AND-OR GATES

INTRODUCTION

NUMBERING SYSTEMS

Man probably first began counting with sticks, stones, and on his fingers. It is reasonable to assume that the 10 digit or decimal system used throughout the civilized world had its beginning with man's 10 fingers. In fact, the word *digit* means *finger*.

The number of different symbols or digits used in a system is called the base or *radix* of the system. The radix of teh decimal system is 10, since 10 digits are used.

While a number system is possible with any radix larger than one, only a few number systems, other than the decimal system, have any practical applications today. Systems that have been used are the binary system (base 2), ternary system (base 3), octonary (base 8), and the duodecimal system (base 120.

Although a computer could be designed to operate with any number system, only the decimal, octonary, and binary systems are the most used. Each system has its own advantages and results in a particular type of computer. Often two systems overlap so that a computer uses both numbering systems and converts between the two.

The advantage of the binary system is that only two symbols are used. These symbols are zero and one; any number, no matter how large, can be written with them. Since only two symbols are used, a computer using the binary system needs only two different signal voltage values to represent any number. This simplifies design and improves accuracy. For this reason, the binary system is widely used in computer and logic equipment.

DECIMAL SYSTEM

Few people take time to examine this number system, which is used every day; yet, it represents one of the great achievements of the human mind. Any number of any value can be written with only 10 different symbols (1, 2, 3, 4, 5, 6, 7, 8, 9 0). These symbols are called *digits*.

The first important feature of the number system used is the idea of *place value*. Not all systems use this place value. For example, the Roman system does not use place value. In figure 5-1, the 3 written in the second column stands for 30. This placement of the digits 980 and 6060 can be easily understood. Thus, in the modern number system the value of a digit depends upon its place in the number.

When the idea of *place value* was first developed, a space was left to indicate that no number appeared in that position. For example, 106 was written 1   6 and 1006 was written 1   6. It is easy to see that this system led to confusion. Does 1   6 represent 1006 or 10006? It is difficult to decide with certainty.

The defect in the system eventually led to the development of a *place holder*, or zero. Although zero represents *NOTHING*, it is one of the truly great inventions of man and makes possible much of the simplicity of our number system and mathematics of today. Zero is the second important feature of the decimal system.

Moving a digit to the left from one column to the next in figure 5-1 has the effect of multiplying the number by 10. The digit *3* in the first column represents the value three. If the digit 3 is moved one more column to the left, it signifies the value 30. If moved one more column, the digit *3* represents the value 300, and so on. Therefore, it is said that this is a decimal system (decimal means *ten*).

A common method of writing numbers is called *powers of 10*. For instance, 10×10×10 may be written 10^3. The number *3* tells how many times 10 is used as a factor. In the statement $10^3 = 1000$, 10 is called the *base*, 1000 is the *power*, and 3 is the *exponent*. The number 10^3 is read "10 to the third power." The radix of a system is used as the base. For example, in the binary system there are "powers of two," since two is the radix.

10^5	10^4	10^3	10^2	10^1	10^0		
					3	=	3
				3		=	30
			3			=	300
		3				=	3000
	3					=	30,000
3						=	300,000
			9	8		=	980
		6		6		=	6060

RCS82-21

Figure 5-1. Values Due to Position

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
128	64	32	16	8	4	2	1	
							1	= 1
						1	0	= 2
					1	0	0	= 4
				1	0	0	0	= 8

RCS82-19

Figure 5-2. Binary System Place Values

BINARY SYSTEM

The binary system uses the radix two. This system has only two symbols, *0* and *1*. Using these two symbols, any number can be written. Zero stands for nothing and is the *place holder* for the system. One stands for unity or a single unit.

Since only two symbols are used, the number *2* must be written by combining these two symbols in some way. The binary system has a place value just as the decimal system. Figure 5-2 is a chart similar to figure 5-1.

Instead of using powers of 10 across the top, the powers of 2 are used since the radix of the binary system is 2. The digit *1* in the first column represents the value one. Moving to the next column to the left, the *1* represents 2^1 or 2. One more column to the left represents 2^2 or 4, and in the third column the digit *1* represents 2^3 or the value of 8. Thus, *1* represents 1, *10* represents 2, *100* represents 4, and *1000* represents 8.

