Appendix A Exercises

A.1 [10] $\langle$A.2$\rangle$ In addition to the basic laws we discussed in this section, there are two important theorems, called DeMorgan's theorems:

$$\overline{A+B} = \overline{A} \cdot \overline{B} \text{ and } \overline{A \cdot B} = \overline{A} + \overline{B}$$

Prove DeMorgan's theorems with a truth table of the form:

$\boldsymbol{A}$	$\boldsymbol{B}$	$\overline{\boldsymbol{A}}$	$\overline{\boldsymbol{B}}$	$\overline{\boldsymbol{A+B}}$	$\overline{\boldsymbol{A}} \cdot \overline{\boldsymbol{B}}$	$\overline{\boldsymbol{A \cdot B}}$	$\overline{\boldsymbol{A}} + \overline{\boldsymbol{B}}$
0	0	1	1	1	1	1	1
0	1	1	0	0	0	1	1
1	0	0	1	0	0	1	1
1	1	0	0	0	0	0	0

Table 1: DeMorgan's Theorems

The table above is given in the text. Note that columns $\overline{A+B}$ and $\overline{A} \cdot \overline{B}$ contain the same values and $\overline{A \cdot B}$ and columns $\overline{A} + \overline{B}$ also contain the same values, thus proving DeMorgan's Theorems.

A.3 [10] $\langle$A.2$\rangle$ Show that there are 2^n entries in a truth table for a function with n inputs.

Let boolean function f with n inputs be written as:

$$f(x_1, x_2, \ \dots \ , x_{n-1}, x_n)$$

Each input x_i can take on one of two values, 0 or 1. The total number of input combinations is given by the product of the possibilities for each of the n inputs:

$$\prod_{i=1}^{n} 2 = 2 \times 2 \times \dots \times 2 \times 2 = 2^n$$

Therefore, the number of different input combinations is 2^n, and since each row in a truth table corresponds to one unique combination of inputs, there are 2^n entries in a truth table for a function with n inputs.

Example where $n = 3$:

	$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{x_3}$
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

Table 2: Truth table with $n = 3$ inputs

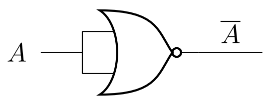
Table contains $2^3 = 8$ rows.

A.5 [15] ⟨A.2⟩ Prove that the NOR gate is universal by showing how to build the AND, OR, and NOT functions using a two-input NOR gate.

a	b	NOR
0	0	1
0	1	0
1	0	0
1	1	0

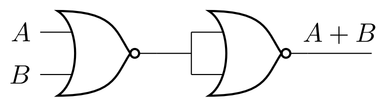
Table 3: NOR gate

NOT



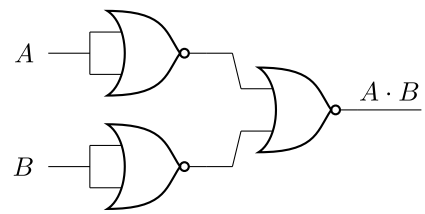
By the truth table for a NOR gate (Table 3), when both inputs A and B are 0, the output is 1, and when both inputs are 1, the output is 0. This behavior mirrors that of a NOT operation. A NOT gate can therefore be created by connecting both inputs of a NOR gate.

OR



The NOR gate produces the inverse of the OR operation. By applying a NOT operation (constructed using NOR gates) to the output of a NOR gate, the OR operation is implemented.

AND



$\overline{\overline{A}+\overline{B}} = \overline{\overline{A}} \cdot \overline{\overline{B}} = A \cdot B$
By DeMorgan's Theorem ($\overline{A} \cdot \overline{B} = \overline{A+B}$)

A.7 [10] ⟨A.2, A.3⟩ Construct the truth table for a four-input odd-parity function (see page A-65 for a description of parity).

A	B	C	D	Parity
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Table 4: Four-input odd-parity function

A.17 [5] ⟨A.2, A.3⟩ Show a truth table for a multiplexor (inputs A, B, and S; output C), using don't cares to simplify the table where possible.

A	**B**	**S**	**C**
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Table 5: 2-1 MUX

A	**B**	**S**	**C**
0	X	0	0
X	0	1	0
1	X	0	1
X	1	1	1

Table 6: 2-1 MUX with don't cares

S	**C**
0	A
1	B

Table 7: 2-1 MUX simplified