LAB 3-Basic Gates:

Part 1- AND, OR, NOT Gates

OBJECTIVES:

- Examine the three logical operations
- Examine basic digital logic gates
- Examine the operation of a standard small-scale integrated TTL device

MATERIALS:

- 74LS08 AND gate IC in a 14-pin DIP
- 74LS32 OR gate IC in a 14-pin DIP
- 74LS04 Hex Inverting (NOT) Gates IC in a 14 pin DIP
- Power supply
- LED

DISCUSSION:

Logical Operations

The operation of computers and all other digital equipment is based on a few logical concepts and operations. They are so simple that it is hard to believe so much can be derived from them. These concepts include truth-values, binary numbers, and Boolean algebra. The operations include the following: AND, OR, and NOT.

Truth-Values & Binary Numbers

Many statements can be evaluated as being either true or false. Some examples are:

- statement A: John is six feet tall.
- statement B: Joe is six feet tall.
- statement C: John and Joe are both six feet tall.

If statements A and B are true, then C must be true. Obviously, if either A or B were false, then C would have to be false. Let the digit 1 represent "is true" and let the digit 0 represent "is false". Then we can summarize the statement possibilities in a table:

A	B		C
0	0		0
0	1		0
1	0		0
1	1		1

Table 1-This is a truth table

Digital logic circuitry produces two output levels that can be referred to as HIGH and LOW, TRUE and FALSE (like in the previous example), ON and OFF, or simply 1 and 0. Because we are dealing with things that can be true or false only, we represent them using binary numbers, which can be 1 or 0 only. Inputs and outputs of gates are represented by voltages: +5 Volts for a logical 1 and 0 Volts (ground) for a logical 0.

AND Operation & Gate

The truth table above represents the logical AND operation. We can write the AND operation as an Boolean equation:

C = A•B or C = AB

where the equal sign (=) represents the words "is true if" and the dot (•) represents the word and. So the equation means:

C *is true if* A is true AND B is true.
Otherwise, C is false.

Look at the electrical circuit below which is composed of a battery, two switches, and a light bulb. For the light bulb (L) to light, switch A and switch B both must be closed. Using 1 to mean "on" and 0 to mean "off", we can write the following Boolean equation **L = A•B**, and the truth table would be the same as Table 1.

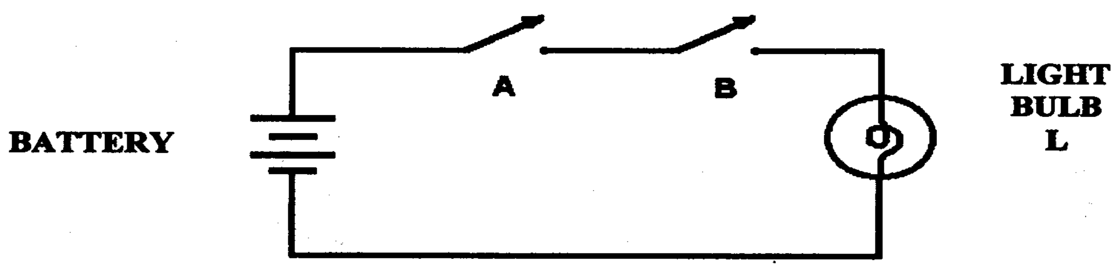
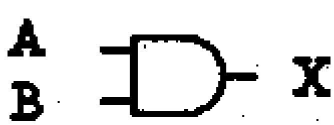


Figure 1-Electrical AND circuit

An AND gate is a device whose output is 1 if and only if all its inputs are 1. It can have multiple inputs, but has only one output.

AND Gate Symbol



AND Gate Truth Table

A	B		X
0	0		0
0	1		0
1	0		0
1	1		1

OR Operation & Gate

We can write the OR operation as an Boolean equation:

X = A+B

where the equal sign (=) represents the words "is true if" and the plus sign (+) represents the word or. So the equation means:

X *is true if* A is true OR B is true.
Otherwise, X is false.

Now look at the circuit below in Figure 2. In this case, in order for the light bulb to be lit either switch A *or* switch B must be closed, or both can be close.