Binary Division

Comparing the decimal system and the binary system will clarify the idea.

2/231		
2/115	R = 1	
		LSD
2/ 57	R = 1	
2/ 58	R = 1	
2/ 14	R = 0	
2/ 7	R = 0	
2/ 3	R = 1	
2/ 1	R = 1	
2 0	R = 1	MSD

The above example shows each step in the division method for converting 231 to its binary equivalent. Thus, 231=11100111.

RULES FOR THE DIVISION METHOD. To convert from decimal to binary notation, the number to be converted is divided by 2. The quotient obtained is then divided by 2 and this is continued until a final quotient of zero is obtained. Each division represents a digit of the binary number. If upon dividing by 2, there is no remainder, the digit is zero. Otherwise, the digit is one. The binary digits are written from right to left, with the first remainder on the right.

Binary Addition

The sum of two or more binary numbers can be determined by first converting them to decimal numbers and then by adding them. Direct addition, such as is done in a computer is as follows:

NOTE: Upon reaching a maximum count in any column (in binary counting, the maximum count is one), the column starts again at zero, and a one is carried over to the next most significant column.

RULES FOR ADDITION:

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 with 1 carried to the next column

The follo... is an example of binary addition:

(E)	(D)	(C)	(B)	(A)
	1	0	1	1
	1	1	1	0
1	1	0	0	1

Step (A): *1 + 0 = 1*

Step (B): *1+1=0* and carry the *1*

Step (C): The *1* carried from the (B) column is added here.

Thus, *0+1+1=0* and carry the *1*.

Step D: *1+1+1+= 1* with a carry of *1* to the next column.

Here the *1* carried from column (C) is added to column (D).

Breaking down the addition, *1+1+=0* with a carry of *1* to the next column.

0+1 (carried over from C)=*1*

Step (E): Since there is nothing in column (E), zero is understood. Therefore, *1+0=1,* where the *1* has been carried over from column (D).

Therefore, the sum is *11001.* Converting the binary numbers to decimal numbers and adding:

```
 1011 = 11
 1110 = 14
----------
11001 = 25
```

This conversion step is a good way to check the answer.

Any number of binary numbers can be added together. It is most important to remember to carry the 1 to the next column when adding.

The following list of rules will aid in remembering the steps in binary addition:

1. For an even number of 1's, including carries from preceding columns, the sum will always be zero.
2. For an odd number of 1's, including the carries from preceding columns, the sum will always be one.
3. Every two 1's in each column will carry a 1 to the next column.

Binary Substraction

There are many methods of subtracting binary numbers; among them are these two:

1. End-around-carry.
2. Direct subtraction.

The end-around-carry is the most frequently used method in computers because of the ease with which it can be accomplished.

The end-around-carry method of subtracting two binary numbers is accomplished by using the one's complement. The one's complement in the binary system corresponds to the nine's complement in the decimal system.

END-AROUND-CARRY. One of the easiest ways to subtract in binary notation is to add a one's complement to use the end-around-carry. A one's complement can be written down at sight by just putting a 1 for 0 and 0 for 1. To subtract using the end-around-carry, follow these rules:

1. Complement the subtrahend.
2. Add.
3. Bring the *1* from the end and carry it around.
4. Add.

Example: Subtract 0101 from 1110.

1. Complement the subtrahend.

```
1100
1010
----
```

2. Add.

```
 1110
 1010
-----
11000
```

3. End-around-carry.

```
 1110
 1010
-----
11000
└──► 1
```

4. Add.

```
 1110
+1010
-----
 1000
+   1
-----
+1001
```

5. Check the result by converting the original numbers to the decimal system and subtracting in the conventional method.

```
 1110 = 14
-0101 = -5
----------
 1001 =  9
```

DIRECT SUBTRACTION: Follow these four rules for direct subtraction.

1. 0 - 0 = 0
2. 1 - 1 = 0
3. 1 - 0 = 1
4. 0 - 1 = 1 with a borrow of 1 from the next column.

NOTE: In direct subtraction in *any* number system, remember that the radix of that system is always borrowed when it is necessary to borrow.

