EE 2720

Handout #9

• n-input AND gate

As said earlier in the semester, a 2-input AND gate is a digital device whose output is 1 if and only if both its inputs are 1. Similarly, an n-input AND gate is a digital device with n inputs whose output is 1 if and only if all its inputs are 1. A truth table for an n-input AND gate and the corresponding figure (it is figure 1) are shown below.

X_1	X_2	....	X_{n-1}	X_n	$X_1 \cdot X_2 \cdot \ldots \cdot X_{n-1} \cdot X_n$
0	0	....	0	0	0
0	0	....	0	1	0
0	0	....	1	0	0
0	0	....	1	1	0
0	0	...1	0	0	0
		⋮			⋮ } all zeroes
1	1	...1	1	0	0
1	1	1	1	1	1

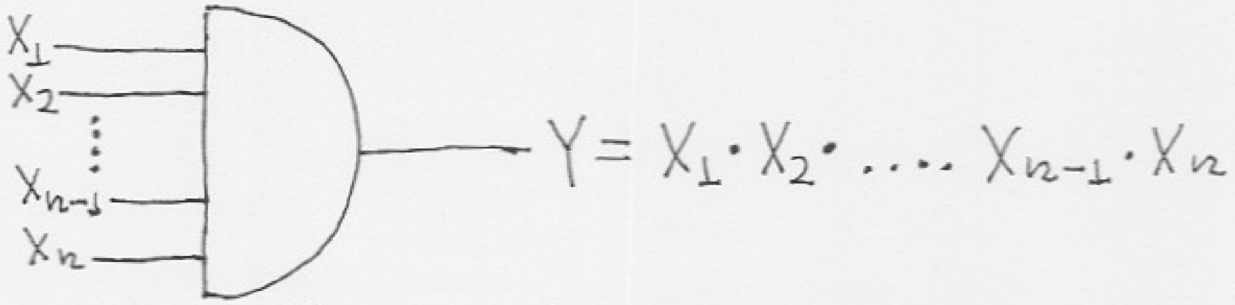


Figure 1; (n-input AND gate).

- n-input OR gate

As you already know, a 2-input OR gate is a digital device whose output is 1 iff either of its inputs or both are 1. Similarly, an n-input OR gate is a digital device with n inputs whose output is 1 if anyone of its inputs is 1. Its output is 0 only if all its inputs are zeroes. A truth table for an n-input OR gate and the corresponding figure (it is figure 2) are shown below.

$X_1\ X_2 \cdots\cdot X_{n-1}\ X_n$	$X_1+X_2+\cdots\cdot+X_{n-1}+X_n$
0 0 .... 0 0	0
0 0 .... 0 1	1
0 0 .... 1 0	1
0 0 .... 1 1	1
0 0 ... 1 0 0	1
⋮	⋮ } all ones
1 1 ... 1 1 0	1
1 1 ... 1 1 1	1

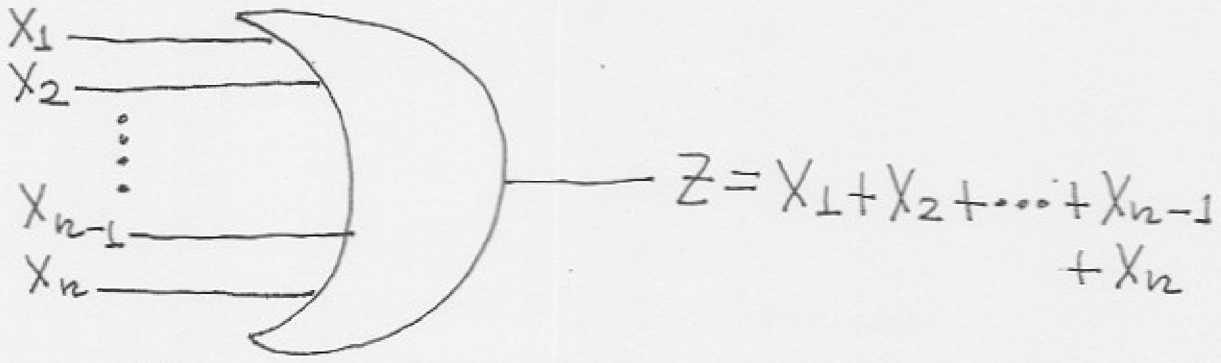


Figure 2; (n-input OR gate).

