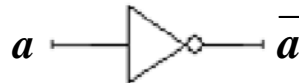


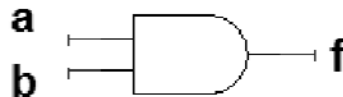
EE 1202 Experiment #3 – Introduction to Digital Circuits

1. **Introduction and Goal:** Digital circuits are the electronic circuits in all computers and microprocessors. Exercise 3 introduces students to digital circuits and Boolean algebra, the mathematics of digital circuits.
2. **Required Equipment List:**
 - Digital Prototyping Unit, Model IDL-800 (see Appendix A).
 - Electronic prototyping board (should be in IDL-800).
 - AND, OR, NOT digital circuits (see last page for schematics).
 - Connecting wire kit.
3. **Experimental Theory:** Boolean algebra has three basic functions AND, OR, and NOT. There are only two values in Boolean algebra: 1 and 0. NOT has only 1 input and 1 output. The output of NOT is the opposite of the input. If the input is 1, the output is 0; if the input is 0, the output is 1. The circuit symbol of NOT is shown below:



We say that given input a , the output $f = \bar{a}$, where \bar{a} = “not a .”

- 3.1. AND can have two or more inputs, although 2 are most common. AND = 0 unless all inputs = 1. For two inputs, the circuit symbol is:



Given inputs a, b , then $f = 1$ if and only if (iff) $a = b = 1$. Otherwise $f = 0$. The same statement is true for an AND gate of any number of inputs.

- 3.2. OR can also have any number of inputs, although 2 is again most common. The output of OR = 1 if one or more inputs of OR = 1. OR = 0 only if all inputs = 0. For two inputs, the OR symbol is:



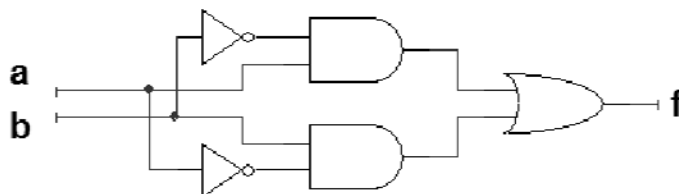
Given inputs a, b above, then $f = 0$ iff $a = b = 0$. Otherwise $f = 1$. The same statement is true for an OR gate of any number of inputs.

- 3.3. Mathematical symbols: \bar{a} = "a-NOT;" $a \cdot b$ = a AND b; $a + b$ = a OR b.
- 3.4. Computers solve problems by performing Boolean functions on binary numbers, using digital circuits. These circuits are often called gates.
- 3.5. Truth tables are used to describe complex Boolean functions, which are made up of the basic Boolean functions. Example: Given inputs a and b , assume a more complex function that = 1 when a and b are opposite (i.e., $a = 1, b = 0$ and $a = 0, b = 1$). We can put this relationship in a truth table:

a	b	f
0	0	0
0	1	1
1	0	1
1	1	0

The Truth table defines Boolean function f that fits the definition above.

- 3.6. Given a truth table that describes a function, we can create any Boolean function from the three basic functions. Any Boolean function can be designed from a group of AND gates followed by one OR gate. This is because of the unique proper of an AND circuit having only a single combinations of its inputs that will produce an output of 1.
- 3.7. Example: In the above truth table, consider the first 1 of the function f . We can create that first 1, given the inputs $a = 0$ and $b = 1$, by ANDing inverted a and non-inverted b , i.e., $\bar{a} \cdot b$. Likewise, when $a = 1$ and $b = 0$ ANDing a and inverted b together creates that function, i.e., $a \cdot \bar{b}$. Note that these two functions are 1 only for a single combination of a and b . Don't believe it? Use the truth table above to see the value of $\bar{a} \cdot b$ or $a \cdot \bar{b}$ for any other combination of variables a and b .
- 3.8. To create the full function f above, we remember that any 1 input to an OR gate produces a 1 on the output. If we then OR the two functions we have created above, we will get an output of 1 when either of them is 1. Thus we have that $f = (a \cdot \bar{b}) + (\bar{a} \cdot b)$. The circuit for such a function is shown below.