Example:

```
 10110
 -1010
------
  1100
```

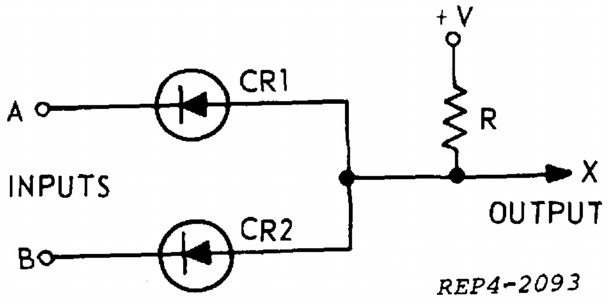


Figure 5-3. Diode AND Gate

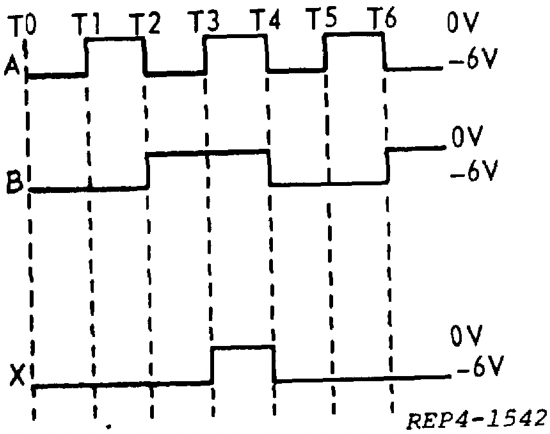


Figure 5-4. AND Gate Waveforms

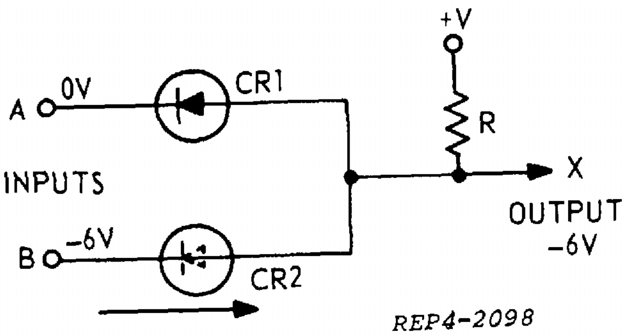


Figure 5-5. AND Gate (Equivalent Circuit of Two Low Inputs)

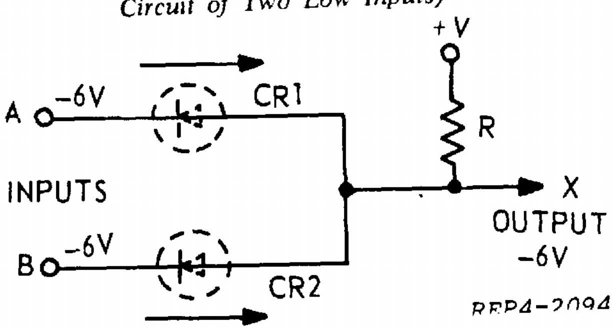


Figure 5-6. AND Gate (Equivalent Circuit, One High Input)

INFORMATION

Logical gating circuits are employed in digital computers to correlate continuous trains of pulse information. To understand these circuits, it is necessary to become familiar with their theory of operation as well as their varied applications.

The following text introduces the theory of logical gating circuits, including a descriptive definition, general types of logical gating circuits, and a complete analysis of circuits in general use.

LOGICAL FUNCTIONS

A computer is no more than a combination of simple devices which perform a few basic operations. The complexity of a computer arises only from the large number of these simple devices and the way they are interconnected. The interaction of signal as they process through a computer is called *logic;* the circuits involved are called *logic circuits.*

Outputs of logic circuits have two voltage values called *logic levels.* One level represents binary *1* and the other represents binary *0.* If the more positive of the two voltage values represents 1, the circuit uses *positive logic.* If the more negative value represents 1, the circuit uses *negative logic.* The first part of this chapter uses positive logic and a separate section discusses negative logic.

It is possible to design a computer using just two kinds of components: *gates,* which transmit signals only when input signals are present in specified combinations; and *storage elements,* which store or remember a signal so that it may be used at a later time. This section is limited to the discussion of *gates.*

DEFINITIONS

Gate

A *gate* is a device having two or more inputs and one output. Some of the inputs may be called *signal* inputs and others may be designated as *control* or *selector* inputs; although the inputs are often indistinguishable from one another.