• n-input NAND gate

An n-input NAND gate is a digital device with n inputs whose output is the complement of the output of an n-input AND gate that has the same inputs. In other words, the output of this gate is 1 if and only if one or more of its inputs are 0. If all its inputs are 1, then its output is 0. A truth table for an n-input NAND gate and some corresponding figures follow:

X_1	X_2	....	X_{n-1}	X_n	$(X_1 \cdot X_2 \ldots X_{n-1} \cdot X_n)'$
0	0	....	0	0	1
0	0	....	0	1	1
0	0	....	1	0	1
0	0	....	1	1	1
0	0	...1	0	0	1
		⋮			⋮ } all ones
1	1	...	1 1	0	1
1	1	...	1 1	1	0

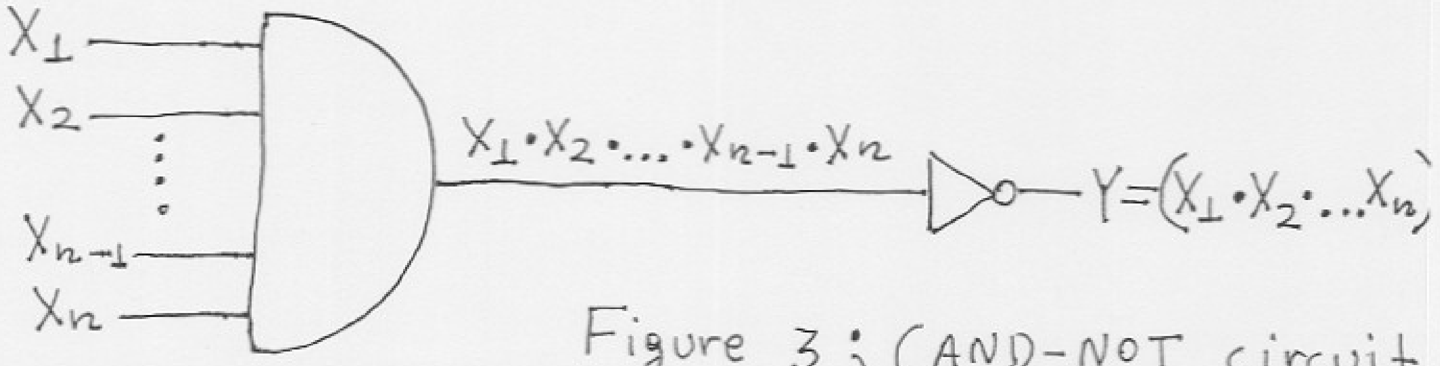

X_1
X_2
X_{n-1}
X_n
$X_1 \cdot X_2 \cdot \ldots \cdot X_{n-1} \cdot X_n$
$Y = (X_1 \cdot X_2 \cdot \ldots X_n)$

Figure 3; (AND-NOT circuit representing the function of the n-input NAND gate).

