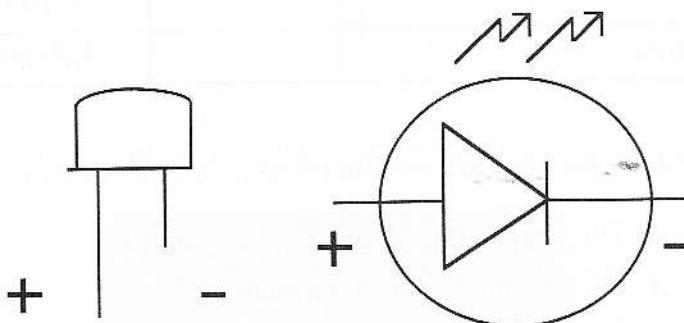


# Appendices

## Appendix A: Electronic Components

### LED

#### Physical Diagram



#### Electronic Symbol

An infrared emitter LED is represented by the same symbol as a standard LED because it still emits light. IR light, however, has a wavelength that the human eye cannot detect.

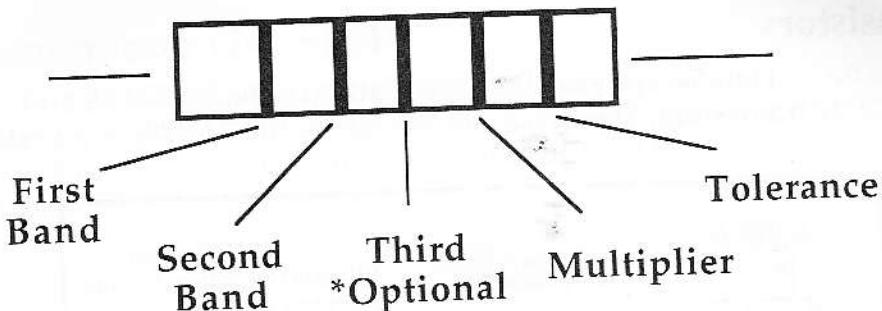
# Resistors

## Resistor Colour Code

	Band 1	Band 2	Band 3
Black	0	0	1
Brown	1	1	10
Red	2	2	100
Orange	3	3	1000
Yellow	4	4	10,000
Green	5	5	100,000
Blue	6	6	1,000,000
Violet	7	7	10,000,000
Gray	8	8	100,000,000
White	9	9	1,000,000,000

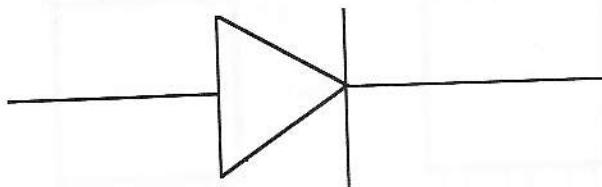
The colour bands indicate the resistance of a resistor.

- The first band represents the first digit.
- The second band is the second digit.
- This two digit number is multiplied by the value of the third band.
- The fourth band (if present) indicates the accuracy of the resistor. Gold is + or - 5% and silver is + or - 10%.

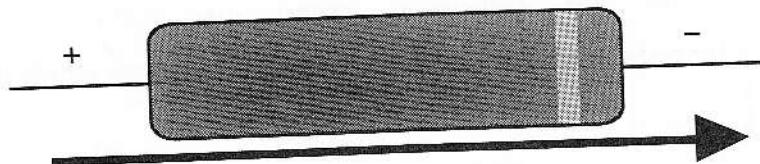


## Diodes

This is a schematic representation of a diode. Current will only flow in the direction of the arrow (left to right).

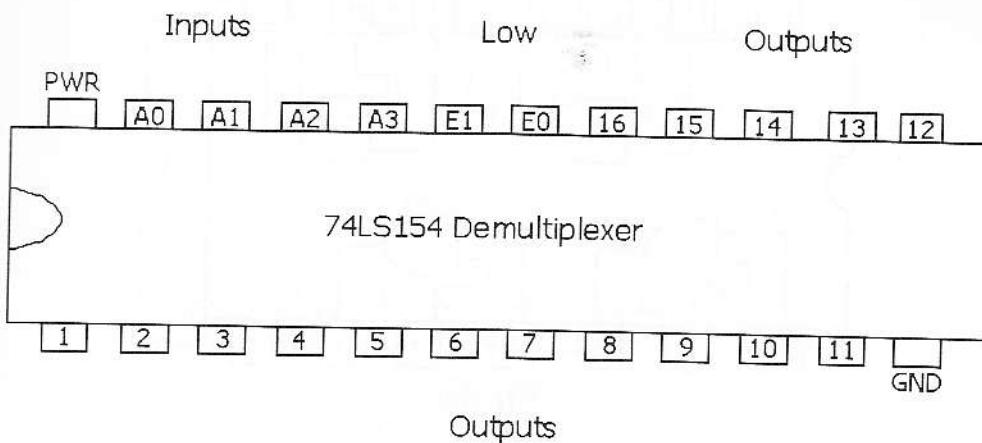


This is a physical representation of a diode.



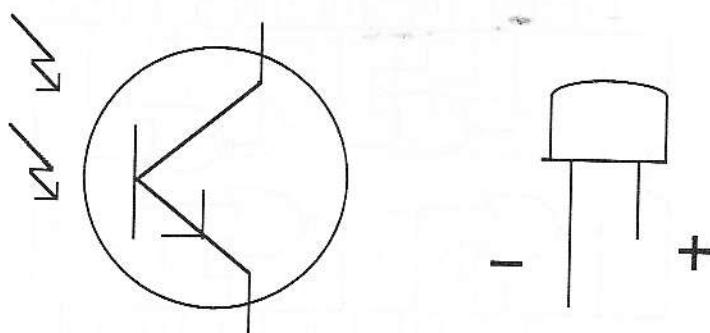
## Demultiplexer (74LS154)

An integrated circuit that accepts four input values (on A0 to A3) and activates 16 individual output pins (pins numbered 1 to 16). E0 and E1 are the enabling pins and must be held low in order to activate the chip.



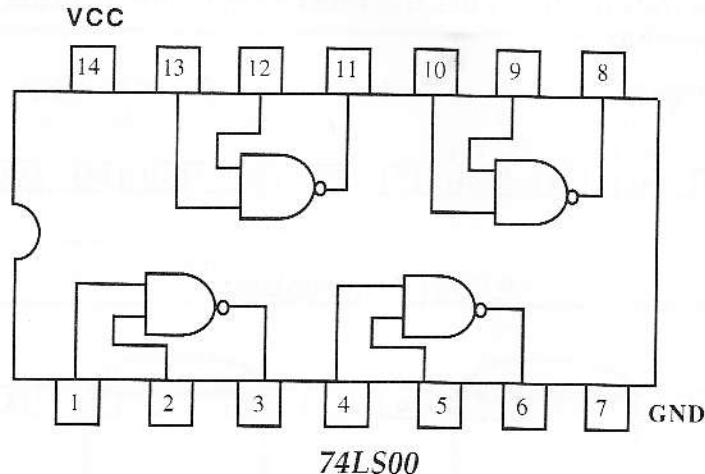
## Opto-Transistor (receiver)

The receiver is a special type of transistor that allows current to flow through it when it senses light. It acts like a switch.



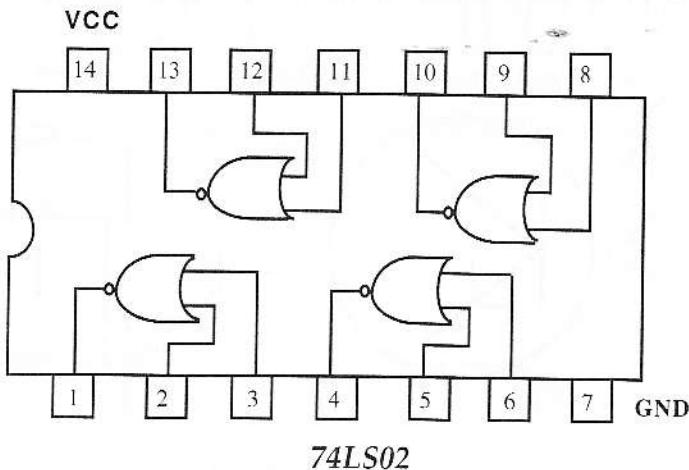
## NAND

Here is the chip schematic for the 74LS00 (NAND) chip.



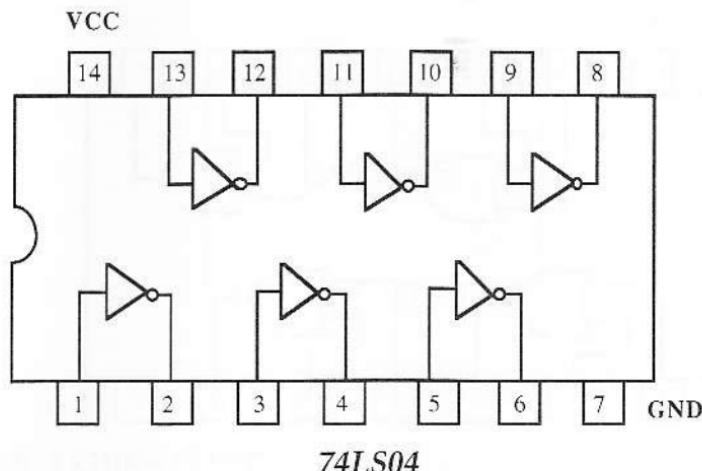
## NOR

Here is the chip schematic for the 74LS02 (NOR) chip.



