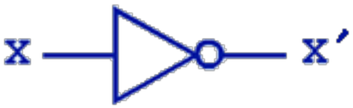
Logic Gates

- A gate is an electronic device that produces a result based on one or more digital input values.
  - In reality, gates consist of one to six transistors, but digital designers think of them as a single unit.
  - Integrated circuits contain collections of gates suited to a particular purpose.
- Digital computer circuits employ logic gates to implement Boolean functions.

- The three simplest gates are the AND, OR, and NOT gates.





X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT X

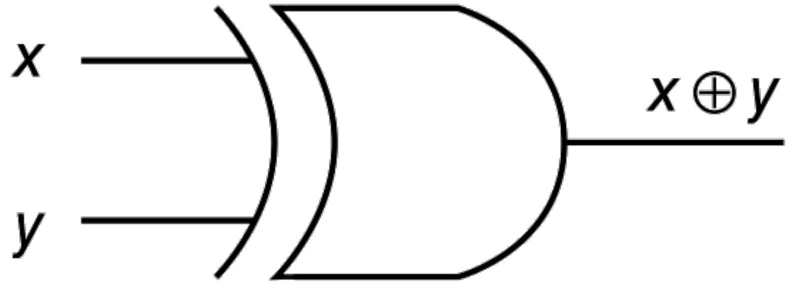
X	X′
0	1
1	0

- They correspond directly to their respective Boolean operations, as shown in their truth tables.

- Another very useful gate is the exclusive OR (XOR) gate.
- The output of the XOR operation is true only when the values of the inputs differ.

X XOR Y

X	Y	X ⊕ Y
0	0	0
0	1	1
1	0	1
1	1	0

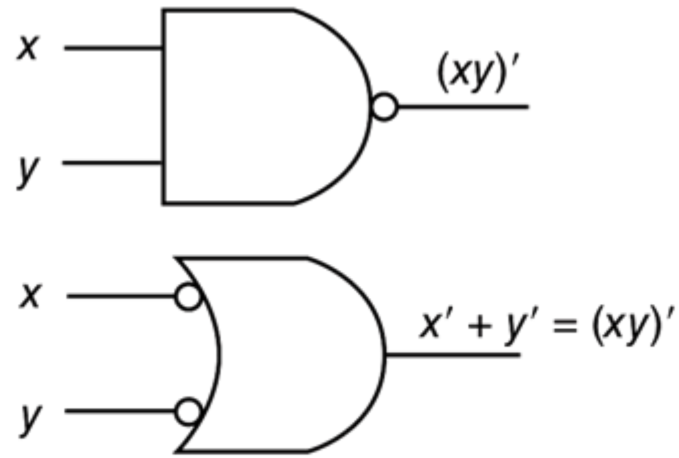


The symbol ⊕ denotes the XOR operator.

- Two other important gates are the NAND and NOR gates. The truth table for the NAND gate is shown below:

X NAND Y

X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0

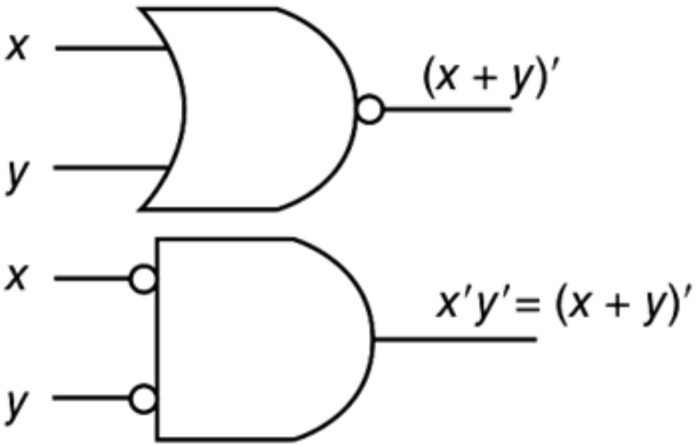


- The open circles are inversion bubbles. The two gates above are equivalent based on DeMorgan’s theorem.