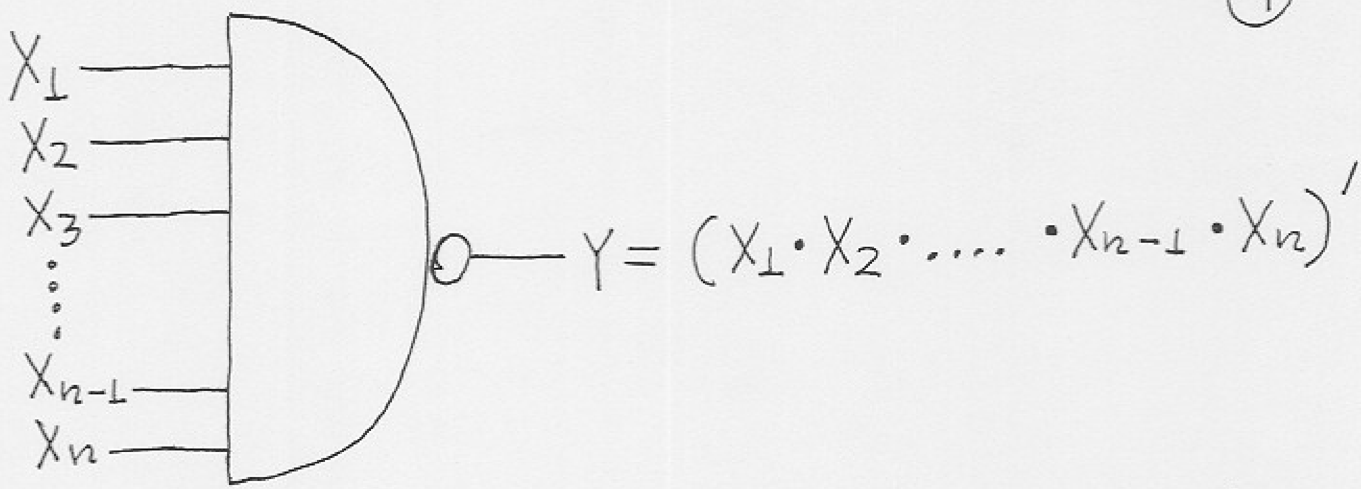


Figure 4; (official logic symbol for an n-input NAND gate).

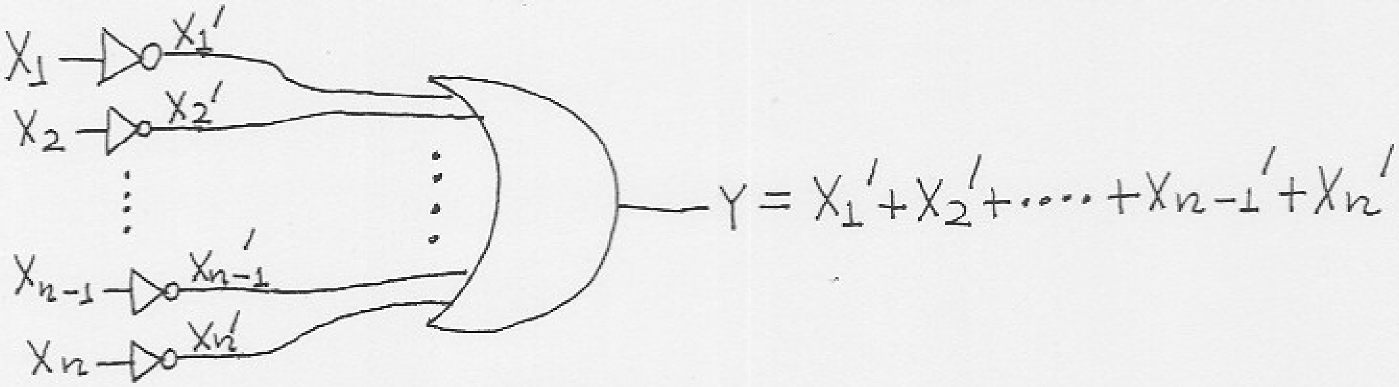


Figure 5; (NOT-OR circuit representing the function of the n-input NAND gate; this figure was obtained from fig. 3 by applying DeMorgan's theorem T13; Figures 3 and 5 are equivalent).

