Laboratory Experiment on Combinational Logic

Introduction to Basic Combinational Logic, Boolean Algebra and the Karnaugh Map

Objective: To familiarize oneself with the application of logic gates to form simple combinational logic circuits.

Equipment required:
Logic probe
Digital trainer
Notebook installed with NI MultiSim™

ICs required:
74LS04 - NOT gate
74LS32 - OR gate
74LS08 - AND gate
74LS00 - NAND gate
74LS02 - NOR gate

Experiment A: Equivalent Circuits

Objective : To prove visually that the circuits (fig 3.1) representing the two Boolean expressions: **AB + AC** and **A.(B + C)** are identical.

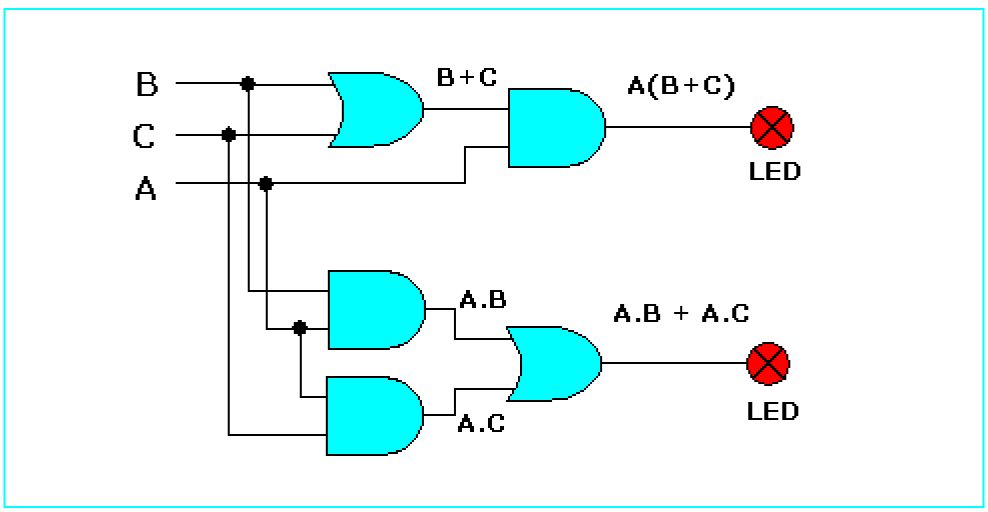
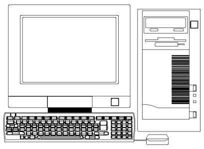


Figure 3.1



Using your Notebook, activate NI MultiSim™ and then capture the circuit as shown in figure 3.1. Supply Vcc and DGND, three SPDT switches and two probes, 3 AND gates and 2 OR gates are required.

1. To start a new circuit, launch the NI Multisim Simulation software from ***Start*** » ***Programs*** » ***National Instruments*** » ***Circuit Design Suite*** » and then select ***Multisim.***

„ Proceed to capture the circuit shown in figure 3.1. Place the required components in the drawing page as shown in figure 3.1a. For the required AND and OR gates, you may use actual IC devices instead of generic gates by accessing the **TTL/74LS** library. Select 74LS08N for the AND gates and 74LS32N for the OR gates. Complete the schematic capture and simulate your circuit by toggling the switches in accordance with the combinational sequence shown in the truth table of Table 3.1. Record the status of the probes in the output columns of the table.

”