- The truth table for the NOR gate is shown below along with two equivalent implementations:

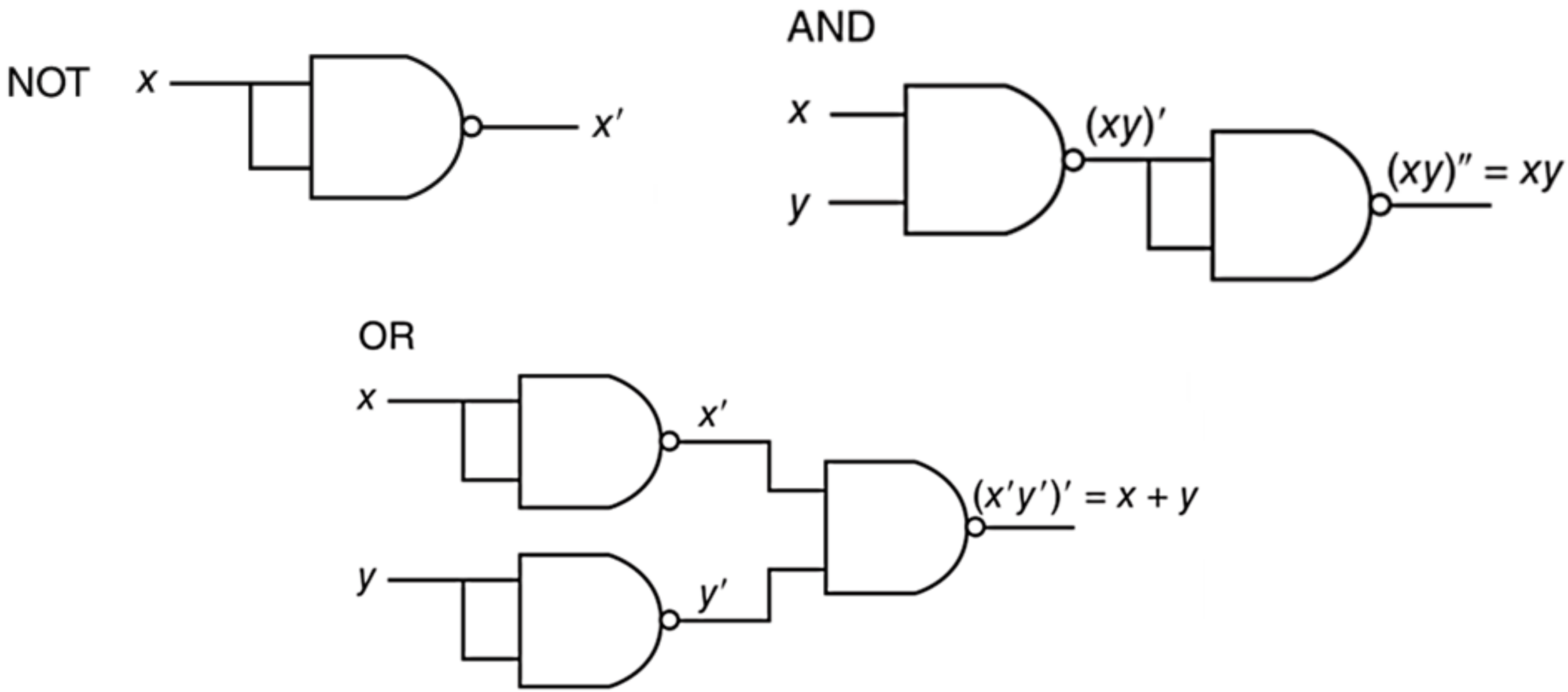
X NOR Y

X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0

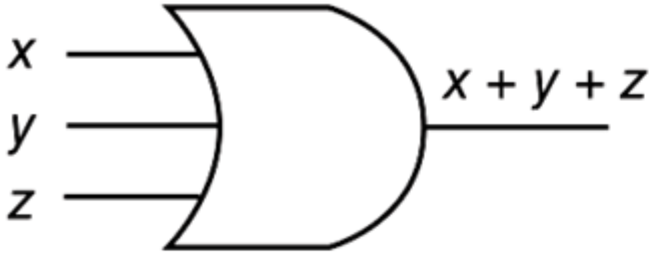


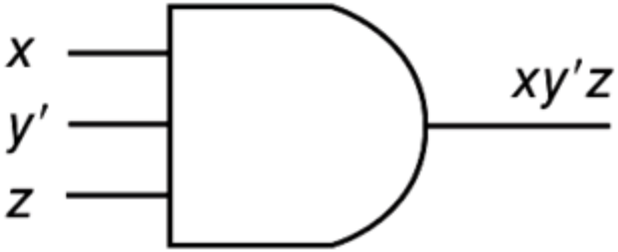
- NAND and NOR gates are said to be *universal* gates because they can be used to implement any logic function.