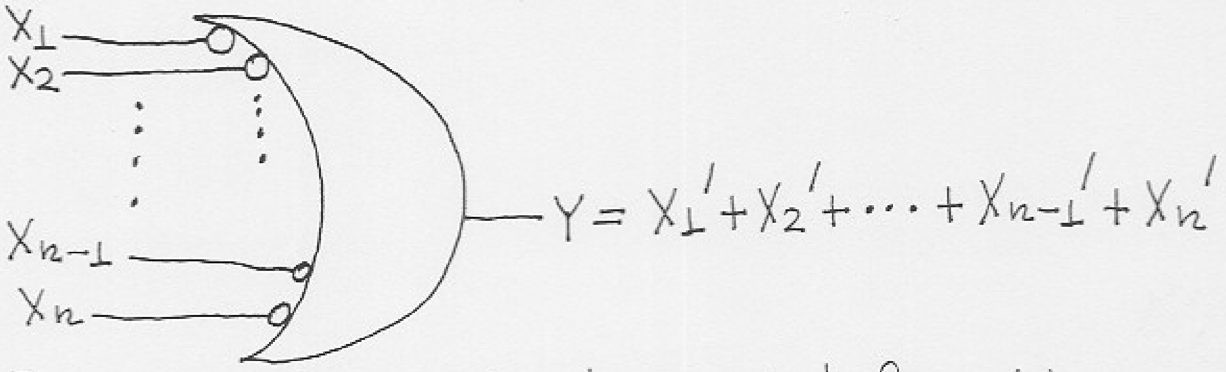


Figure 6; (equivalent symbol for the n-input NAND gate).

Note: When you want to represent an n-input NAND gate you should use the logic symbols of either figure 4 or figure 6

• n-input NOR gate

An n-input NOR gate is a digital device with n inputs whose output is the complement of the output of an n-input OR gate that has the same inputs. A truth table for an n-input NOR gate follows. This table makes the situation clear. Also, some corresponding figures follow:

X_1	X_2	....	X_{n-1}	X_n	$(X_1+X_2+....+X_{n-1}+X_n)'$
0	0	....	0	0	1
0	0	....	0	1	0
0	0	....	1	0	0
0	0	....	1	1	0
0	0	...1	0	0	0
		⋮			⋮ } all zeroes
1	1	...1	1	0	0
1	1	...1	1	1	0

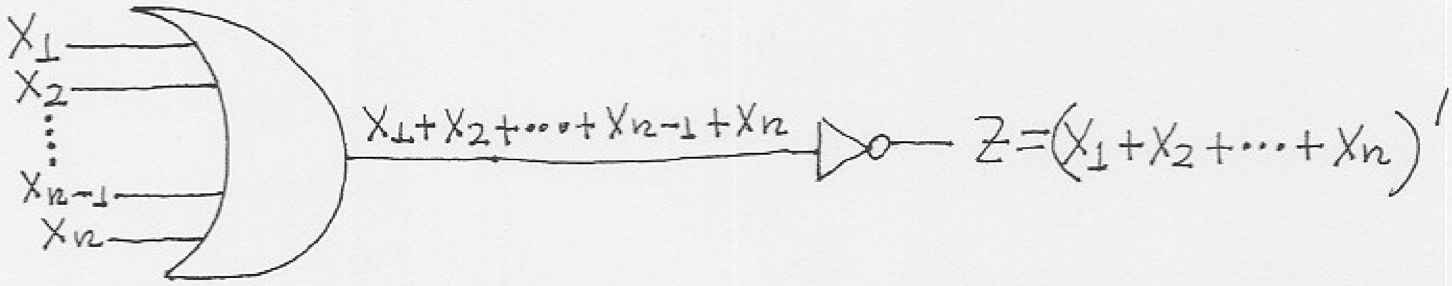


Figure 7; (OR-NOT circuit representing the function of an n-input NOR gate).