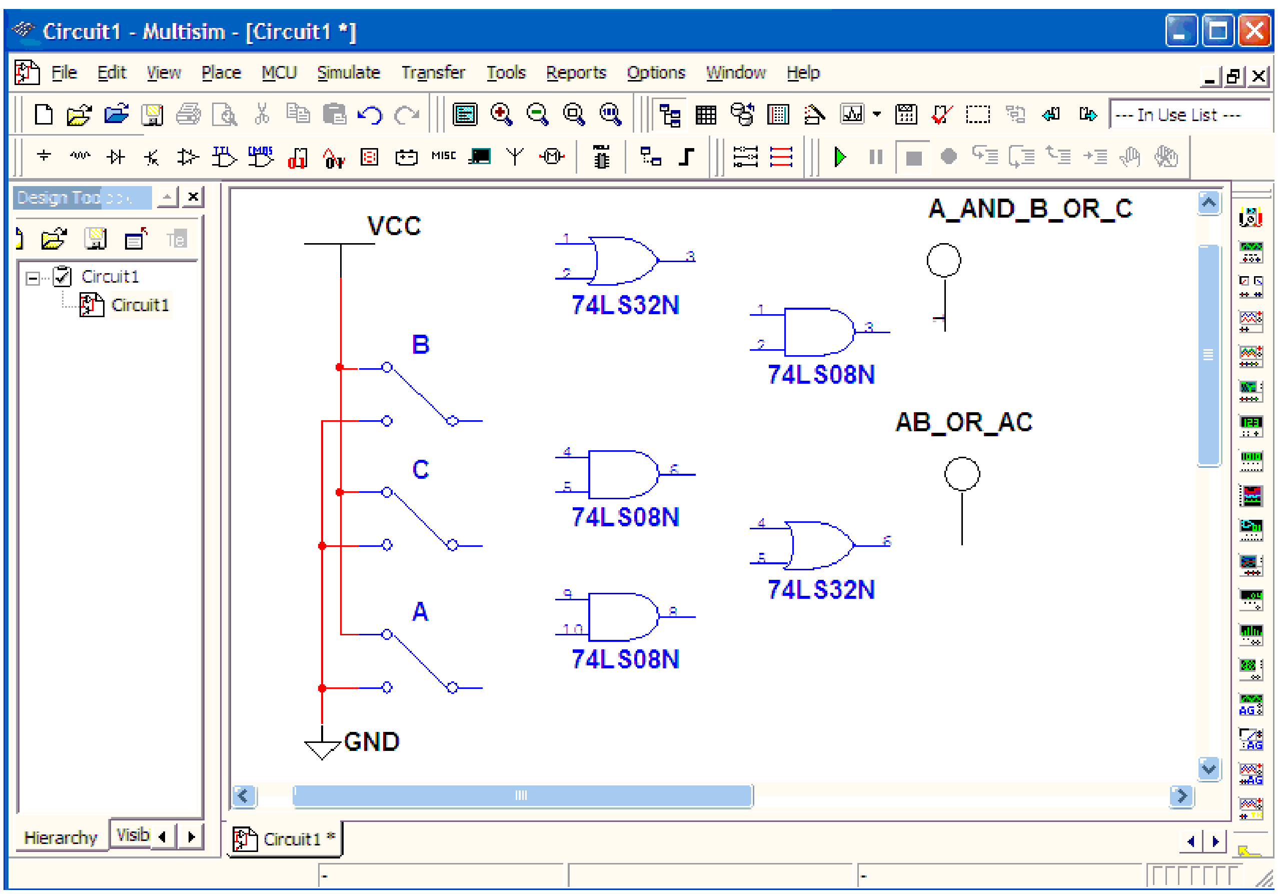


Figure 3.1a

C	B	A	AB + AC	A(B + C)
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

Table 3.1

2. Connect on the logic trainer, the circuit of Figure 3.1, together with 3 input logic switches A, B, C and, 2 LEDs to the outputs. *Login to blackboard on the Lab PC and go to the Lab Folder of the Course Documents Menu and Launch the Guide for Experiment 3 A. Two versions are provided for your convenience to help you in your circuit connections.*

Always remember to connect power supply (**Vcc** and **ground**) to **all** ICs used.

3. Set the input switches to the condition as given in first row of the table 3.1. Observe the LEDs for the output response. In the output columns of the table write 0s if the LEDs are off and 1s if the LEDs are on.

4. Set the input switches for all the input combinations listed in the table while repeating step 3.

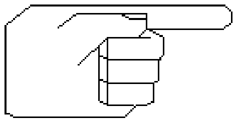
The Boolean expression AB + AC reads: - A AND B OR A AND C.

The Boolean expression A.(B + C) reads: - A AND (B OR C).

The circuits representing A(B+C) has one logic gate fewer than the circuit representing AB + AC.

Experiment B Equivalent Circuits for $\overline{A . B}$

Objective : To prove visually that the circuits representing the two Boolean expressions: $\overline{A . B} = \overline{A} + \overline{B}$ produce identical results.