- The examples below show how NAND gates alone can be used to implement NOT, AND, and OR functions:

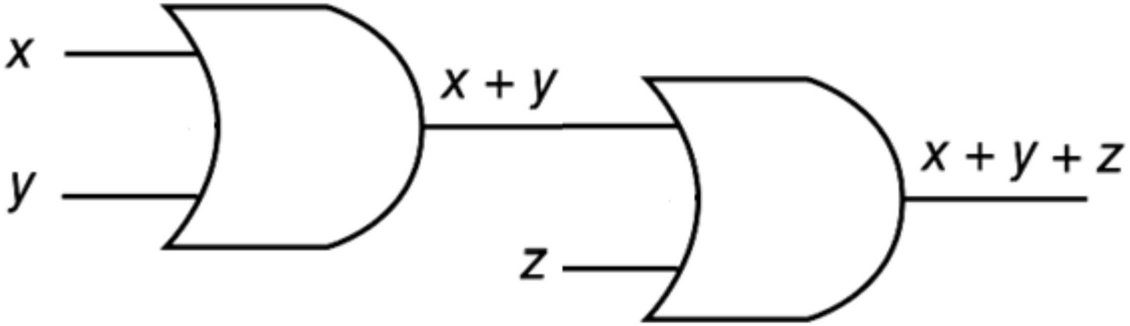


- Here are examples of 3-input gates:

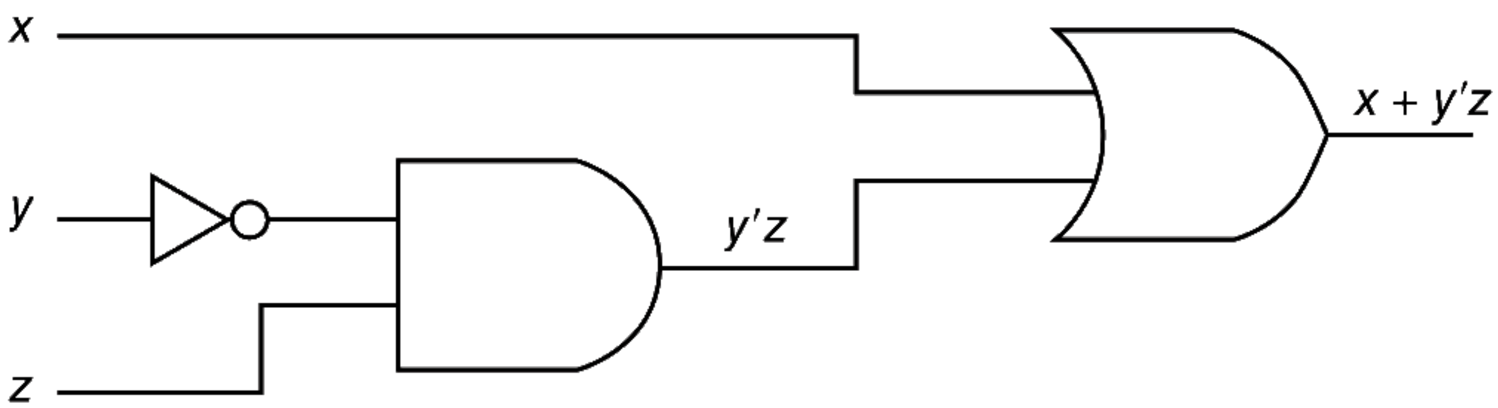




- However the same result can be generated using multiple 2-input gates:

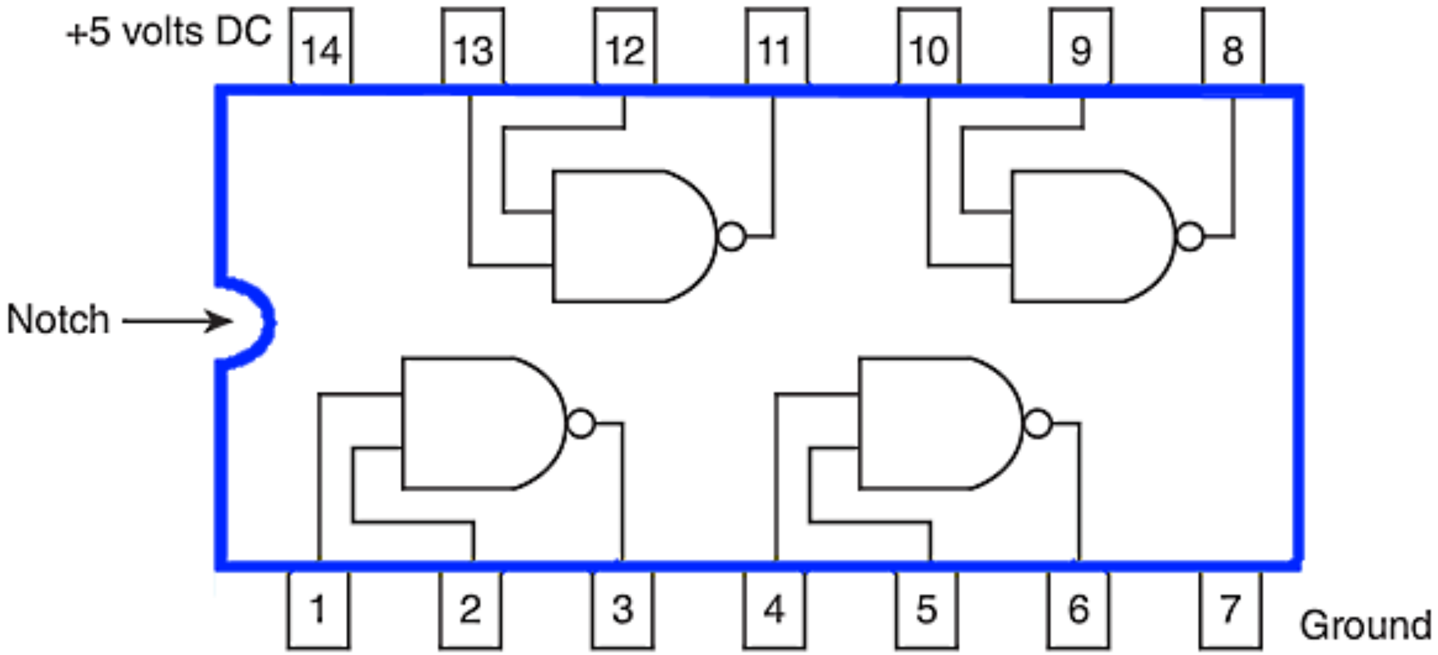


- The circuit below implements the Boolean function **F(x,y,z) = x+y′z**:



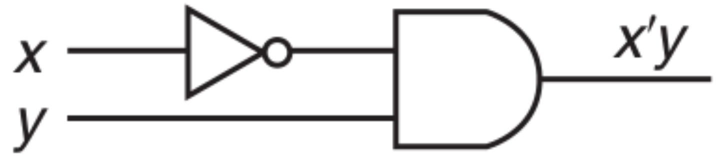
- The logic gate implementation of any function can be derived from its truth table. However, the resulting expression should be simplified to minimize the number of gates required.