Matrix

A *matrix* is any combination of AND or OR gates or both for computer detection operations. The matrix has a number of inputs and outputs and is so connected that signals representing information expressed in a certain code, when applied to the inputs, cause output signals to appear which are representations of the input information in a different code.

Diode AND Circuitry

Circuits which perform the AND function are called *AND gates.* AND gates may use diodes as shown in figure 5-3. The circuit has one output, at which a pulse appears if, and only if, pulses are applied simultaneously to *BOTH* inputs. If the inputs are not of the same time duration, the output will appear only during the time interval that the input pulses overlap.

When both diodes have a high input, the output is *high.* When either diodes has a low input, or if both diodes have low inputs, the output is *low.* A low output is considered *NO* output and represents binary zero.

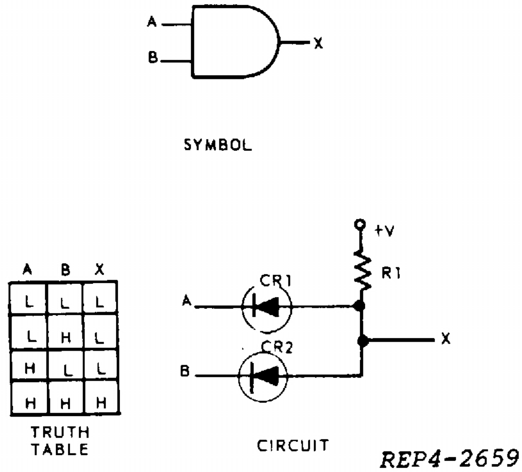
In the explanation of AND circuits, the logic levels are the numeric values of the high or low outputs. If the outputs are zero and -6, zero volts represent a high and -6 volts represents a low. The most positive value is high and the most negative value is low. The high represents binary 1; and the low represents binary zero.

Time *T0* to *T1* in figure 5-4 shows a -6 volts being applied to both diode inputs. Both diodes are conducting heavily, and the outut is -6 volts (low). Figure 5-5 shows the equivalent circuit with the diodes shorted.

Times T1 and T1, figure 5-4, show one input at zero volts (high) and the other at -6 volts (low). CR2 is forward biased and conducts heavily, clamping the output and the anode of CR1 to -6 volts. CR1 is reverse biased and does not conduct. Figure 5-6 shows the equivalent circuit with CR2 shorted. The -6 volt (low) output is identified as *no output.*

A *truth table* shows all the possible input conditions, and the output of each case. Figure 5-7 is a truth table which relates all the important features of AND gates. In this truth table the low (L) represents -6 volts and the high (H) represents zero volts. Compare each line of the truth table with specific circuitry as follows: In figure 5-5, when both diodes have a -6V input (L), the output is -6V (L). Figure 5-6 shows either diode input -6 volts (L) with the other diode input zero volts (H); the output is -6 volts (L). Figure 5-8 shows zero volts (H) on both diodes; this is the AND condition which provides an output (H).

The important feature to remember about an AND gate (figure 5-7) is that all inputs must be high.



A	B	X
L	L	L
L	H	L
H	L	L
H	H	H

TRUTH TABLE

Figure 5-7. AND Gate Features

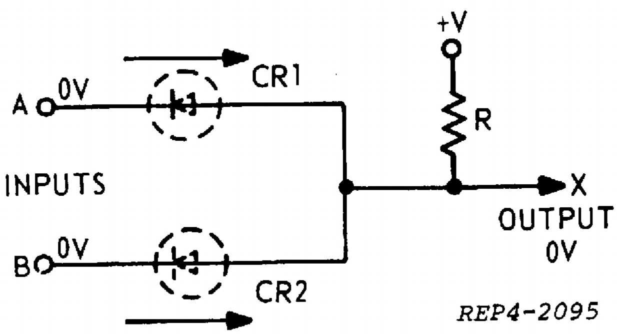


Figure 5-8. AND Gate (Two High Inputs)

Diode OR Circuitry

The diode OR circuit consists of two or more crystal diodes connected in the manner shown in figure 5-9. The load resistor is connected to a negative potential, the presence of information is represented by a high, while the absence of information is represented by a low.