## NOT

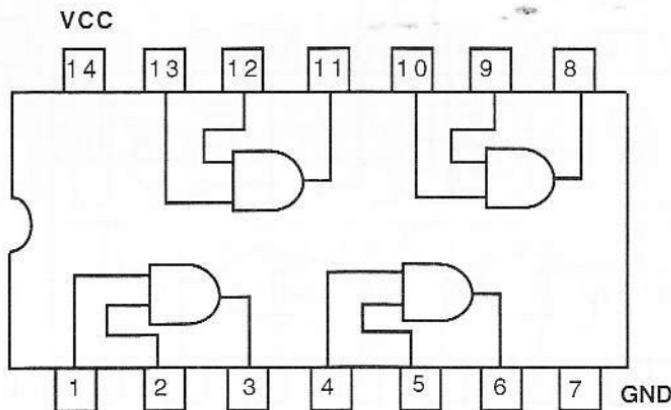
Here is the chip schematic for the 74LS04 (NOT) chip.



74LS04

## AND

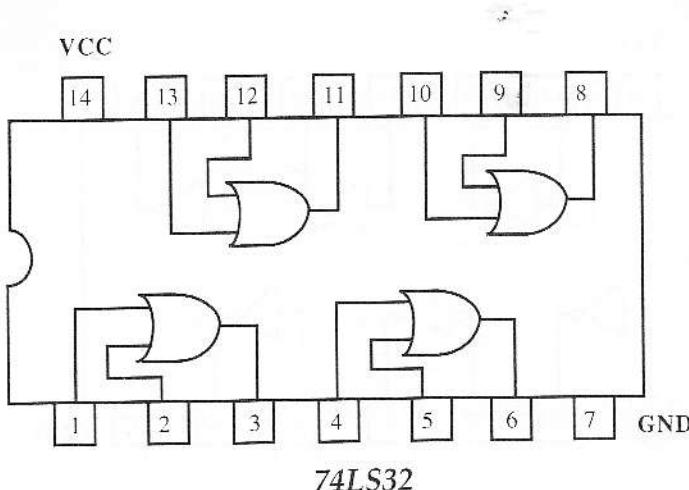
Here is the chip schematic for the 74LS08 (AND) chip.



74LS08

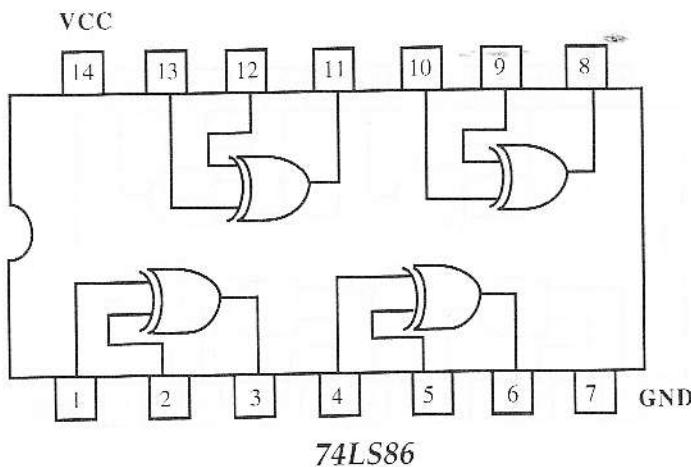
## OR

Here is the chip schematic for the 74LS32 (OR) chip.



## EOR

Here is the chip schematic for the 74LS86 (EOR) chip.



## 4.10.2 Associative Law Activity # 13

The associative law in Boolean algebra governs the grouping of inputs.

### Purpose:

To demonstrate the associative law of Boolean algebra.

### Theory:

In mathematics the associative law under addition is  $(A+B)+C=A+(B+C)$ . The Boolean notation is the same. The associative law permits the grouping of inputs.

### Procedure:

Use the following diagrams to wire the circuits. Connect pin 14 to a +5V power source and pin 7 to ground.

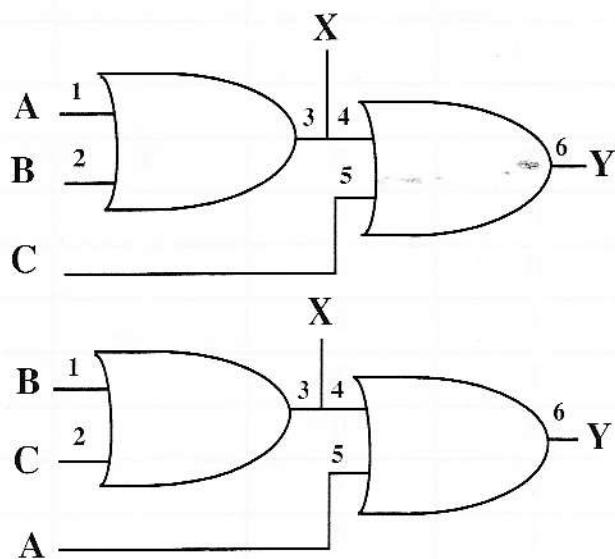


Figure 4.17

## 4.10 Boolean Algebra Laws

There are a number of laws in Boolean algebra which are used to simplify circuit design and development. These laws include:

- commutative,
- associative,
- distributive, and
- DeMorgan's.

### 4.10.1 Commutative Law - Activity # 12

The commutative law in Boolean algebra governs the order in which inputs may be carried out.

#### Purpose:

To demonstrate the commutative law of Boolean algebra.

#### Theory:

In mathematics the commutative law for addition is  $x + y = y + x$  and for multiplication is  $xy = yx$ . Boolean algebra borrowed the symbol to add and called it OR and borrowed the symbol to multiply and called it AND.

**Example 1:**  $3 + 5 = 5 + 3$

**Example 2:**  $7 \bullet 6 = 6 \bullet 7$

Both examples demonstrate the commutative law in mathematics. Example 1 uses the operation of addition and example two uses the operation of multiplication.

The symbol + in mathematics is now an OR gate in Boolean algebra and the multiplication dot is an AND gate in Boolean algebra.

The inputs A and B to an OR gate and an AND gate are commutative just as adding and multiplication are commutative in mathematics.

### Procedure:

Use the following diagram to wire the circuit. Connect pin 14 to a +5V power source and pin 7 to ground.

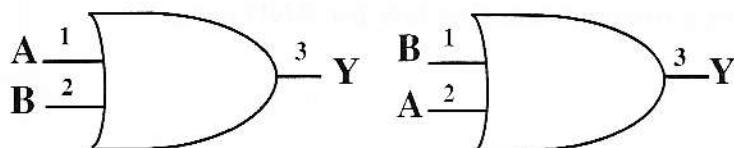


Figure 4.16

### Results:

Complete these Truth Tables for the above circuits.

B	A	Y
0	0	
0	1	
1	0	
1	1	

A	B	Y
0	0	
0	1	
1	0	
1	1	

### Conclusion:

Compare the results at Y for both Truth Tables. State the commutative law for OR gates.

**Questions:**

1. Draw a circuit diagram for

$$Y = A \bullet B$$

$$Y = B \bullet A$$

2. Wire the two circuits for the Boolean equations in Question 1.
3. Draw Truth Tables for the circuits in Question 2.
4. Compare the results at Y.
5. State the commutative law for AND gates.

#### 4.10.4 De Morgan's Law - Activity # 15

De Morgan's law governs which of the fundamental gates can be exchanged.

##### Purpose:

To verify De Morgan's law:

$$\overline{A} + \overline{B} = \overline{A \bullet B}$$

##### Theory:

AND and OR gates can be related using NOT gates. The Boolean equation

$$\overline{A} + \overline{B}$$

has two inputs: A and B. The two inputs are inverted before entering the OR (+) gate. The Boolean equation

$$\overline{A \bullet B}$$

is an AND gate that has the output inverted (that is a NAND gate). The two Boolean equations are equal.

##### Diagram:

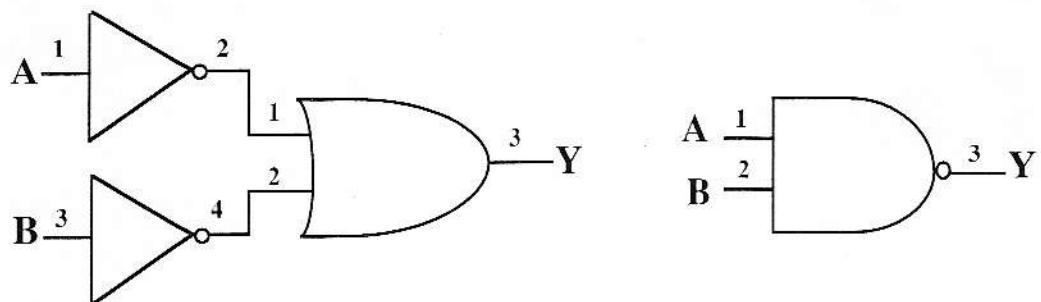


Figure 4.19

**Results:**

Complete these Truth Tables for the two expressions above.

A	B	Y
0	0	
0	1	
1	0	
1	1	

A	B	Y
0	0	
0	1	
1	0	
1	1	

**Conclusion:**

What is the relationship between the two Y values in the above Truth Tables?