3.9. You will practice with a more complex Boolean function in Worksheet #3 and the experimental procedure, below.

- 4. Pre-Work:** Prior to the laboratory, study the outline below and the theory in section 3. Make sure that you have completed worksheet #3.
- 5. Experimental Procedure:** Refer to the equipment manual (Appendix A) to become familiar with the IDL-800. Study circuit diagrams on the last page of this exercise.

5.1. Verifying the Validity of the Basic Boolean Functions: Plug the IDL-800 prototyping system into AC power outlet and turn on bench power and the IDL-800 power switch.

5.1.1. Select a NOT chip and plug it into the IDL-800 circuit board (make sure the circuit is plugged in across a channel). Note from the circuit diagram that there are six NOT circuits on the chip, with the pin connections shown. Using wires from the connecting wire kit, connect +5V. and ground to pins 14 and 7, respectively. Wire one of the NOT circuit outputs to an LED, and its input to one of the eight switches below the circuit board. Note the input switch position. If it is down, the input is 0, so the LED should be 1 (the inverse of 0). Now turn it “on” – that is, to 1. The LED should go out. Try this several times to verify NOT operation.

5.1.2. Turn off power, remove NOT chip, and replace with an AND chip (power connections remain the same). Based on the AND chip circuit diagram, select one AND gate and connect two of the switches to the inputs and the output to an LED. Turn on power and try various switch combinations. The LED will light up only when both inputs are 1. Experiment until you are satisfied with AND gate operation.

5.1.3. Now plug in an OR chip. Select one OR circuit and connect inputs to switches and the output to an LED. Turn on power and try various input combinations. The output LED should light whenever one or more inputs = 1. Turn off power.

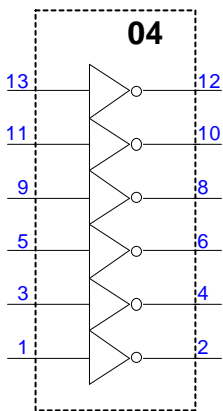
5.2. Creating a Circuit to Match a Truth Table: Study the truth table below. It defines another function X of the variables a , b , and c . Given your three circuit chips, NOT, AND, and OR, build that circuit.

Remember, using inverters plus AND gates, you can define a function that is a 1 for a specific combinations of a , b , and c . Then, you can OR those functions together to get the function X that is 1 for all those combinations. Note that even though you need to AND three things together, you can do it with 2-input AND gates: $a \cdot b \cdot c = (a \cdot b) \cdot c$.

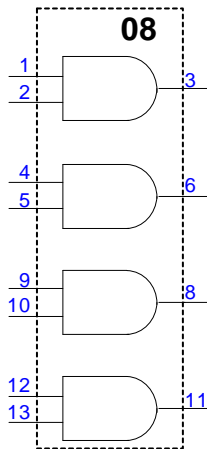
<i>a</i>	<i>b</i>	<i>c</i>	<i>X</i>
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

- 5.3. Demonstration of Working Circuit:** Wire the three inputs to three switches, and the output to an LED. The output LED should only be 1 if inputs *b* and *c* are 1 and *a* is 0, or if *a* and *c* are 1 and *b* is 0. Fill out the data sheet truth table based on your observations. Does it match the one above?
- 5.4. Design problem:** A chemical process is monitored by three digital trouble sensors: Temperature (*x*), pressure (*y*), and heater power (*z*). If temperature is at or below normal, *x* is 0; if too high, it is 1. If pressure is normal or below, *y* is 0; if high, *y* = 1. Heater power off corresponds to *z* = 0; heater on sets *z* to 1. There are two dangerous conditions: (1) Temperature (*x*) and pressure (*y*) high and heater power (*z*) on (which will lead to higher temperature and pressure and could cause an explosion), and (2) temperature (*x*) and pressure (*y*) normal but power (*z*) off (which means that the process may be approaching too low an operating point, and thus ruin the batch of chemicals in process).
- 5.4.1. Design a circuit as follows:** If condition (1) exists, your circuit should generate a logic 1 (a signal to turn off the power). If condition (2) exists, the circuit must also generate a logic 1 on a different line (a signal to turn on heater power).
- 5.4.2. Use the truth tables in your data sheet to develop a Boolean expression that satisfies each of the above constraints. Then build the circuit and verify that it works as described. When you have completed exercise 5.4, the lab is complete.**
- 6. Laboratory Area Cleanup:** Replace parts in the parts kit and the connecting wires in the wire kit, and return kits to the cabinet. Make sure that your work area is clean.
- 7. Writing the Laboratory Report:** In your report, do the following:
- 7.1. Complete the truth tables in #1 on the data sheet for NOT, AND, and**