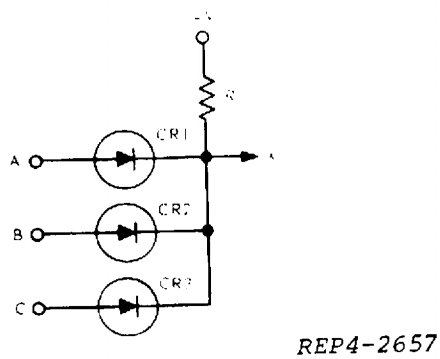


Figure 5-9. OR Circuit

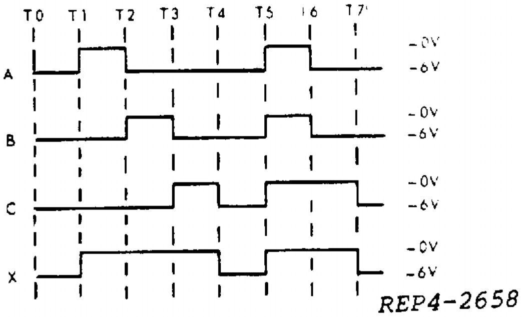


Figure 5-10. OR Circuit Waveforms

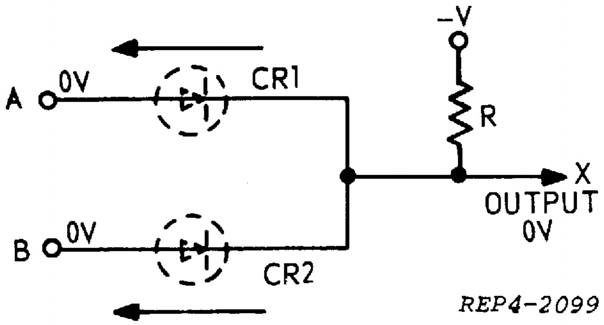


Figure 5-11. OR Circuit (One High Input)

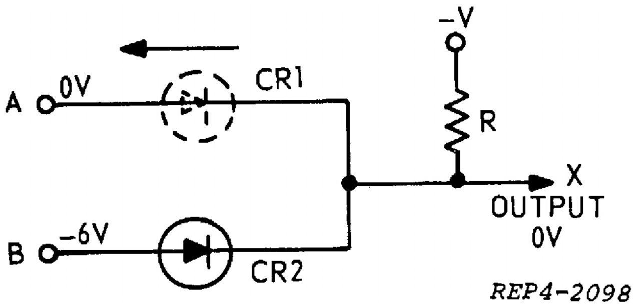


Figure 5-12. OR Circuit (Two High Inputs)

Figure 5-10 shows waveforms of the OR circuit. At time T0 to T1, all diodes are cut off, and the output si low (-6V). At time T1 to T2, CR1 conducts; and the output goes high (zero volts). At time T2 to T3, CR1 cuts off, but CR2 conducts; so the output remains high. From T3 to T4, CR3 conducts; and the output is high. All inputs must be low to have a low output; one or more diodes conducting makes a high output.

Figure 5-11 shows a two-diode OR circuit. With one diode anode at zero volts (high) and the other diode anode at -6 volts (low), the output is high (zero volts).

Assume no voltage drop across conducting CR1, the output is clamped to zero volts. This clamping action causes a zero potential to be felt on the cathode of CR2. Thus, the cathode of CR2 is positive with respect to its anode, causing CR2 to be reverse biased; and the diode will not conduct. This operation occurs when one input is high and the other is low; the output is high.

If both inputs are at zero (high), then both diodes will conduct; and the output is high, as can be seen in figure 5-12.

Figure 5-13 is the truth table which relates all of the important features of teh OR gate. In this truth table, the low (L) represents -6 volts while the high (H) represents zero volts. Verify the truth table which indicates the following:

All inputs low gives a low output.

Any one input high gives a high output.

All inputs high gives a high output.

The important feature to remember from figure 5-13 is that any or all high (H) inputs will produce a high (H) output.