State De Morgan's law.

**Questions:**

1. Draw a circuit diagram for  $\overline{A} \bullet \overline{B}$  and  $\overline{A + B}$ .
2. Wire the two Boolean equations.
3. Compare the two results.
4. State De Morgan's second law.

### 4.10.3 Distributive Law - Activity # 14

The distributive law in Boolean algebra governs the order in which operations are executed.

#### Purpose:

To demonstrate the distributive law of Boolean algebra.

#### Theory:

The distributive law in mathematics is

$$x(y + z) = xy + xz$$

Multiplication can be distributed over addition. This law is used extensively for factoring in algebra. The equivalent law in Boolean algebra is

$$A \bullet (B + C) = (A \bullet B) + (A \bullet C)$$

where the dot represents an AND gate and + represents an OR gate.

#### Procedure:

Use the following diagrams to wire the circuits. Connect pin 14 to a +5V power source and pin 7 to ground on both chips.

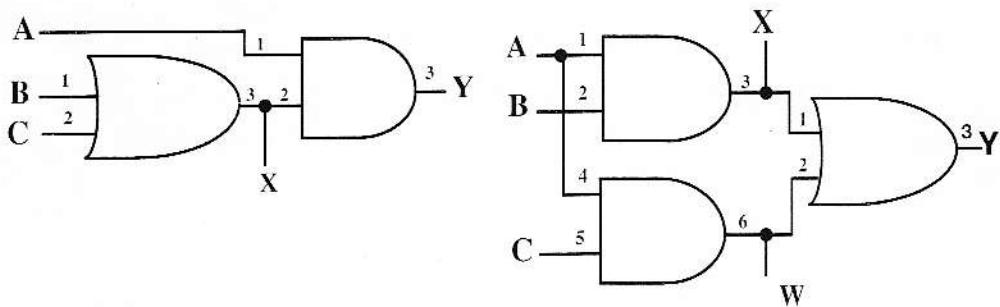


Figure 4.18

**Conclusion:**

State how an AND gate can be distributed over an OR gate.

**Questions:**

1. In mathematics addition CANNOT be distributed over multiplication. In multiplication

$$x + (yz)$$

does not equal

$$(x + y)(x + z)$$

The equivalent law in Boolean algebra would be

$$A + (B \bullet C) = (A + B) \bullet (A + C)$$

- (a) Draw two circuit diagrams, one to represent the Boolean equation

$$A + (B \bullet C)$$

and a second diagram to represent

$$(A + B) \bullet (A + C)$$

- (b) Wire the above two diagrams and construct Truth Tables for each.
- (c) Compare the final results for each Truth Table.
- (d) State the relationship between

$$A + (B \bullet C)$$

and

$$(A + B) \bullet (A + C)$$

2. How many distributive laws are there in mathematics?

## 3.6 Emulation

If every manufacturer using a computer chip in a product or process had to design that chip from scratch, the cost of the final product would be enormous. One of the reasons that the cost of electronic components is dropping is that manufacturers are able to incorporate already-existing chips and customize them to meet their needs.

What this customization process often involves is getting a chip designed to perform one function to carry out a different function. This process is called **emulation**, because the goal is to get one chip to behave in the same fashion as another chip.

The NAND gate is the most commonly-used gate because it can emulate the other five fundamental gates: AND, OR, NOT, NOR and EOR. For this reason, NAND is called the **universal gate**.

The following five examples show how a series of NAND gates can be used to emulate other gates. In each of these cases, the output of the Y of the traditional gate and the corresponding emulated gate will be the same.

### Emulated AND Gate

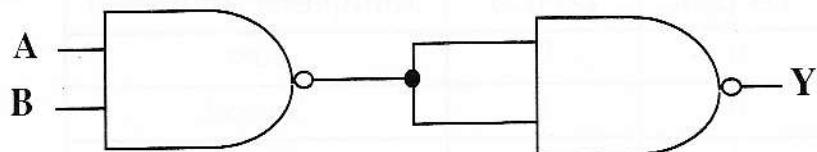


Figure 3.38

### Emulated OR Gate

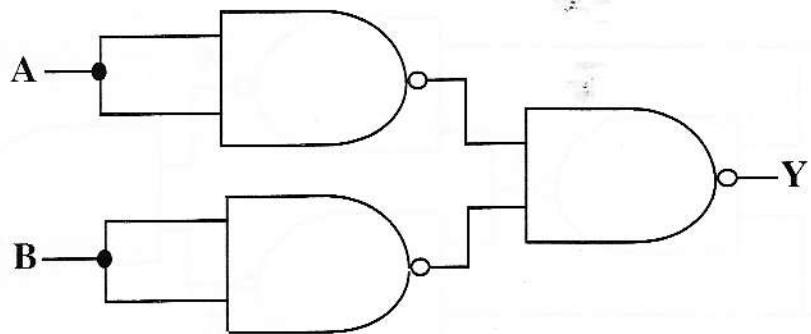


Figure 3.39

### Emulated NOT Gate

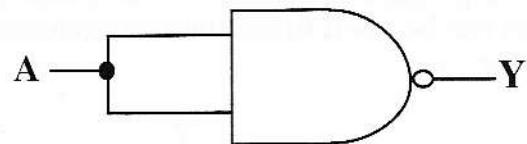


Figure 3.40

### Emulated NOR Gate

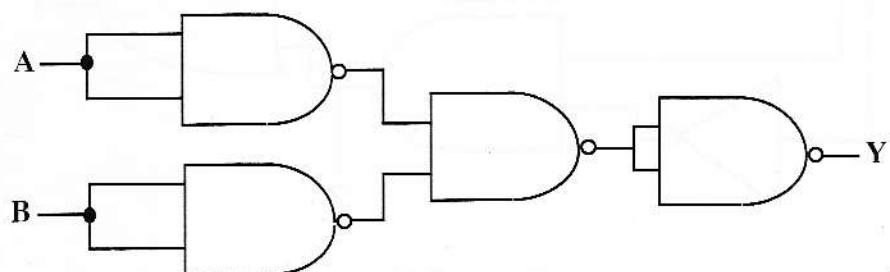
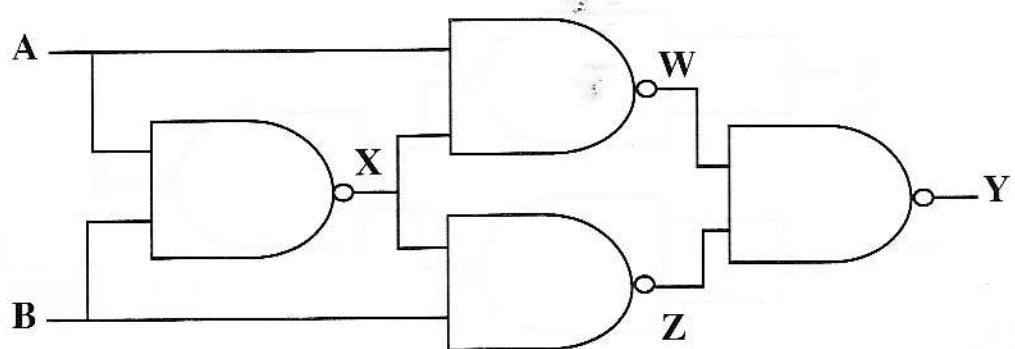


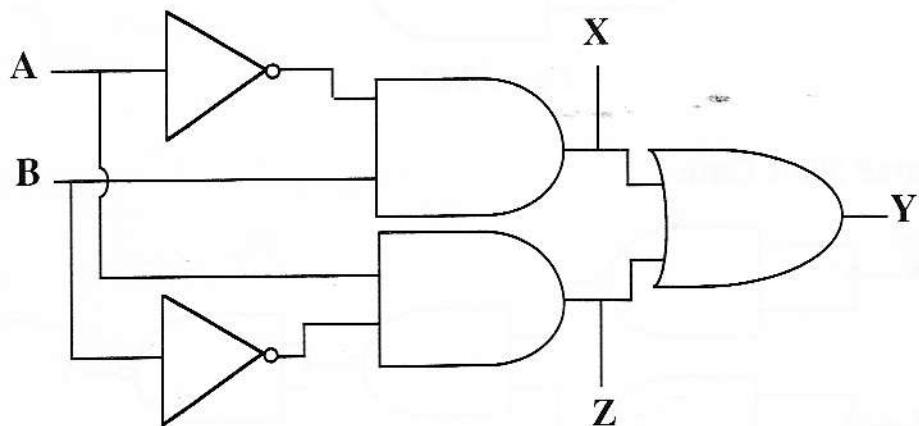
Figure 3.41

### Emulated EOR Gate



*Figure 3.42*

Gates other than NAND can also be used for emulators. For example, the following figure shows how a combination of AND, OR, and NOT gates can be used to emulate an exclusive OR gate.



*Figure 3.43*

Here is the Boolean algebra for the emulated exclusive OR gate shown above.

$$X = \overline{A} \bullet B$$

$$Z = A \bullet \overline{B}$$

$$Y = X + Z$$

$$Y = (\overline{A} \bullet B) + (A \bullet \overline{B})$$

Here is the Truth Table for the emulated exclusive OR gate.