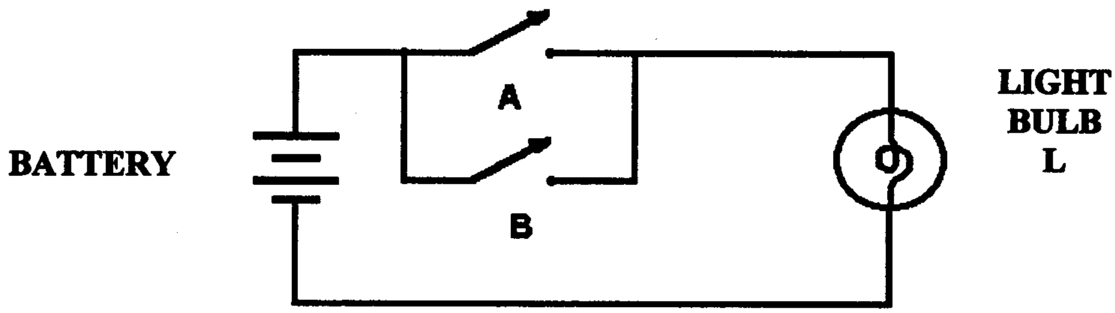
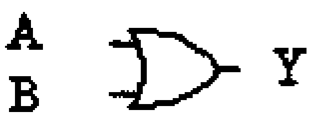


Figure 2-Electrical OR circuit

An OR gate is a device whose output is 1 if at least one of its inputs is 1. Therefore, the output of an OR gate is 0 only if all of its inputs are 0. The OR gate can have multiple inputs, but has only one output.

OR Gate Symbol



OR Gate Truth Table

A	B		Y
0	0		0
0	1		1
1	0		1
1	1		1

NOT Operation & Gate

The last basic operation is NOT, which is also called logical inversion. NOT means that one thing is logically the opposite of the other, for example A=1 and X=0. We can write this as a Boolean equation:

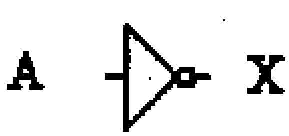
$$X = \overline{A} \text{ or } X = A'$$

where the bar above (and apostrophe beside) **A** designates the complement of **A**. The statement should be read as

X is equal to the complement of A, or
X *is NOT* equal to A.

A NOT, or inverter, gate is a device whose output is 1 if its input is 0, and vice versa.

Inverter (NOT) Gate Symbol



Inverter (NOT) Gate Truth Table

A		X
0		1
1		0

PROCEDURE:

1. Insert the 74LS08 chip onto the breadboard so that it straddles the center line.
2. Using the pin layout in Figure 3, insert wire at the inputs and a LED at the output of one AND gate.

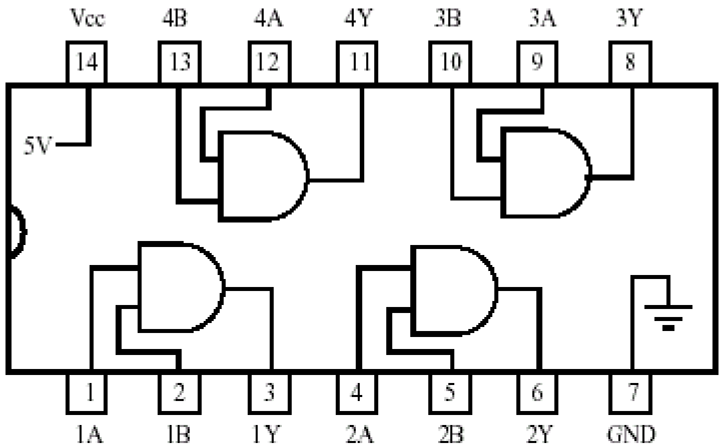


Figure 3-74LS08 Quadruple 2-Input AND Gate

3. Complete the truth table below by connecting the two inputs to logic 0 (ground) or logic 1 (+5 Volts). A lit LED represents a logic 1 and an unlit LED represents a logic 0.

Input 1	Input 2	LED	Voltage Output
0	0		
0	1		
1	0		
1	1		

Do your results verify an AND operation?