A	B	X
L	L	L
L	H	H
H	L	H
H	H	H

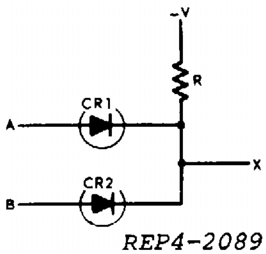
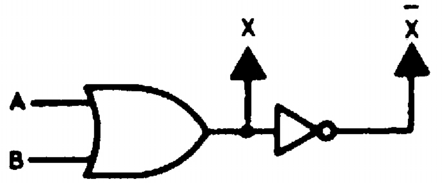


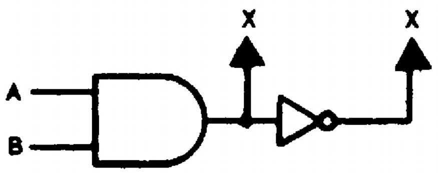
Figure 5-13. OR Gate Features

REP4-2661

Figure 5-14. Inverter Symbols



A	B	X	$\bar{X}$
L	L	L	H
L	H	H	L
H	L	H	L
H	H	H	L



A	B	X	$\bar{X}$
L	L	L	H
L	H	L	H
H	L	L	H
H	H	H	L

REP4-2660

Figure 5-15. NOT Function

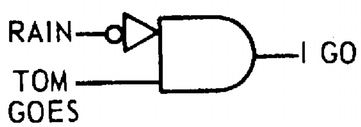
NOT Logic

Another logic operation of importance is the NOT function, which denotes an alternate or converse value. It can also be called INVERSION. When inverted, every high becomes a low. Similarly, an inverted low becomes a high. A line drawn over an input indicates a NOT function. NOT A is $\bar{A}$. If A equals 1, the $\bar{A}$ equals zero. If A equals zero, the $\bar{A}$ equals 1. An inverting amplifier can be used to obtain a NOT function. The logic symbols for such an amplifier are shown in figure 5-14. The symbol for an amplifier is the triangle, the small circle represents inversion; without the circle, the amplifier has no inversion.

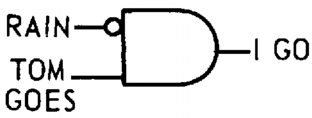
The small circle placed before or after the amplifier symbol is the *state indicator.* The circle at the input to figure 5-14A indicates that a LOW is required at the input to active the amplifier and produce a HIGH output. Without a LOW input this circle is not activated, and the output is LOW. The absence the input of figure 5-14B indicates a HIGH is required to activate the circuit. Without a HIGH input, the output remains HIGH. A HIGH input produces a LOW output. The polarity signals (in addition to L and H) indicate signal inversion occurs within the amplifier.

An amplifier can be used to obtain the inversion necessary for a NOT function. For example, the common emitter amplifier inverts its input signal: a positive going input gives a negative going output. Common base and common collector amplifiers develop outputs with the same waveforms as their inputs; they are noninverting amplifiers and cannot be used for a NOT function.

The output of an AND or OR circuit can become a NOT function by adding an inverting amplifier, as shown in figure 5-15. Recall that the AND function requires all inputs to be HIGH to get a HIGH output. Two HIGH inputs to a two-input AND function produce a HIGH at point X, which activates the amplifier, and the output at point $\bar{X}$ is LOW; at all other times the output remains HIGH. In the OR function, a HIGH at either input (or both) produces a HIGH at point X, which activates the amplifier and produces a LOW at point $\bar{X}$. Unless the amplifier is activated, the signal at $\bar{X}$ is HIGH.



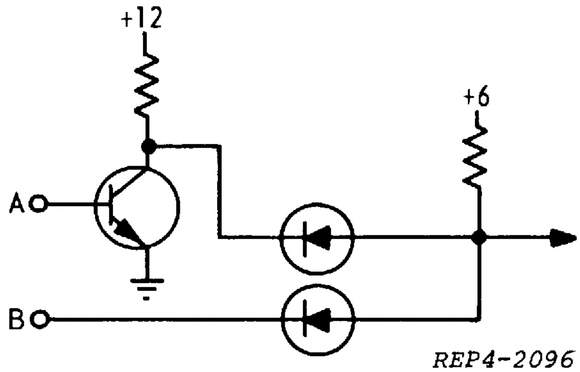
A. AMPLIFIER SYMBOL



B. STATE INDICATOR

	RAIN	TOM GOES	I GO
1.	H	H	L
2.	H	L	L
3.	L	H	H
4.	L	L	L

C. TRUTH TABLE



D. SCHEMATIC

Figure 5-16. Inhibit Function

INHIBIT GATE. The NOT function is often used in conjunction with the iinput to an OR or AND circuit. For example, someone might say, "I'll go if *Tom* does and it does *not* rain." Examination shows that this involves an AND function and a NOT function. This situation can be diagrammed as shown in figure 5-16A. A more complete method of diagramming is to omit the amplifier symbol and show only the state indicator in conjunction with the AND symbol as in figure 5-16B.