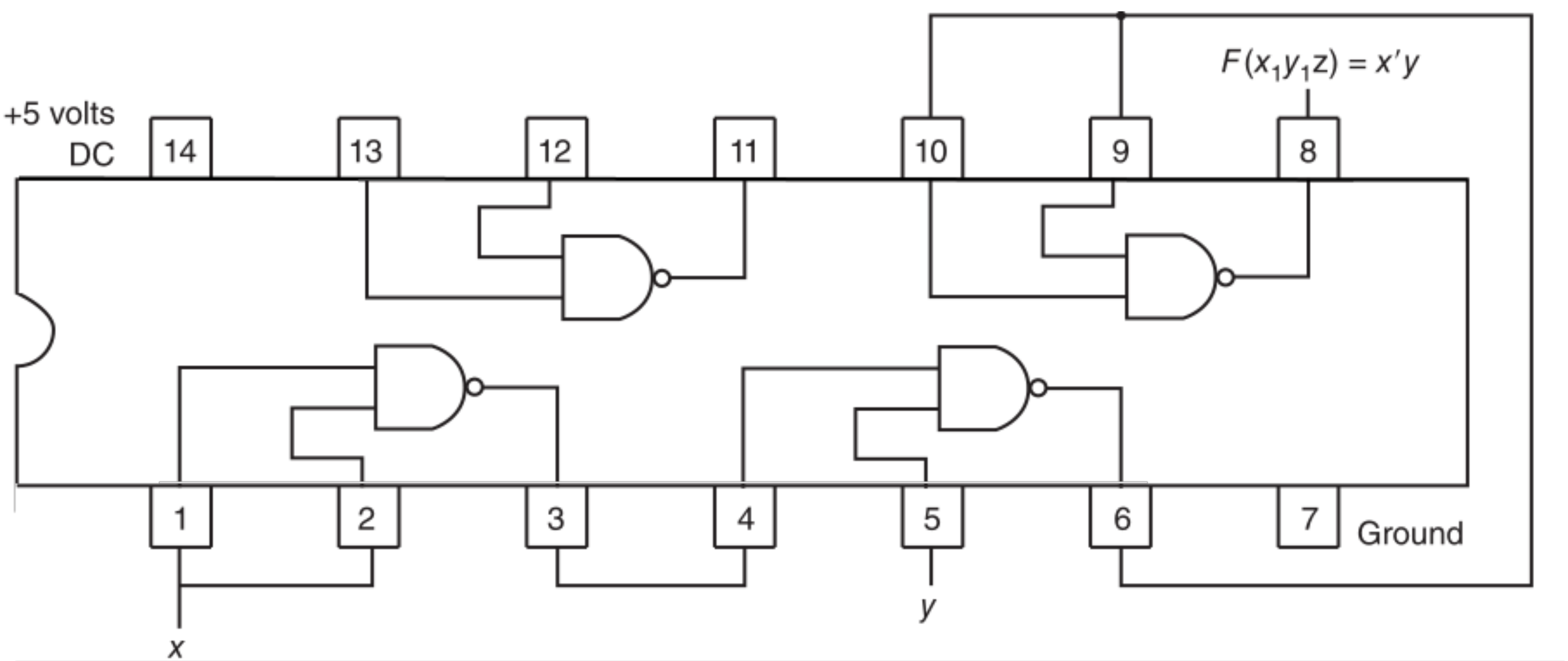
- Standard digital components are combined into single integrated circuit packages.



This is a small scale integrated circuit containing 4 NAND gates.

This chip's pins can be wired as shown below to implement the function:





Suppose we wanted to design a lighting control system that turns lights off when they are not needed.

Assume the lights are to be turned off if it is before 6 PM or if the Sun is out and it is a week day.

The decision would be based on 3 inputs:

- $x = 1$ if the time is before 6 PM (or 0 otherwise)
- $y = 1$ if the Sun is out (or 0 otherwise)
- $z = 1$ if it is a week day (or 0 otherwise)

The output should = 1 if the lights are to be turned off

The truth table for the lights out function is:

Before 6 PM	Sun is out	Week day	Lights out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Using the logical sum of the non-zero min-terms we get:

lights out = $X' Y Z + X Y' Z + X Y Z$

Simplifying we get:

lights out = X' Y Z + X Y' Z + X Y Z = (X' Y + X Y' + X Y) Z

= ((X' Y) + X(Y' + Y)) Z = (X' Y + X) Z = (X' + X)(X+Y) Z

= (X + Y) Z