A	B	X	Z	Y
0	0	0	0	0
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

The result at Y is the same as an EOR gate. Therefore this circuit is another way of emulating an EOR gate without using NAND gates and without using the single EOR gate itself.

Using emulation in combination with DeMorgan's Laws and the Laws of Boolean algebra simplifies circuit design and development, which in turn makes the circuits easier to debug, lowers development and operating costs and reduces energy use.

It is important to note, however, that using chips to emulate other chips has certain drawbacks. For example, this can decrease the efficiency of the circuit and can also increase the assembly time because emulated circuits require more gates.

The circuit diagram for the negated AND gate

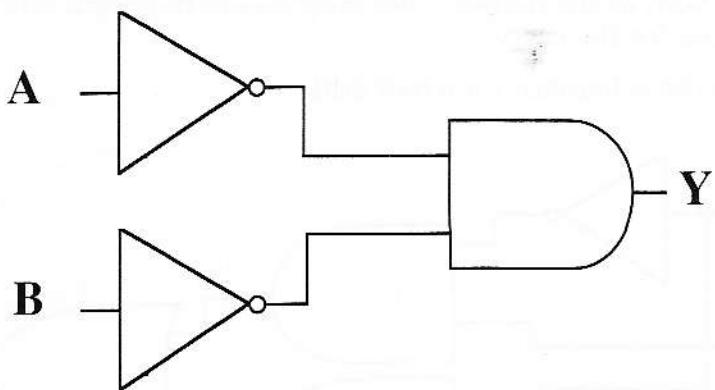


Figure 3.45

is equivalent to the above diagram. Other gates can have their inputs negated in a similar fashion.

## 3.8 Adders and Subtractors

We will now look at an example of combining the six fundamental gates to build more complex electronic structures.

All computers must contain electronic circuits that perform the arithmetic operation of addition. To add two simple numbers like 5 and 3, many operations are necessary. Some of these operations involve changing the decimal numbers 5 and 3 to binary numbers. The binary numbers must be added and then the sum, in binary, must be translated back to base 10 (see also Sections 1.4 and 3.5.3). **Half adders** and **full adders** combine multiple gates to facilitate this process.

### 3.8.1 Half Adder

A half adder is a circuit that will accept a two-bit input and give the binary sum as the output. Two outputs are required: one for the sum and one for the carry.

Here is the schematic for a half adder.

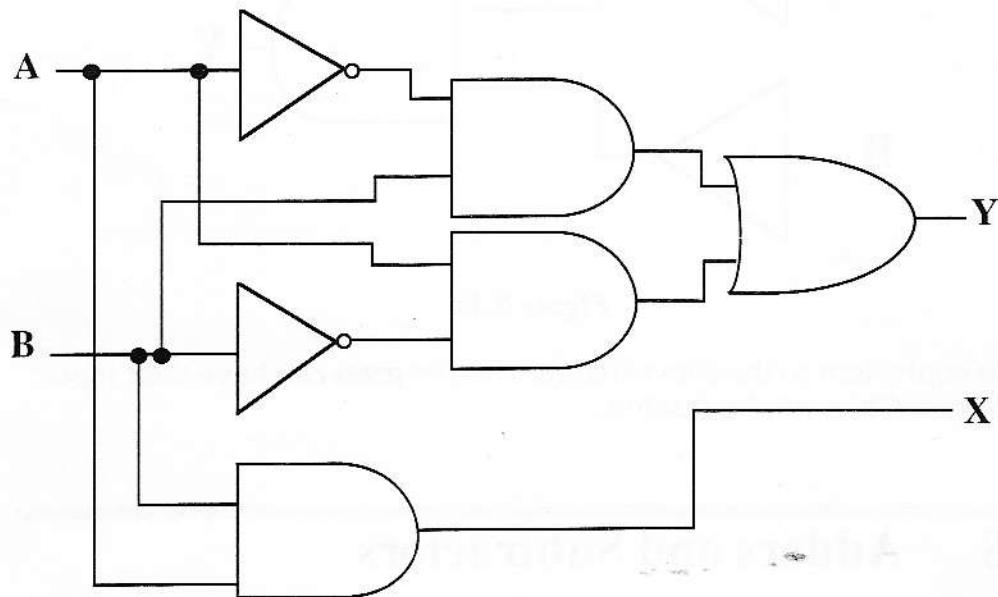


Figure 3.46

Here is an algebraic representation of the process by which a half adder adds two digits. The subscript number which follows the letter indicates that the addition is in binary (base 2). A is the first binary digit to be added and B is the second binary digit. Y indicates the sum and X is the carry.

$$\begin{array}{r}
 A_2 \\
 + B_2 \\
 \hline
 XY_2
 \end{array}$$

Here is a mathematical example of adding the two binary digits 1 and 1 to produce a binary answer of 10.

$$\begin{array}{r} 1_2 \\ + 1_2 \\ \hline 10_2 \end{array}$$

The above half adder requires six gates (two NOT gates, three AND gates, and one OR gate). A simpler circuit consisting of only two gates will produce the same results:

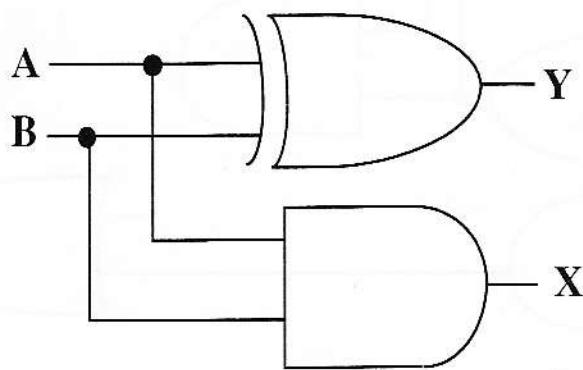


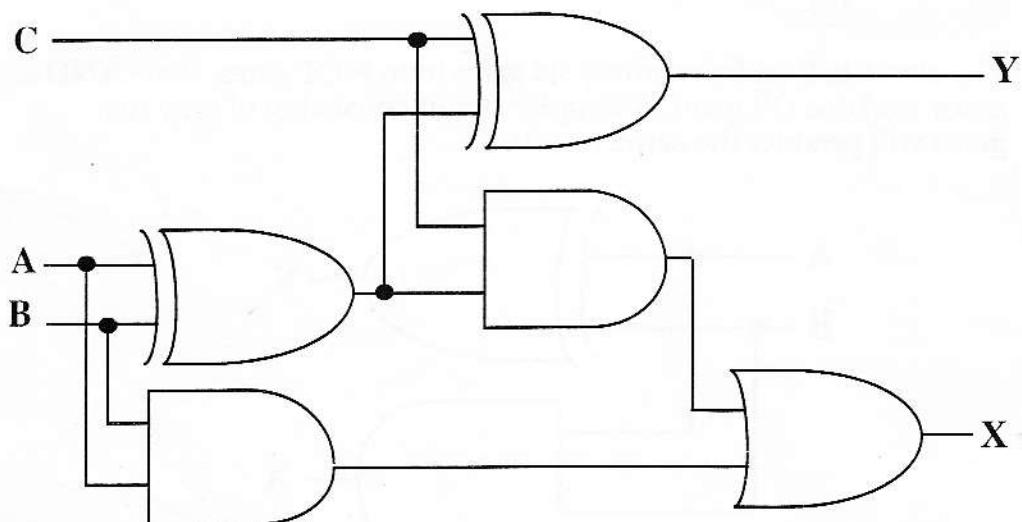
Figure 3.47

Again, A and B are the bits to be added, Y is the sum, and X is the carry.

A half adder can therefore add two bits and produce a sum and a carry out to the next column. If another two bits in the next column are included, the carry out from the first column must be added in. Adding these three bits together requires a full adder.

### 3.8.2 Full Adder

A full adder, like the half adder, has two inputs labeled A and B. The full adder, however, also has an input C which is the carry from adding two previous digits. These three inputs again produce a sum Y and a carry out to the next column of X.



*Figure 3.48*

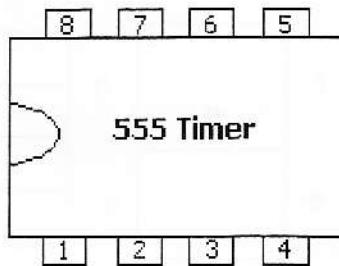
When several groups of two digits are to be added, a half adder can be used for the first pair of digits (since there cannot be a carry-in from a previous addition). A full adder is, however, needed for any subsequent pair of digits, since there will be a carry-in. The carry-in will be either 0 or 1.

### 3.8.3 Half Subtractor

A half subtractor is a circuit that will accept a two-bit input and give the difference between those two bits as the output. Two outputs are required: one for the difference and one for the borrow. (See Section 4.12.1 for more information on half subtractors.)