Note: This experiment proves the first of De Morgan's theorems that a NAND gate function is identical to a negated-input OR function. The procedures are the same as those used for the preceding experiment.

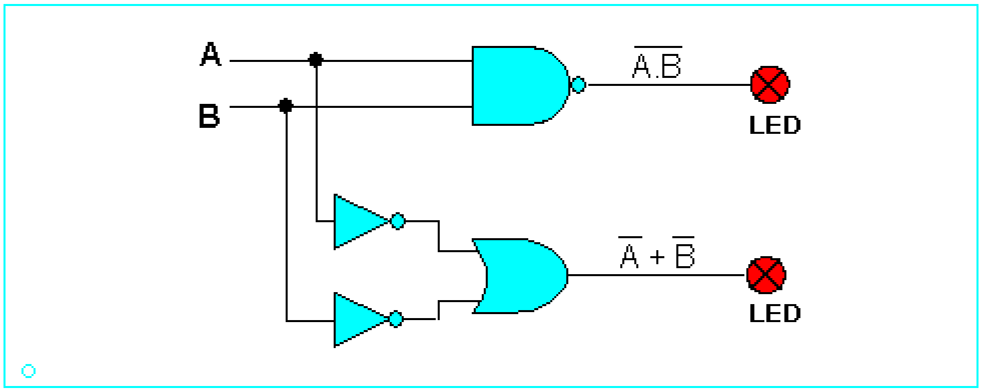
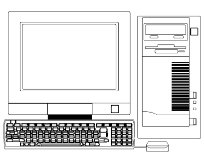


Figure 3.2



Using NI Multisim^{TM} setup a clean working window and then capture the circuit shown in figure 3.2. Simulate your circuit by toggling the logic switches in accordance with the combinational sequence of table 3.2 and hence, record the status of the probes in the output columns of this table.

Inputs		Outputs	
B	A	$\overline{A . B}$	$\overline{A} + \overline{B}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Table 3.2

2. Connect on the logic trainer, the circuits of Figure 3.2. No guidelines will be given this time as you should have an idea of how to connect your circuit. Include 2 logic switches A and B at the inputs and 2 LEDs at the outputs

3. Toggle the input switches in accordance to the logic combinations listed in table 3.2 and monitor the LEDs for the output response. In the output columns of the truth table write 0s if the LEDs are off and 1s if the LEDs are on.

4. Verify that the results obtained in step3 are the same as those obtained from circuit simulation.

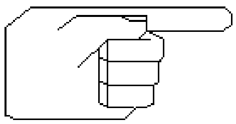
The Boolean expression $\overline{A.B}$ reads: _(A AND B) bar, or (A AND B) NOT.

The Boolean expression $\overline{A}+\overline{B}$ reads: __A bar OR_ B bar , or A NOT or__ B _NOT___.

The two logic circuits prove the ___ theorems of Boolean algebra.

Experiment C Equivalent Circuits for $\overline{A + B}$

Objective : To prove visually that the circuits representing the two Boolean expressions: $\overline{A + B} = \overline{A} . \overline{B}$ produce identical results.



Note: This experiment proves the second De Morgan's theorems that a NOR gate function is identical to a negated-input AND function. The procedures are the same as those used for the preceding experiment.

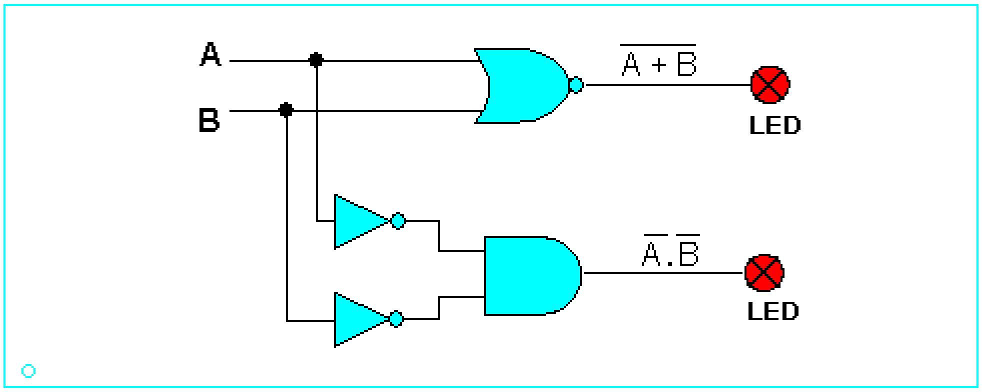
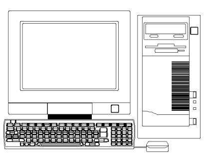


Figure 3.3



Using NI Multisim™ set up a clean working window and capture the circuit shown in figure 3.3. Simulate your circuit by toggling the logic switches in accordance with the combinational sequence of table 3.3 and hence, record the status of the probes in the output columns of this table.