4. Insert the 74LS32 chip onto the breadboard so that it straddles the center line.
5. Using the pin layout in Figure 4, insert wire to the inputs and a LED at the output of one OR gate.

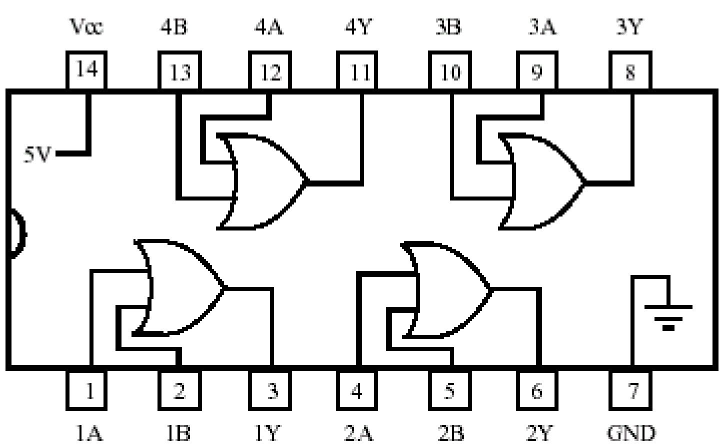


Figure 4-74LS32 Quadruple 2-Input OR Gate

6. Complete the truth table below by connecting the two inputs to logic 0 (ground) or logic 1 (+5 Volts).

Input 1	Input 2	LED	Voltage Output
0	0		
0	1		
1	0		
1	1		

Do your results verify an OR operation?

7. Insert the 74LS04 chip onto the breadboard so that it straddles the center line.
8. Using the pin layout in Figure 5, insert wire to the input and a LED at the output of one inverter (NOT) gate.

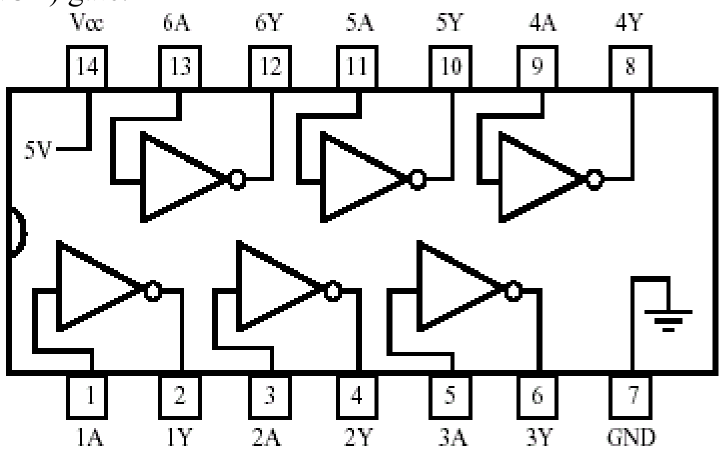


Figure 5-74LS04 Hex Inverter

9. Complete the truth table below by connecting the input to logic 0 (ground) or logic 1 (+5 Volts).

Input	LED	Voltage Output
0		
1		

Do your results verify an NOT operation?

Part 2- NAND, NOR, XOR, XNOR Gates

OBJECTIVES:

- Examine common combinations of the three logical operations
- Examine basic digital logic gates
- Examine the operation of a standard small-scale integrated TTL device

MATERIALS:

- 74LS00 NAND gate IC in a 14-pin DIP
- 74LS02 NOR gate IC in a 14-pin DIP
- 74LS86 XOR gate IC in a 14 pin DIP
- 74LS04 Hex Inverting (NOT) Gates IC in a 14 pin DIP
- Power supply
- LED

DISCUSSION:

Every possible logical operation can be performed using a combination of the AND, OR, and NOT functions. The digital gates that you will study from this point on are just combinations of these three logical functions. However, several combinations are so common that they are also considered basic gates: NAND, NOR, XOR, and XNOR.