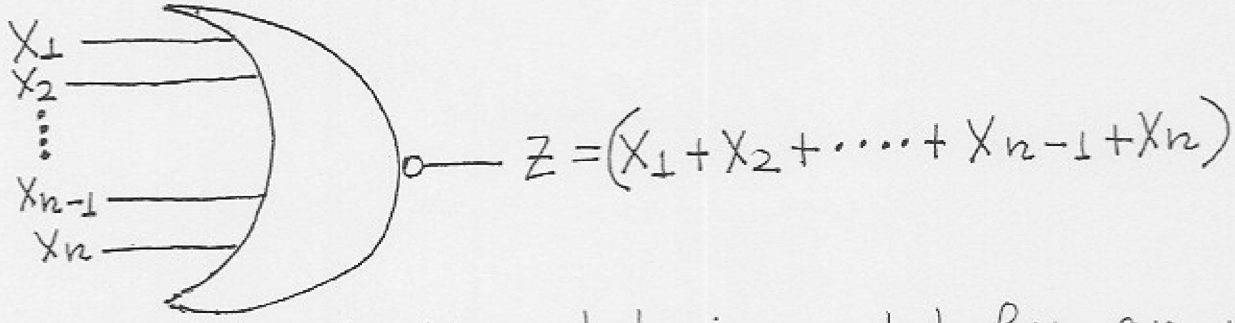


Figure 8; (official logic symbol for an n-input NOR gate).

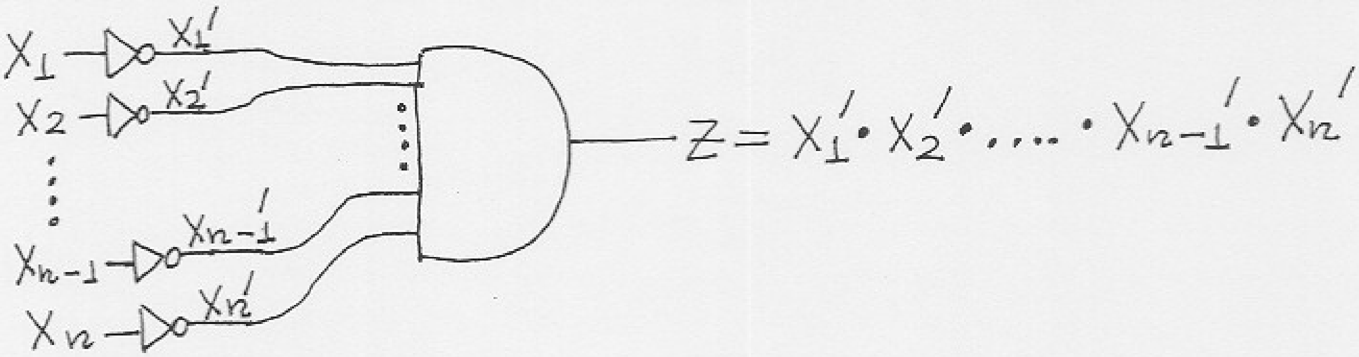

X_1
X_1'
X_2
X_2'
X_{n-1}
X_{n-1}'
X_n
X_n'
$Z = X_1' \cdot X_2' \cdot \ldots \cdot X_{n-1}' \cdot X_n'$

Figure 9; (NOT-AND circuit representing the function of an n-input NOR gate; this figure was obtained from fig. 7 by applying DeMorgan's theorem T13'. Figures 7 and 9 are equivalent)