If rain is present (H), it prevents the AND circuit from producing an output. This prevention of the AND operation is called *inhibiting.* When a state indicator is used at the input of an AND circuit, the function is termed as *inhibit function.* The circuit which provides the inhibit function is called an *inhibitor.*

The truth table for the inhibit gate is shown in figure 5-15C. The truth table simply means:

1. It is raining and Tom is going, but I am not going.

2. It is not raining and Tom is not going so I am not going.

3. It is not raining and Tom is going so I will go.

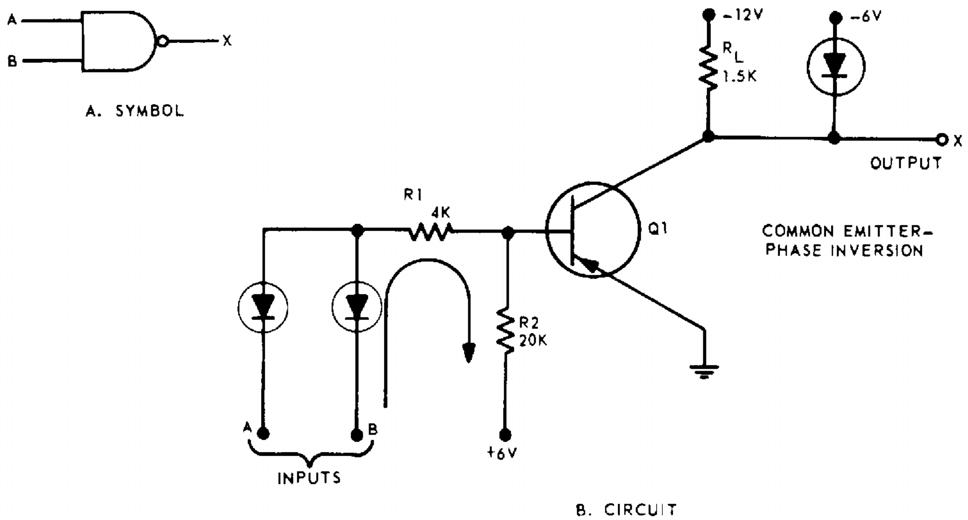
4. It is not raining and Tom is not going so I am not going.

NAND GATE. An AND symbol with a state indicator at its output, figure 5-17A, makes a NOT-AND (NAND) symbol. The state indicator on the output of the AND symbol indicates a relatively low voltage out. Thus, with two HIGH inputs, the output will be LOW.

Figure 5-17B show a NOT-AND (NAND) circuit. An AND gate at inputs A and B controls the bias on the base of amplifier Q1. The potentials applied at A and B are either zero volts, representing a high, or -6 volts, representing a low. The inputs at A and B are never in an open condition.

If a low input is applied to either diode A or B, or to both simultaneously, a total of 12 volts is established across voltage divider R1-R2. Ten volts will be dropped across the 20K ohm resistor, establishing a negative potential (-4V) at the base of Q1. This forward biases the transistor, and it conducts. Current flow through the transistor clamps the output to zero volts. Note that a low input (A or B) gives a high out. Now, consider the circuit with zero volts applied to both A and B at the same instant; we have 6 volts across voltage divider R1-R2. Five volts will be dropped across the 20K ohm resistor, establishing a base potential of +1.0V. This reverse biases the transistor and cuts it off. No current flow through the transistor now clamps the output at -6 volts. The diode in the collector circuit allows current to flow through RL holding the collector at -6 volts. The purpose of this diode is to establish the low logic level. Thus a high input at both A and B, at the same time, gives a low output.

Figure 5-17C shows the truth table for the NAND circuit which you can verify. Keep in mind that the low is -6 volts and the high is zero volts. When Q1 conducts, the output is high (zero volts); with Q1 cut off, the output is low (-6 volts). An AND function output is HIGH, but a NAND function output is LOW.



