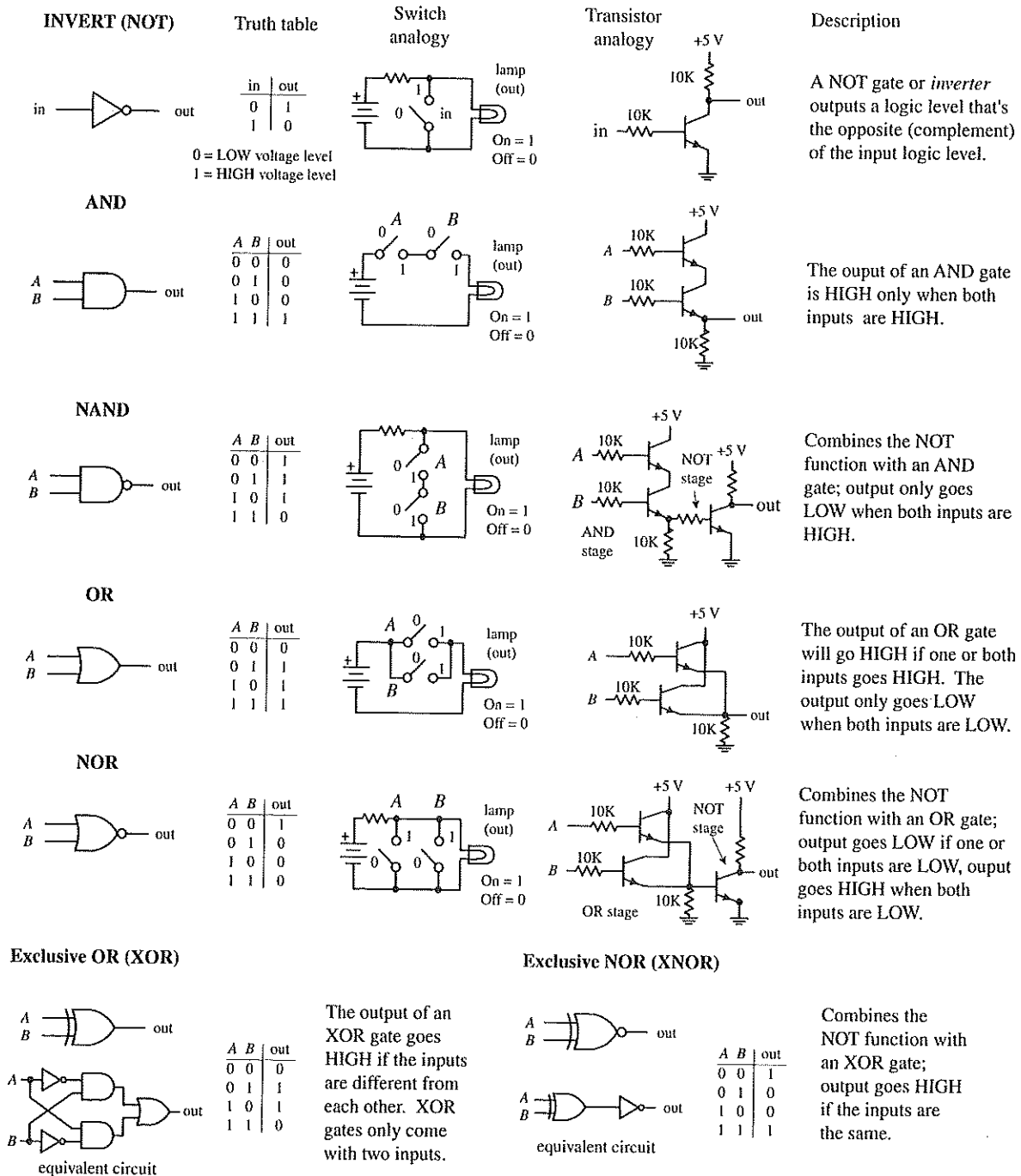


12.2 Logic Gates

Logic gates are the building blocks of digital electronics. The fundamental logic gates include the INVERT (NOT), AND, NAND, OR, NOR, exclusive OR (XOR), and exclusive NOR (XNOR) gates. Each of these gates performs a different logical operation. Figure 12.10 provides a description of what each logic gate does and gives a switch and transistor analogy for each gate.