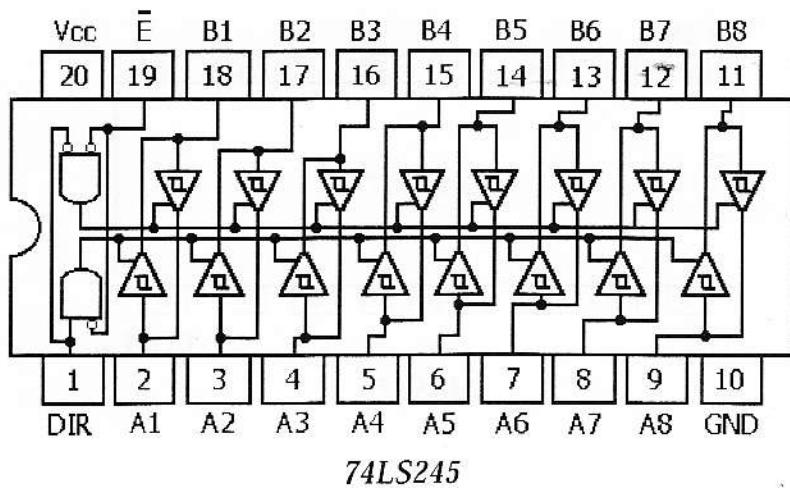
## 555 TIMER

Here is the chip schematic for the 555 Timer gate.



## Octal Bus Transceiver

Here is the chip schematic for the 74LS245 octal bus transceiver.



## 4.16 Timers - Activity # 26

Timers are chips that control the speed of the on/off signals in a computer.

### Purpose:

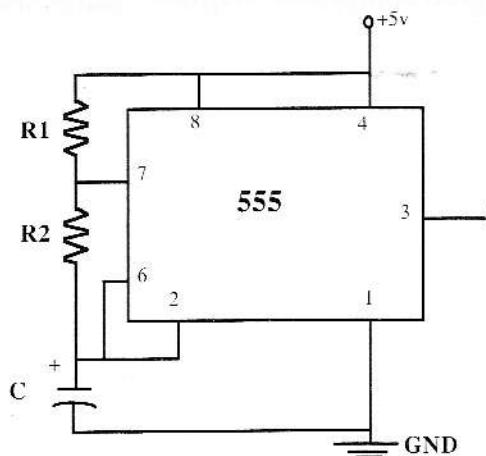
To build a clock circuit.

### Theory:

The 555 timer chip is widely used to generate a constant on/off signal. The duration of the on/off time can be changed by varying R2 in the diagram. A capacitor is a device that will store a charge and then release it in a burst (here lighting an LED) at a speed determined by the resistors R1 and R2.

### Procedure:

Use the following diagram to wire the circuit. Connect pin 4 and pin 8 to a +5V power source and pin 1 to ground.



*Figure 4.32*

Notes: R1 is resistor 1 (always 2 k $\Omega$ ).

R2 is resistor 2 (here assuming four different values).

C is the capacitor (always  $2.2\mu F$ ).

The times can only be approximated since the LED switches quickly.

### Results:

Complete the Truth Table.

C	R1	R2	Time on	Time off
$2.2\mu F$	$2 k\Omega$	$51 k\Omega$		
$2.2\mu F$	$2 k\Omega$	$100 k\Omega$		
$2.2\mu F$	$2 k\Omega$	$510 k\Omega$		
$2.2\mu F$	$2 k\Omega$	$1.5 M\Omega$		

### Conclusion:

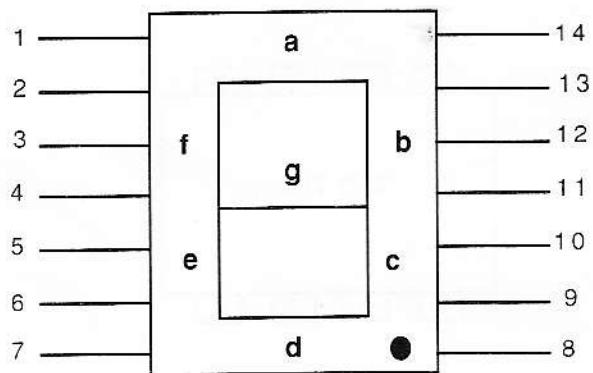
What happens to the speed of the flashing as R2 increases in value?

### Question:

1. What could the output at Y simulate?
2. What is the relationship between the time the LED is on and the time it is off?
3. How can the clock speed be increased?

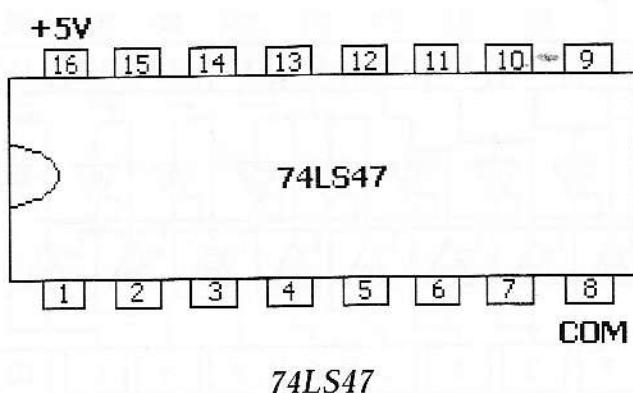
## Seven-Segment Display

Here is the chip schematic for the seven-segment display.



## Decoder

Here is the chip schematic for the 74LS47 (decoder) chip.



## 4.14 Seven-Segment Display - Activity # 24

The seven-segment display provides a way of expressing binary information in a decimal format.

### Purpose:

To determine the pins in a seven-segment display that will light each of the seven segments.

### Theory:

A seven-segment display is composed of seven small bars, each of which is an LED. By knowing which segment to light, any digit from 0 through 9 can be formed. Pin 3 and pin 14 are connected to a +5V power source (also called a positive) through a  $220\Omega$  resistor. This is a common anode configuration for a 'MAN72A' display since pin 14 is always connected to +5V.

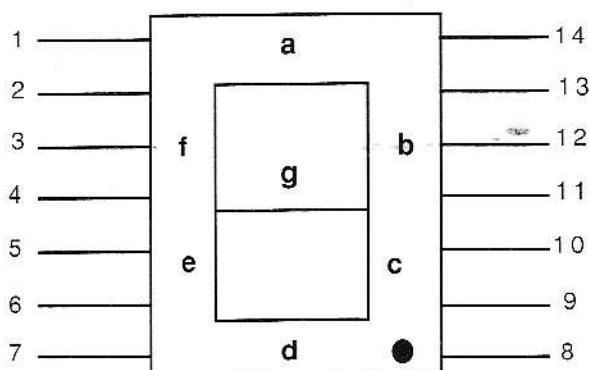


Figure 4.30

## 4.15 Binary to Base 10 Decoder - Activity # 25

Digital computers deal with all information in terms of two numbers: 0 and 1. Each of these binary (base 2) numbers represents the state of the flow of electricity through the circuitry, with 0 indicating off or no flow and 1 indicating on or flow. Because human beings process numerical information most easily in the decimal system (base 10), there must be a facility that provides the translation from binary to decimal.

### Purpose:

To translate or decode a binary (base 2) input to a base 10 output.

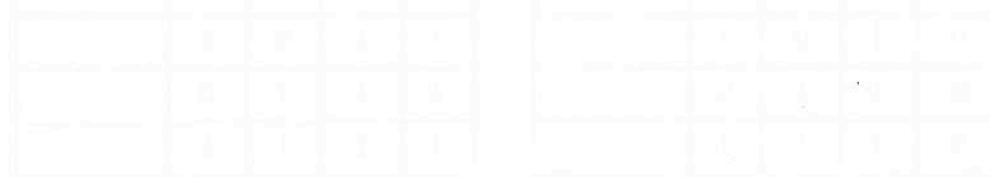
### Theory:

Digital computers deal with on (1) and off (0). The 74LS47 will translate a binary input into signals to a seven-segment display. The seven-segment display will produce any single digit from 0 through 9. (Refer to the Appendices for the exact pin placements on the seven-segment display and the decoder (74LS47) chip.)

### Procedure:

Use the following diagram to wire the circuit. Connect pin 16 to a +5V power source and pin 8 to ground. Note that in this schematic pin 8 is connected to COM. This term is often used instead of GND (ground) and means the same thing.

For ease of drawing, the pins are not arranged in order on the diagram.



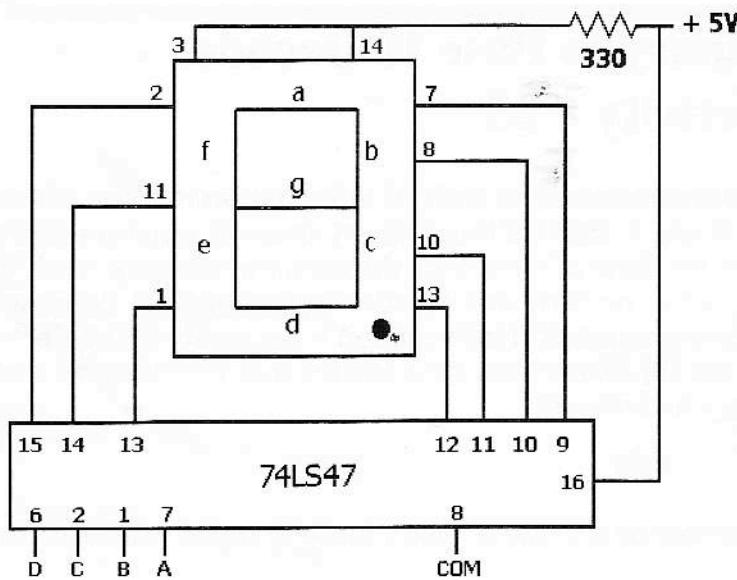


Figure 4.31

**Results:**

Complete these Truth Tables.