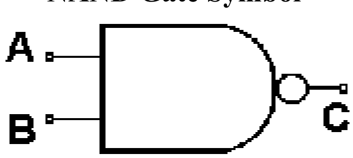
NAND OPERATION & GATE

The NAND function is the complement of the AND operation, and can be written as a Boolean equation:

$$C = \overline{A \bullet B} \text{ or } C = \overline{AB} \text{ or } C = (AB)'$$

The NAND gate is a combination of an AND gate followed by an inverter (NOT gate). Its logic symbol consists of an AND gate with a bubble at the output. This bubble implies inversion. The output of a NAND gate is 1 if at least one of its inputs is 0. Therefore, the output of a NAND gate is 0 only if all of its inputs are 1. The NAND gate can have multiple inputs, but has only one output.

NAND Gate Symbol



NAND Gate Truth Table

A	B		C
0	0		1
0	1		1
1	0		1
1	1		0

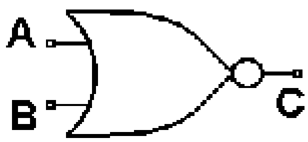
NOR OPERATION & GATE

The NOR function is the complement of the OR operation, and can be written as a Boolean equation:

$$\mathbf{C = \overline{A+B} \text{ or } C = (A+B)'}$$

The NOR gate is a combination of an OR gate followed by an inverter (NOT gate). Its logic symbol consists of an OR gate with a bubble at the output. This bubble implies inversion. The output of a NOR gate is 0 if at least one of its inputs is 1. Therefore, the output of a NOR gate is 1 only if all of its inputs are 0. The NOR gate can have multiple inputs, but has only one output.

NOR Gate Symbol



NOR Gate Truth Table

A	B		C
0	0		1
0	1		0
1	0		0
1	1		0

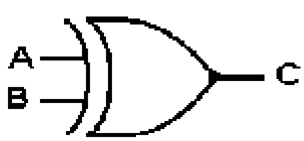
XOR OPERATION & GATE

The exclusive-OR, or XOR, function is a variation of the OR operation, and can be written as a Boolean equation:

$$\mathbf{C = A \oplus B}$$

The output of a XOR gate is 1 if **exactly** one of its inputs is a 1. Therefore, the output of a XOR gate is 0 only if both of its inputs are 0 or if both of its inputs are 1. Unlike the previous gates, the XOR gate can have only two inputs, and has only one output.

XOR Gate Symbol



XOR Gate Truth Table

A	B		C
0	0		0
0	1		1
1	0		1
1	1		0

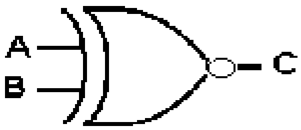
XNOR OPERATION & GATE

The exclusive-NOR (XNOR), or equivalence, function is the complement of the XOR operation, and can be written as a Boolean equation:

$$\mathbf{C = (A \oplus B)'}$$

The output of a XNOR gate is 0 if **exactly** one of its inputs is a 1. Therefore, the output of a XNOR gate is 1 **only** if both of its inputs are 0 or if both of its inputs are 1. Unlike the previous gates, the XNOR gate can have only two inputs, and has only one output.

XNOR Gate Symbol



XNOR Gate Truth Table

A	B		C
0	0		1
0	1		0
1	0		0
1	1		1

PROCEDURE:

1. Insert the 74LS00 chip onto the breadboard so that it straddles the centerline.
2. Using the pin layout in Figure 6, insert a wire to the inputs and a LED at the output of one NAND gate.