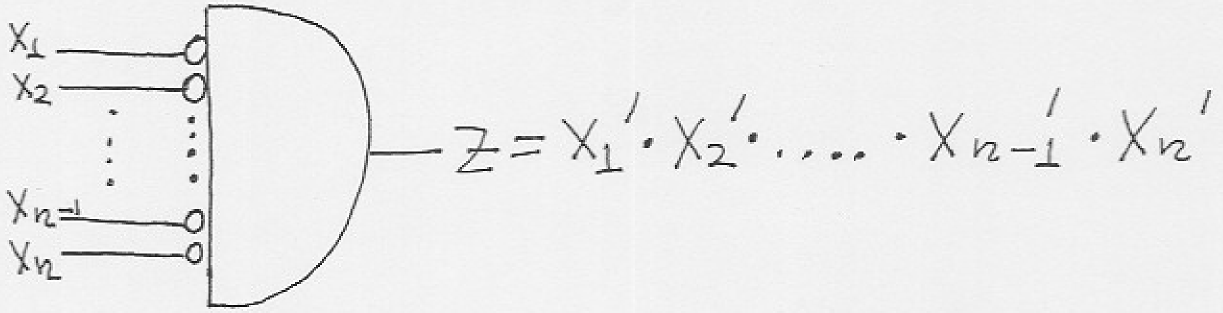

X_1
X_2
X_{n-1}
X_n
$Z = X_1' \cdot X_2' \cdot \ldots \cdot X_{n-1}' \cdot X_n'$

Figure 10; (equivalent symbol for an n-input NOR gate).

Note: When you want to represent an n-input NOR gate you should use the logic symbols of either figure 8 or figure 10.

Note: As we said at the very beginning of the semester, any digital function can be realized with only the AND, OR and NOT gates. We have already seen that. We therefore call the set of AND, OR and NOT operators functionally complete.

Statement 1: Any digital function can be realized using only NAND gates. In other words, the NAND operator is a complete operator.

Proof: All that I have to prove is that by using only NAND gates I can implement a NOT gate, an AND gate and an OR gate; (remember that with NOT, AND, OR gates any digital function can be realized). I will demonstrate this with three figures provided below.

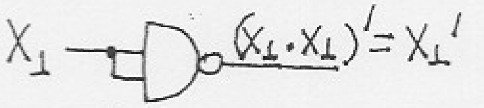


Figure 11; (realization of a NOT gate using a NAND gate).

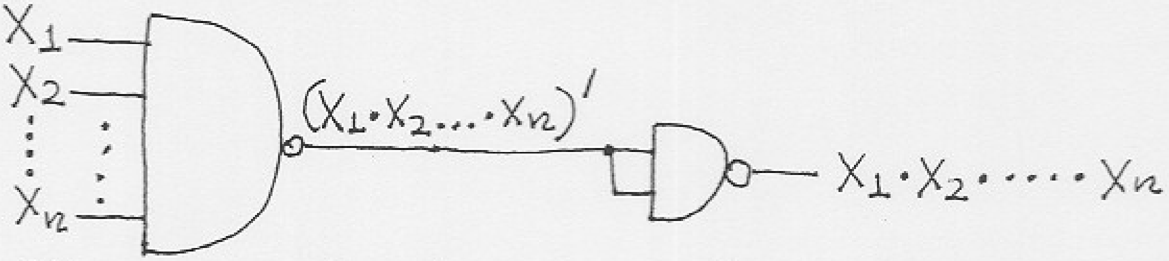


Figure 12; (realization of an AND gate using NAND gates).