D	C	B	A	Display
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	

D	C	B	A	Display
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

**Procedure:**

Connect a  $220\Omega$  resistor from pin 14 and pin 3 to +5V.

Only two wires are connected to the display at one time. The first wire connects +5V to pin 3 and pin 14 through a  $220\Omega$  resistor (colour bars red, red, brown). The second wire connects pin 1 to ground. Note which segment lights. Move the second wire from pin 1 to pin 2 and again note which segment lights. Repeat for all pins except 3 and 14.

The segments are labeled a, b, c, d, e, f, g, and dp (decimal point).

**Results:**

Complete the following table.

PIN	1	2	3	4	5	6	7
Segment			+5V	NP	NP		
PIN	8	9	10	11	12	13	14
Segment					NP		+5V

Note 1: NP means no pin is present.

Note 2: Pin 3 and pin 14 are connected to +5V through a resistor. Connecting pin 3 and pin 14 directly to +5V will damage the seven-segment display.

**Conclusion:**

Name the pin number that controls each of these segments.

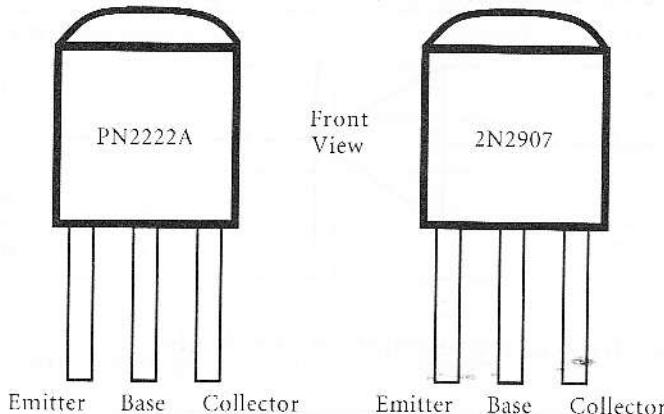
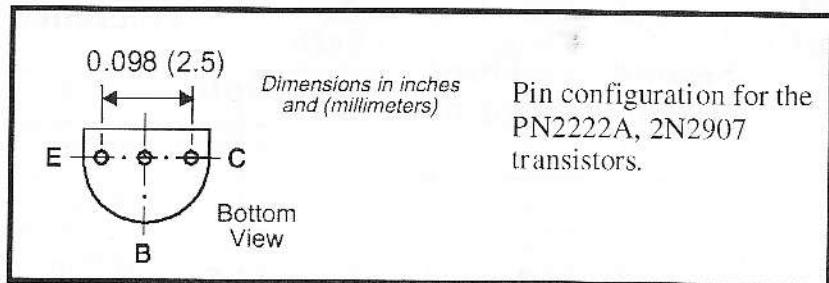
Segment	Pin Number
a	
b	
c	
d	
e	
f	
g	
dp	

**Question:**

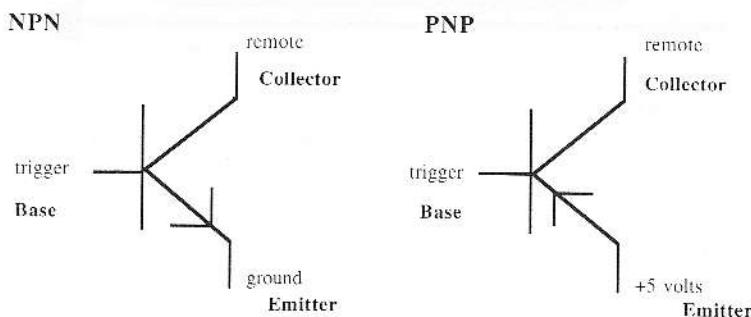
1. Which pins must be connected to ground to form the number 1?
2. Which pins must be connected to ground to form the number 8?
3. Why would this particular seven-segment not display letters well?
4. Why is this seven-segment called a numeric display?

# Transistors

Here is a physical representation of the PN2222A and 2N2907 transistors. The PN2222A is a NPN. The 2N2907 is a PNP.

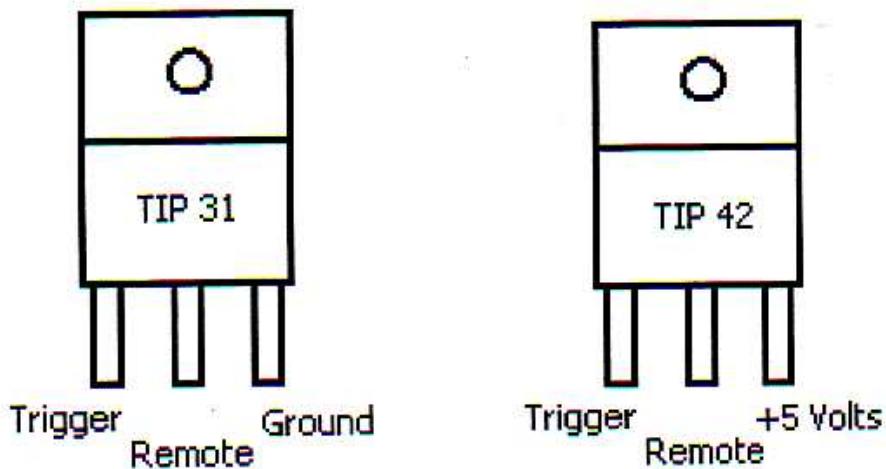


In a schematic transistors are represented by these symbols.

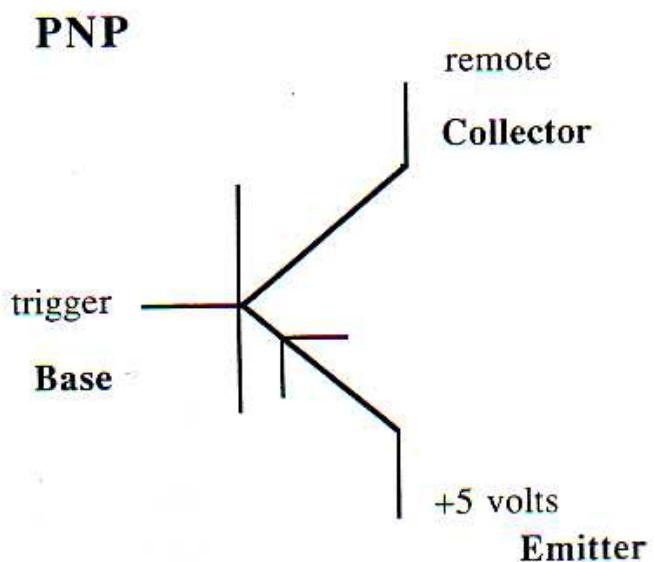
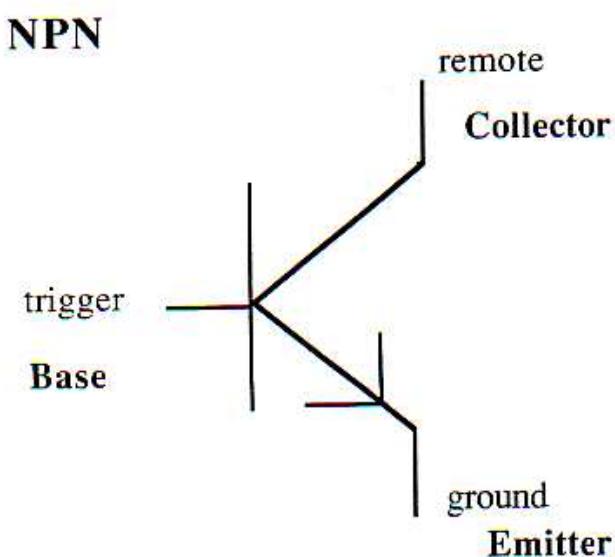


# Transistors

Here is a physical representation of the TIP 31 and TIP 42 transistors. The TIP 31 is an NPN. The TIP 42 is a PNP.



In a schematic transistors are represented by these symbols.

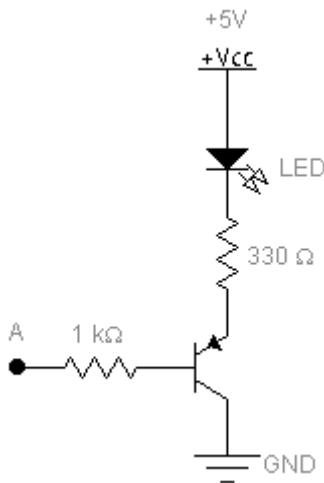


## Electronics Lab Exercise #1: Transistors

**Purpose:** To discover how transistors work, and see their connection to logic gates.

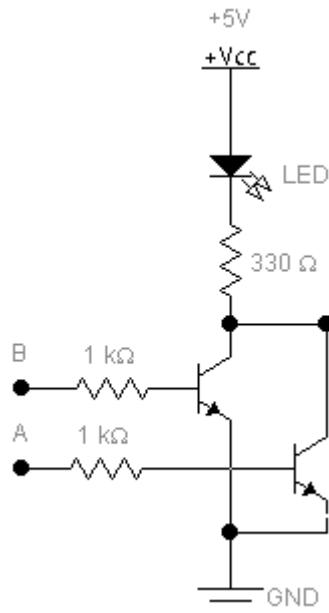
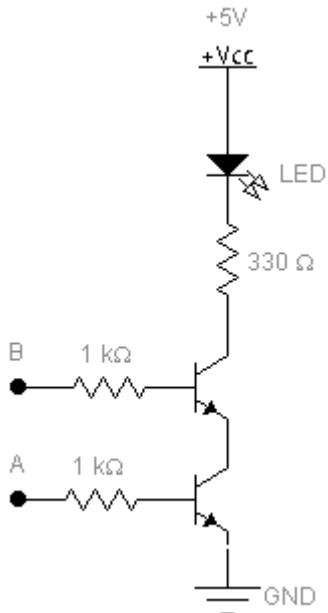
**Theory:** Transistors act like switches, and can be used to create logic gates.