Inputs		Outputs	
B	A	$\overline{A + B}$	$\overline{A} . \overline{B}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Table 3.3

2. Repeat the experiment by physically connecting the circuit on the logic trainer. Remember to also connect 2 logic switches A and B and 2 LEDs to the outputs.

3. Toggle the input switches in accordance with the logic combinations listed in Table 3.3 and observes the LEDs for the output response. In the output columns of the truth table record the responses in terms of 1s and 0s.

4. Verify that the results for both circuit simulation and the physical experiment are the same.

The Boolean expression $\overline{A + B}$ reads: ___ A OR B, or ___ OR ___ NOT.

The Boolean expression $\overline{A} . \overline{B}$ reads: ____ A AND ____ B, or A ____ AND B ____.

The two logic circuits prove the ___ theorems of Boolean algebra.

Experiment D A simple combinational logic circuit

Objective : Given the Truth Table as shown in table 3.4, derive and hence simplify the logic expression using:

- Boolean algebra
- Karnaugh Map

Inputs				Output
D	C	B	A	X
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Table 3.4

1. State the algebraic expression for the truth table shown above.

Boolean expression: $X = \overline{D}C\overline{B}\,\overline{A} +$

Double click on the equation editor object above to complete your Boolean expression.

2. Use Boolean algebra to minimize the expression derived in [step 1]

$$X = \overline{D}\,C\,\overline{B}\,\overline{A} +$$

Use the equation editor object above to complete and simplify the Boolean expression.

Minimized expression: $X =$

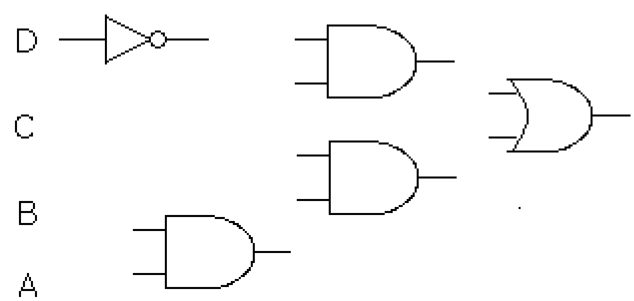
3. By using the Karnaugh map verify that the expression obtained in [step 2] uses the minimum number of terms.

	$\overline{C}\,\overline{D}$	$\overline{C}\,D$	$C\,D$	$C\,\overline{D}$
$\overline{A}\,\overline{B}$				
$\overline{A}\,B$				
$A\,B$				
$A\,\overline{B}$				

Double click on the K-Map to edit.

4. Setup a clean working window on NI Multisim^{TM} and then proceed to capture the circuit using AND, OR and NOT gates only, of the expression obtained in [steps 2 & 3]. Name the input switches using the same variable names as in the expression and then simulate your circuit by toggling the switches for all the combinations listed in table 3.4. Record the output responses and verify that the circuit implements the truth table of Table 3.4.

5. Using the simplified logic expression obtained in step 2 or step 3, construct the circuit on the trainer using AND, OR and NOT gates only. Hence, verify that the circuit implements the truth table of Table 3.4.



Double click on the image to edit

On your digital trainer you only have 2 input gates available - how can these gates be used to implement 3 input expressions which are obtained from steps 2 & 3?

Instant Review Logic gates are combined to produce logic functions that control electronic circuits. Very often an expression representing a logic function has a dual or equal expression. Similarly, complex logic functions can be reduced or simplified so that fewer logic gates are needed in the actual circuit. Truth tables can be used to verify the equality of two Boolean expressions. Notice that the dual expression simply has its AND/OR operations reversed, and in some cases 0 becomes 1, and vice versa.