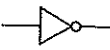

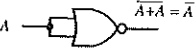

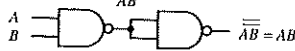

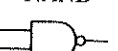
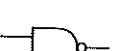




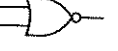
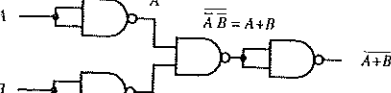

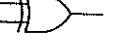

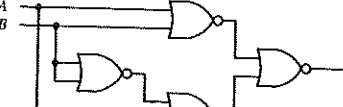
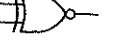
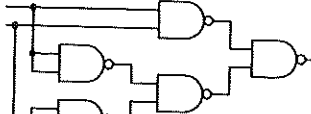
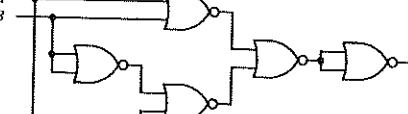
FIGURE 12.10



Universal Capability of NAND and NOR Gates

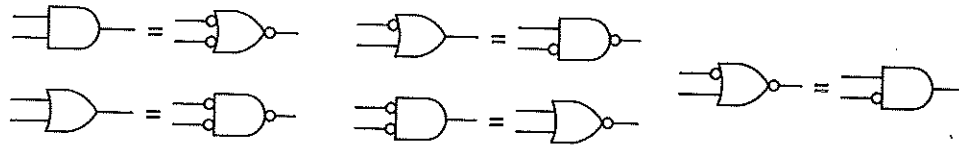
NAND and NOR gates are referred to as *universal gates* because each alone can be combined together with itself to form all other possible logic gates. The ability to create any logic gate from NAND or NOR gates is obviously a handy feature. For example, if you do not have an XOR IC handy, you can use a single multigate NAND gate (e.g., 74HC00) instead. The figure below shows how to wire NAND or NOR gates together to create equivalent circuits of the various logic gates.

FIGURE 12.24

Logic gate	NAND equivalent circuit	NOR equivalent circuit
	 $\overline{AA} = \bar{A}$	 $\overline{A+A} = \bar{A}$
	 $\overline{\overline{AB}} = AB$	 $\overline{\overline{A+B}} = AB$
		 $\overline{A+B}$
	 $\overline{\overline{A} \overline{B}} = A+B$	 $\overline{\overline{A+B}} = A+B$
	 $\overline{A+B}$	
		
		

Bubble Pushing

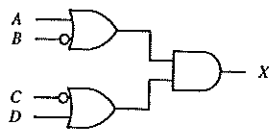
A shortcut method for forming equivalent logic circuits, based on De Morgan's theorem, is to use what's called *bubble pushing*.



AND-OR-INVERT Gates (AOIs)

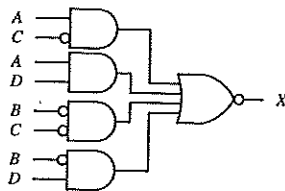
When a Boolean expression is reduced, the equation that is left over typically will be of one of the following two forms: *product-of-sums* (POS) or *sum-of-products* (SOP). A POS expression appears as two or more ORed variables ANDed together with two or more additional ORed variables. An SOP expression appears as two or more ANDed variables ORed together with additional ANDed variables. The figure below shows two circuits that provide the same logic function (they are equivalent), but the circuit to the left is designed to yield a POS expression, while the circuit to the right is designed to yield a SOP expression.

Logic circuit for POS expression



$$X = (A + B)(\bar{C} + D)$$

Logic circuit for SOP expression



$$X = A\bar{C} + AD + \bar{B}C + \bar{B}D$$

Table made using SOP expression
(it's easier than POS)

A	B	C	D	$A\bar{C}$	AD	$\bar{B}C$	$\bar{B}D$	X
0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	1	1	1
0	0	1	0	0	0	0	0	0
0	0	1	1	0	0	0	1	1
0	1	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0
1	0	0	0	1	0	1	0	1
1	0	0	1	1	1	1	1	1
1	0	1	0	0	0	0	0	0
1	0	1	1	0	1	0	1	1
1	1	0	0	1	0	0	0	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	0	0	0
1	1	1	1	0	1	0	0	1

FIGURE 12.25

Bubble pushing involves the following tricks: First, change an AND gate to an OR gate or change an OR gate to an AND gate. Second, add inversion bubbles to the inputs and outputs where there were none, while removing the original bubbles. That's it. You can prove to yourself that this works by examining the corresponding truth tables for the original gate and the bubble-pushed gate, or you can work out the Boolean expressions using De Morgan's theorem. Figure 12.23 shows examples of bubble pushing.