→ Go to next page →

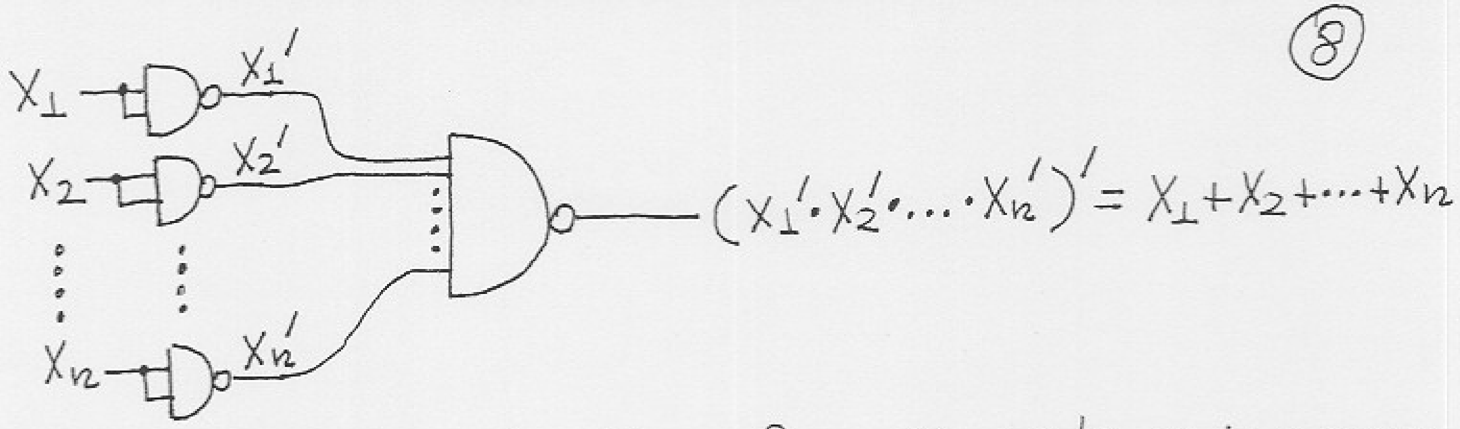

X_1
X_1'
X_2
X_2'
X_n
X_n'
$(X_1' \cdot X_2' \cdot \ldots \cdot X_n')' = X_1 + X_2 + \cdots + X_n$

Figure 13; (realization of an OR gate using NAND gates; here J also used DeMorgan's theorem).

The proof is now completed.

Statement 2: Any digital function can be realized using only NOR gates. In other words, the NOR operator is a complete operator.

Proof: Again, all that I have to prove is that by using only NOR gates I can implement a NOT gate, an OR gate and an AND gate. Again I will demonstrate this with three figures provided below.

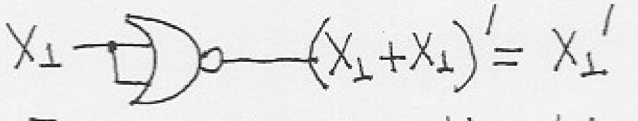


Figure 14; (realization of a NOT gate using a NOR gate).

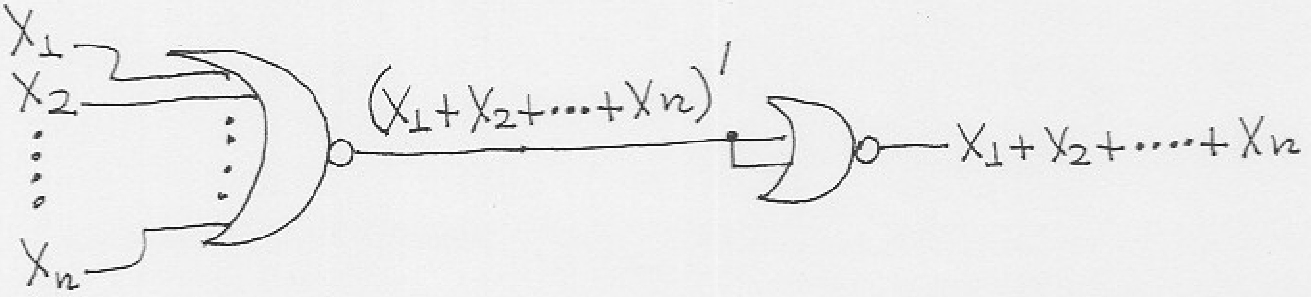


Figure 15; (realization of an OR gate using NOR gates).