**Procedure #1:** Construct and test the following circuit, using both types of transistor (3904, 3906). Make sure that you include the appropriate resistors to prevent damage to the components. "A" is one of the standard inputs.



1. Which transistor requires A to be on in order to turn on the LED?
2. Using the transistor from question 1, try replacing the LED with the bi-directional motor and the buzzer. Are you able to get these to work?

**Procedure #2:** Construct the following circuits, using the transistor mentioned in procedure #1.



1. Write truth tables for each of these circuits. What logic gates do these circuits represent?
2. Construct a circuit that represents any one of the other 2-input logic gates.

## 7.5.6 Stepper Motor – Activity # 6

### Purpose:

To control a DC stepper motor with the computer.

### Theory:

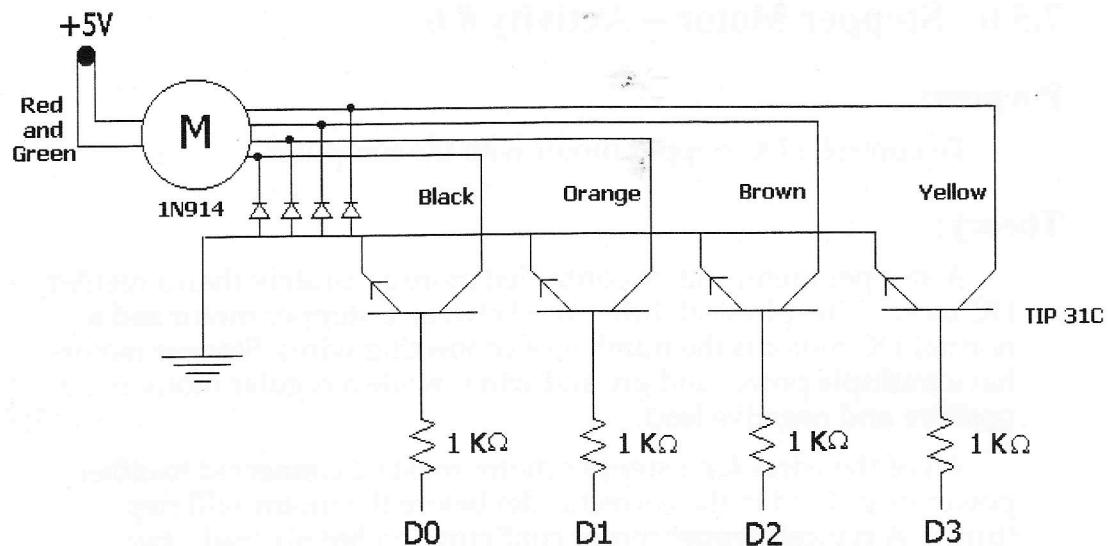
A stepper motor can be controlled more accurately than a regular DC motor. The physical difference between a stepper motor and a normal DC motor is the number of connecting wires. Stepper motors have multiple power and ground wires, while a regular motor has a positive and negative lead.

All of the wires for a stepper motor must be connected to either power or ground in the correct order before the motor will step (turn). A typical stepper motor configuration has six leads: two positive and four negative. The two positive leads can be connected to the power supply and the other four must be grounded individually through the TIP 31 transistor which is triggered by the computer. Diodes are used to protect the transistors in the circuit from feedback.

### Diagram:

Here is a diagram of a stepper motor that is powered through four transistors using pins 2 through 5 on the parallel port.

Note 1: Isolate the computer. (See Section 7.2 for more information on protecting the computer.)

*Figure 7.9***Code:**

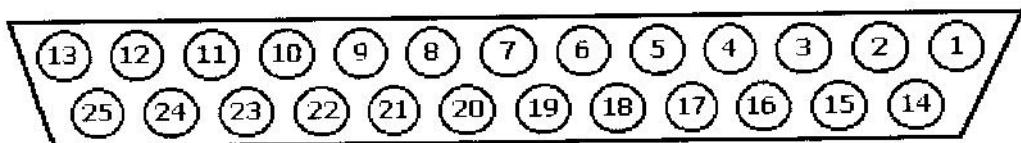
This program rotates the motor forward, stops the motor, then rotates it backwards.

```
% Repeats the sequence one hundred times.
for count : 1..100
    % Counts 0, 1, 2, and 3 which activates D0, D1, D2,
    % and D3 successively.
    for counter : 0 .. 3
        % Outputs to the parallel port.
        parallelput (2**counter)
        % Delays for one-tenth of a second.
        delay (100)
    end for
end for
% Delays for one second.
delay (1000)
```

## 7.1 Understanding the Parallel Port

Before you can begin constructing the interfaces in the Activities, you must first have a thorough understanding of how the parallel port works and the function of each of its pins.

The parallel port is a connector at the back of the computer that provides direct access to the computer bus. The bus is the electrical pathway which connects the processor, the memory, and the peripheral devices. The signal carried between them via the bus is a 5-volt electrical signal with low current capability. The parallel port is most commonly used to connect the printer to the computer. The diagram below shows the pin configuration of the parallel port when viewed from the back of the computer.



*Figure 7.1*

The parallel port connector has twenty five pins. These pins allow the computer to be connected to external devices through a parallel port cable. The port normally has eight pins capable of transmitting data out to external devices. These are the pins numbered 2 through 9 on the port itself. On circuit diagrams, these eight pins are labelled D0 through to D7. The D indicates that these are **data pins**. These data pins send information to the interface via the parallel cable.

There are five input pins that allow information to be sent back into the computer. These pins are numbered 10, 11, 12, 13 and 15. On circuit diagrams, these five pins are labelled I0 to I4. The I indicates these are **input pins**. Information can be sent from the computer to the peripheral and from the peripheral back into the computer. Software controls the output of information on the eight

## 7.5.8 Inputting Data to the Parallel port – Activity # 8

### Purpose:

To build a simple circuit that will return a value to the computer when the circuit is completed.

### Theory:

In the previous Activities, information has been output from the computer using the *parallelput* command. A similar command will allow information to be received through the parallel port using I0-I4. I0 should be isolated. The ground wire connects to one side of the switch. The other side of the switch attaches to A1 of the 74LS245 chip. B1 connects to I0. Pins 10, 19, and 20 of the 74LS245 chip must be grounded. Pin 1 is set high.

### Diagram:

Here is a diagram of a switch connected to the first input pin on the parallel port.

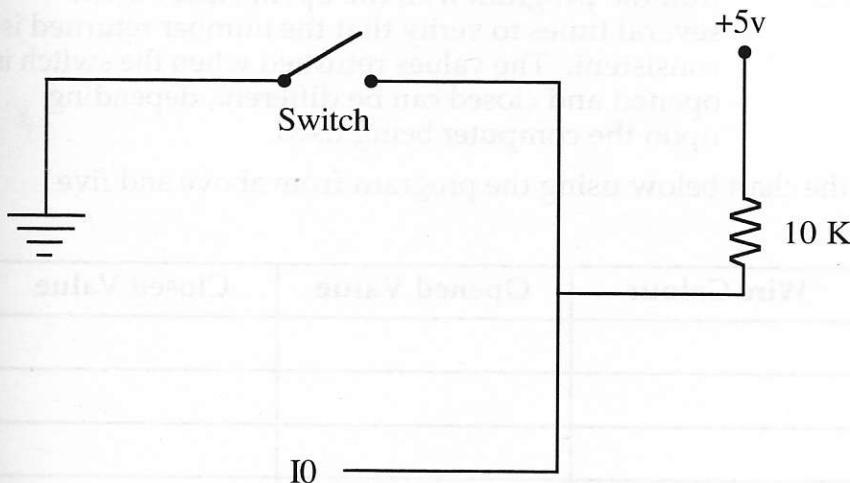
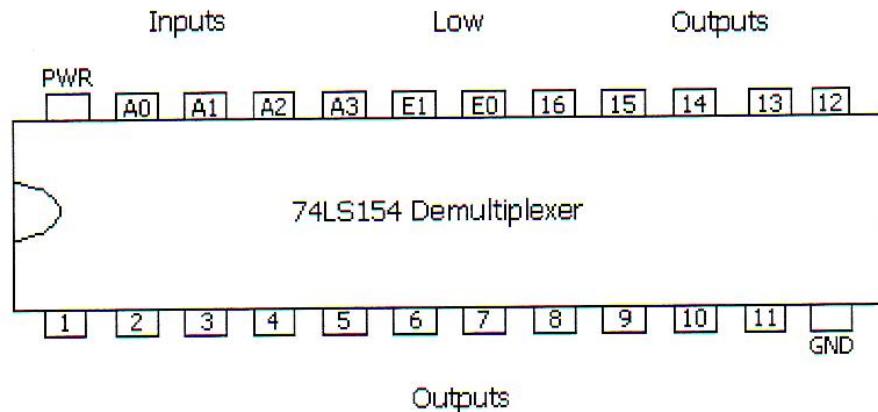


Figure 7.11

Here is the electronic schematic for the demultiplexer.