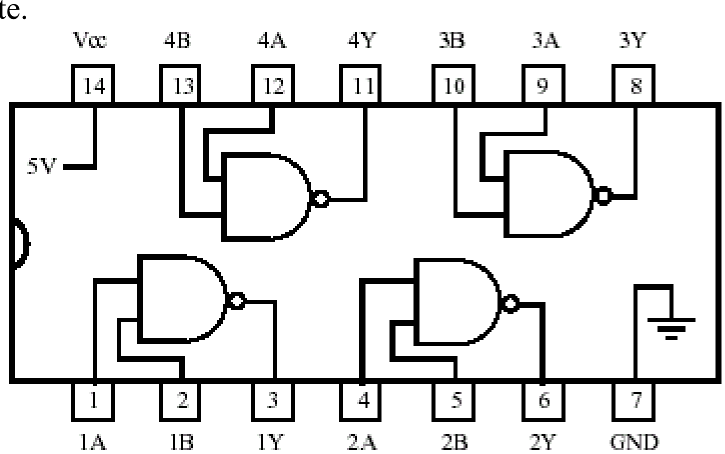


Figure 6-74LS00 Quadruple 2-Input NAND Gate

3. Complete the truth table below by connecting the two inputs to logic 0 (ground) or logic 1 (+5 Volts).

Input 1	**Input 2**	**LED**	**Voltage Output**
0	0		
0	1		
1	0		
1	1		

Do your results verify an NAND operation?

4. Insert the 74LS02 chip onto the breadboard so that it straddles the centerline.
5. Using the pin layout in Figure 6, insert a wire to the inputs and a LED at the output of one NOR gate.

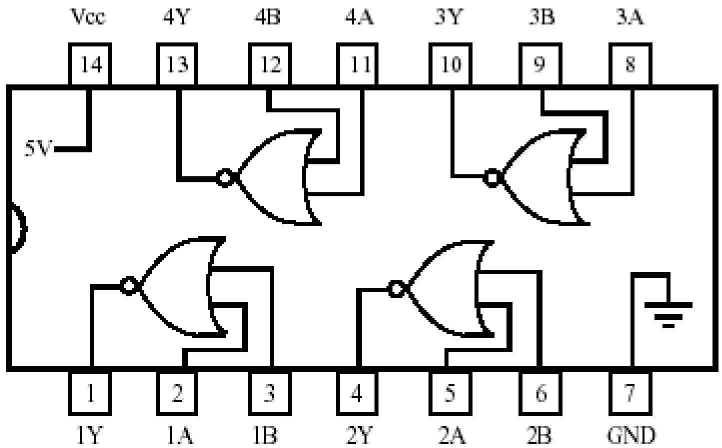


Figure 6-74LS02 Quadruple 2-Input NOR Gate

6. Complete the truth table below by connecting the two inputs to logic 0 (ground) or logic 1 (+5 Volts).

Input 1	Input 2	LED	Voltage Output
0	0		
0	1		
1	0		
1	1		

Do your results verify an NOR operation?

7. Insert the 74LS86 chip onto the breadboard so that it straddles the centerline.
8. Using the pin layout in Figure 7, insert a wire to the inputs and a LED at the output of one XOR gate.

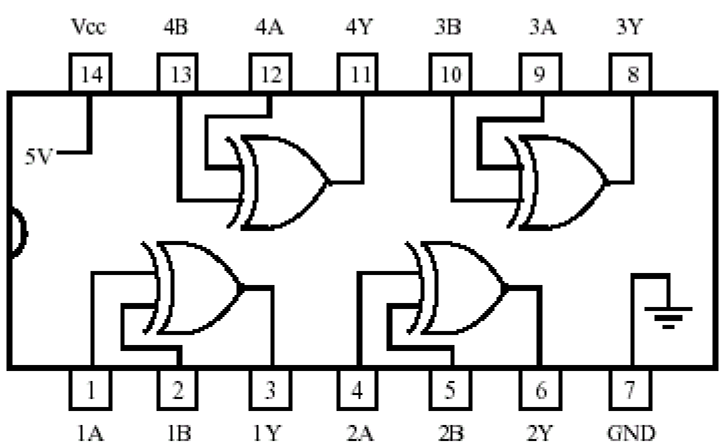


Figure 7-74LS86 Quadruple XOR Gate

9. Complete the truth table below by connecting the input to logic 0 (ground) or logic 1 (+5 Volts).

Input 1	Input 2	LED	Voltage Output
0	0		
0	1		
1	0		
1	1		

Do your results verify an XOR operation?

10. Create an XNOR gate by inverting the output of one of the XOR gates using an inverter from the 74LS04 chip.
11. Complete the truth table below by connecting the input to logic 0 (ground) or logic 1 (+5 Volts).

Input 1	Input 2	LED	Voltage Output
0	0		
0	1		
1	0		
1	1		

Do your results verify an XNOR operation?