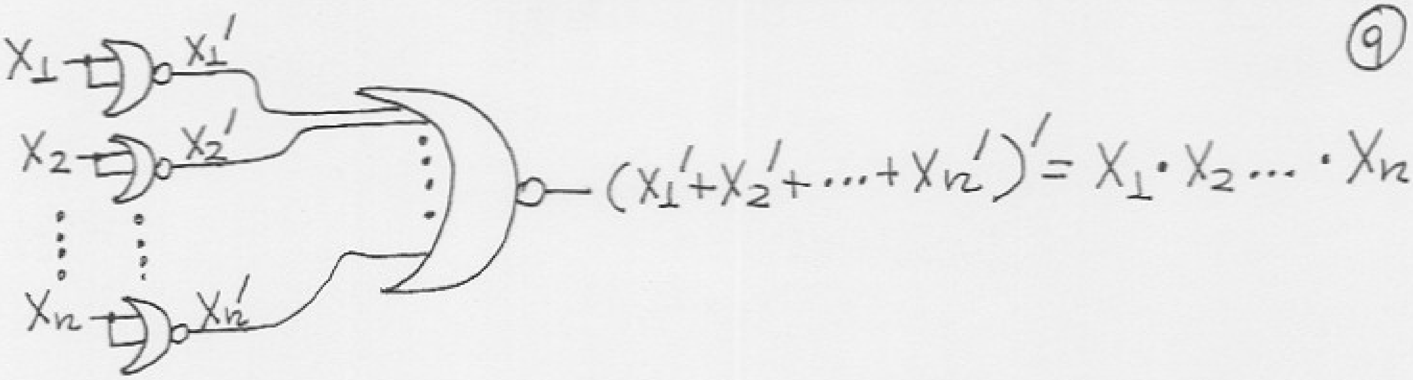


Figure 16; (realization of an AND gate using NOR gates; here I also used DeMorgan's theorem).

The proof is now completed.

Note: Logic designers frequently use NAND and NOR gates because they are generally faster and use fewer components than AND or OR gates. That is why NAND and NOR gates are important.

Note: Some equivalent logic symbols for a NOT gate, an n-input AND gate and an n-input OR gate are provided by the three figures below; (they can be used if you want to convert circuits in different equivalent alternative forms).

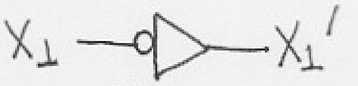


Figure 17; (equivalent logic symbol for a NOT gate).

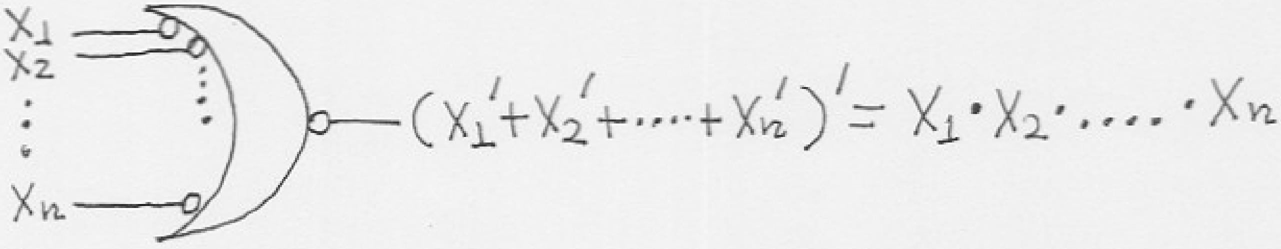
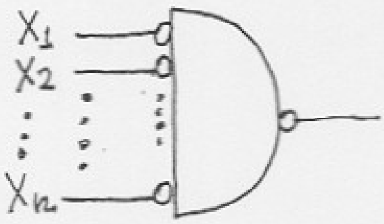

X_1
X_2
X_n
$(X_1'+X_2'+\dots+X_n')' = X_1 \cdot X_2 \cdot \dots \cdot X_n$

Figure 18; (equivalent logic symbol for an n-input AND gate; here I used DeMorgan's law).



$$(X_1' \cdot X_2' \cdot \ldots \cdot X_n')' = X_1 + X_2 + \cdots + X_n$$

Figure 19; (equivalent logic symbol for an n-input OR gate; here I used DeMorgan's law).