*Figure 3.35*

E0 and E1 must be held low to activate any output pins. E0 and E1 are also called the enabling pins since they both must be low to enable the output pins.

For example if A0 to A3 are all set low, then pin 1 will be set low and pins 2 through 16 will be high. If all of the pins from 1 to 16 are inverted (using NOT gates) then all outputs will be low except for the first pin. Inputting all lows on A0 to A3 sets pin 1 to high.

Similarly if A0 to A3 are all set high then pin 16 would be low and pins 1 through 15 would be high. Inverting pins 1 through 16 would again isolate a single pin as high, namely pin 16. Following this method, any one of the pins from 1 through 16 can be set high. A demultiplexer by itself will not activate two pins simultaneously.

Using a 4 to 16 demultiplexer, four pins (A0 – A3) can address sixteen separate pins when the enabling pins are set low.

In the following chart, column 1 represents all the possible inputs to pins A0 to A3. The demultiplexed output is on pins 1 through 16 in columns two to five and the inverted outputs are listed in the last four columns.

### 7.5.7 Two Bytes – Activity # 7

#### Purpose:

To control sixteen LEDs using four data-out lines.

#### Theory:

The 74LS154 chip is a demultiplexer and is used to decode four input signals into sixteen output signals. The four bits (nybble) can generate sixteen numbers. Each one of these combinations will have its own output pin on the 74LS154.

#### Diagram:

Here is a diagram of a demultiplexer controlling sixteen individual LEDs where the pins have been labelled for ease of drawing.

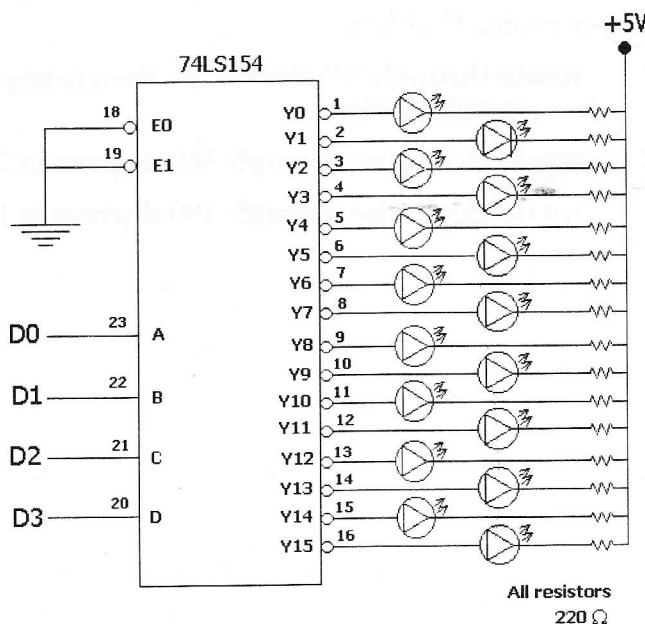


Figure 7.10

In order for the motor to function, one side must have power and the other side must be grounded. The wire from the right side of the motor is connected to the upper-right transistor (which is not letting electricity flow through it) and through the left side diode. From there, electricity flows through the lower-left transistor and from there to ground.

When D0 is high and D1 is low, the motor gets power from the left wire and is grounded through the right wire. It is not too difficult to show that if D0 is low and D1 is high, the motor gets power through the right wire and is grounded through the left wire.

### Diagram:

Here is a diagram of a bi-directional motor controlled by two transistors using pin 2 and pin 3 on the parallel port. (See Section 6.1.4 for more information on transistors.)

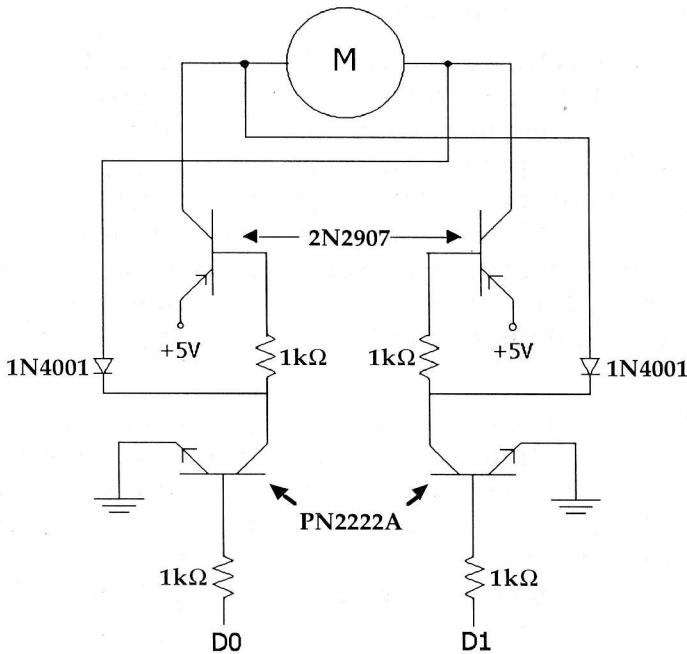


Figure 10.9 Schematic for Activity 10.5.5

- Note 1: Isolate the computer. (See Section 7.2 for more information on protecting the computer.)
- Note 2: See the Appendices for more information on the correct pin configurations.

### Truth Table:

Column 1 represents the four inputs from D3 to D0 (isolated). These are the inputs A, B, C, and D on the 74LS154. The 1 to 16 are represented in columns 2, 3, 4, and 5. All sixteen outputs are high (except one). Each of the sixteen outputs can be inverted (using 3 74LS04 chips). This would result in only one output high.

Output from PC	De-multiplexed Output				Inverted De-multiplexed Output			
0000	1111	1111	1111	1110	0000	0000	0000	0001
0001	1111	1111	1111	1101	0000	0000	0000	0010
0010	1111	1111	1111	1011	0000	0000	0000	0100
0011	1111	1111	1111	0111	0000	0000	0000	1000
0100	1111	1111	1110	1111	0000	0000	0001	0000
0101	1111	1111	1101	1111	0000	0000	0010	0000
0110	1111	1111	1011	1111	0000	0000	0100	0000
0111	1111	1111	0111	1111	0000	0000	1000	0000
1000	1111	1110	1111	1111	0000	0001	0000	0000
1001	1111	1101	1111	1111	0000	0010	0000	0000
1010	1111	1011	1111	1111	0000	0100	0000	0000
1011	1111	0111	1111	1111	0000	1000	0000	0000
1100	1110	1111	1111	1111	0001	0000	0000	0000
1101	1101	1111	1111	1111	0010	0000	0000	0000
1110	1011	1111	1111	1111	0100	0000	0000	0000
1111	0111	1111	1111	1111	1000	0000	0000	0000

# Appendix C: Boolean Algebra Rules

1.  $A + A = A$
2.  $A + \bar{A} = 1$
3.  $0 + A = A$
4.  $1 + A = 1$
5.  $\bar{\bar{A}} = A$
6.  $\bar{A} \bullet A = 0$
7.  $A \bullet A = A$
8.  $0 \bullet A = 0$
9.  $1 \bullet A = A$
10.  $A + B = B + A$
11.  $A \bullet B = B \bullet A$
12.  $(A + B) + C = A + (B + C)$
13.  $(A \bullet B) \bullet C = A \bullet (B \bullet C)$
14.  $A \bullet (B + C) = A \bullet B + A \bullet C$
15.  $A + B \bullet C = (A + B) \bullet (A + C)$
16.  $\overline{A + B} = \bar{A} \bullet \bar{B}$
17.  $\overline{A \bullet B} = \bar{A} + \bar{B}$
18.  $A + A \bullet B = A$
19.  $A \bullet B + \bar{A} = B + \bar{A}$

## 10.5.5 Bi-Directional DC Motor – Activity # 5

### Purpose:

To create a motor circuit that is able to turn in either direction using the computer to trigger the directional change.

### Theory:

The direction that a DC motor turns depends upon the direction of current flow through the motor. The way that the current flows through a circuit is determined by the placement of the power source (positive) and the ground.

Since we are working with a +5V power supply using conventional current, the current flows from the +5V to the GND. Reversing the direction that the current flows through the motor changes the direction the motor turns.

The flow of electricity in this circuit is somewhat complicated. When D0 is set high, the Trigger pin on the lower-left transistor is high, allowing current to flow through the transistor. This essentially grounds the Trigger pin on the upper-left transistor (creates an electrical path between the Trigger pin and the ground). The upper-left transistor is a PNP transistor which lets electricity flow through when the Trigger pin voltage is substantially less than the voltage in the Emitter pin. (The Trigger pins on all the transistors are protected with a current limiting resistor to stop the Trigger pin from getting too much current.)

When D0 is set high, electricity flows from the left +5V power supply through the transistor and into the left side of the motor. It also flows through the right diode and into the Trigger pin of the upper-right transistor. This causes the upper-right transistor to not allow electricity to flow. Of course, this depends upon D1 being set to 0, which prevents any electricity from flowing through the lower-right transistor.