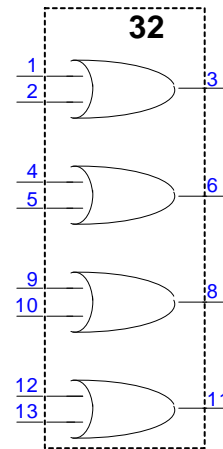
- OR**, based on your observations from 5.1, above.
- 7.2. Draw the Boolean schematic of your circuit from 5.2, given the inputs and output lines shown, using the symbols for AND, OR, and NOT.
 - 7.3. Complete the data sheet truth tables for the design problem in 5.4, and draw the two circuits that generate the required signals in 5.4 that you built.
 - 7.4. Discuss any difficulties that you had in doing this project, including working with the new notation and functions in digital logic.



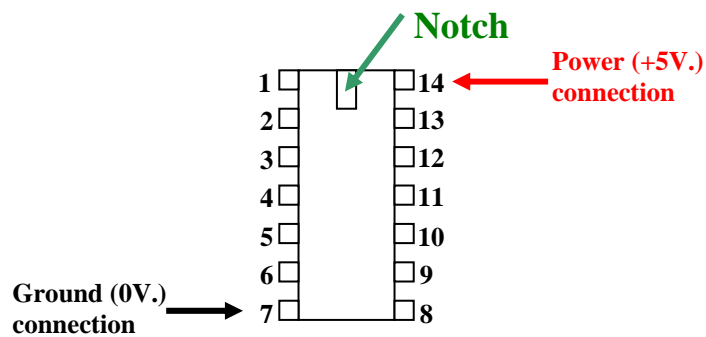
SN 74LS04 Hex inverter gate



SN 74LS08 Quad 2-input AND gate



SN 74LS32 Quad 2-input OR gate



74 LS XXX Chip Outline

Experiment #3 Data Sheet

1. AND, OR, and NOT – your observations. Fill in truth tables:

NOT	
<i>x</i>	<i>f</i>
0	
1	

AND		
<i>x</i>	<i>y</i>	<i>f</i>
0	0	
0	1	
1	0	
1	1	

OR		
<i>x</i>	<i>y</i>	<i>f</i>
0	0	
0	1	
1	0	
1	1	

2. Truth Table for Circuit in 5.2:
Show circuit below.

<i>a</i>	<i>b</i>	<i>c</i>	<i>X</i>
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	0

3. Truth Tables for Design Problem:

<i>x</i>	<i>y</i>	<i>z</i>	<i>Signal 1</i>
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

<i>x</i>	<i>y</i>	<i>z</i>	<i>Signal 2</i>
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Experiment #3 Worksheet

Note: Refer to Experiment #3 to answer the questions below.

1. Fill in the truth table at the right, for the Boolean function AND, using the definition in Experiment #3.

AND		
<i>a</i>	<i>b</i>	<i>f</i>
0	0	
0	1	
1	0	
1	1	

2. Fill in the truth table at the right, for the Boolean function OR, using the definition in Experiment #3.

OR		
<i>a</i>	<i>b</i>	<i>f</i>
0	0	
0	1	
1	0	
1	1	

3. A Boolean function f of the variables x , y , and z is level 1 when x and y are 0 and z is 1, when x and y are 1 and z is 1, and when x is 0 and y and z are 1, and 0 otherwise. Fill in the truth table with the correct values for f , write down the Boolean expression for the function, and then draw the circuit that fulfills the function. Remember that AND and OR gates can have any number of inputs.

**Practice Truth Table for
Boolean Function in #3.**

<i>x</i>	<i>y</i>	<i>z</i>	<i>f</i>
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	