A	B	X
L	L	H
L	H	H
H	L	H
H	H	L

C. TRUTH TABLE

REP4-2662

Figure 5-17. NAND Circuit

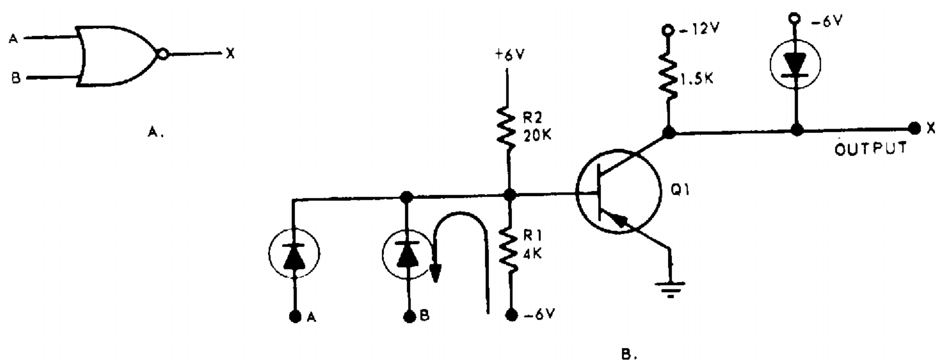
NOR GATE. An OR symbol with a state indicator at its output, figure 5-18A makes a NOT-OR (NOR) symbol. Notice that the state indicator on the output of the NORsymbol indicates a relatively low voltage out. Recall the normal output of an OR gate is HIGH with either input HIGH. This symbol indicates the output will be LOW.

Compare the NOR circuit in figure 5-18B to the NAND circuit of figure 5-17B. Note that the input diodes have been reversed and the base bias circuit has been changed.

When both input diodes have -6 volts (low) applied at the same time, the 2V drop across R1 forward biases Q1. The transistor conducts and the output is zero volts. Therefore, inputs must be low to give a high output in a NOR circuit.

Applying a high (zero volts) input to either diode will cause the transistor to cease conduction. The zero volt input at A or B will cause the corresponding diode to conduct. Now, the voltage drop across R1 will equal 6 volts, making the base and emitter of Q1 at the same potential, which cuts the transistor off. The cutoff condition gives a low (-6 volts clamped) output. The diode at the output establishes the low logic level; saturation conduction of Q1 establishes the high logic level.

Figure 5-18C shows the truth table for the NOR circuit which you can verify. Notice that the two input diodes with resistors (R1 and R2) make up the OR gate circuit. The common emitter amplifier provides phase inversion. Any HIGH input (A or B or both) causes a LOW output as represented by the NOR symbol.

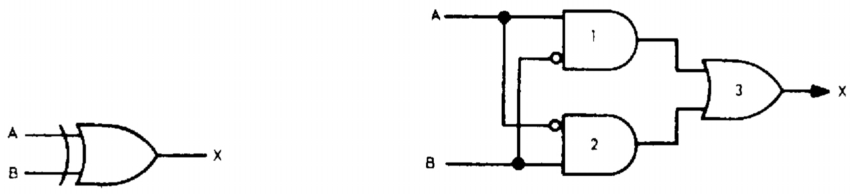


A	B	X
L	L	H
L	H	L
H	L	L
H	H	L

C.

REP4-2663

Figure 5-18. NOR Circuit



A. SYMBOL

B. LOGIC DIAGRAM

A	B	X
L	L	L
L	H	H
H	L	H
H	H	L

C. TRUTH TABLE

Figure 5-19. Exclusive OR

REP4-2664

Exclusive OR Logic

Another logic function of importance is the *exclusive OR*. Figure 5-19A shows the symbol. An exclusive OR will develop the output pulse when either input A or B is present, but not when *both* inputs are present.

Figure 5-19B shows an example of the exclusive-OR logic diagram. Notice that the exclusive OR is a combination of two inhibited ANDs and an OR symbol.

Figure 5-19C shows the truth table which you can verify. Look back to figures 5-16B and C to review the inhibit function condition for a HIGH output. Then compare the truth table with the logic diagram of figure 5-19B.