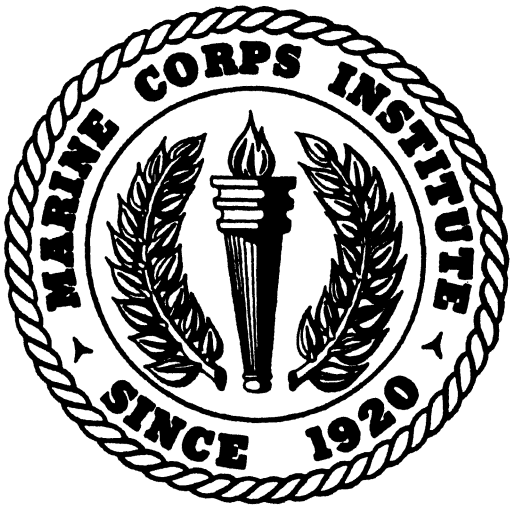
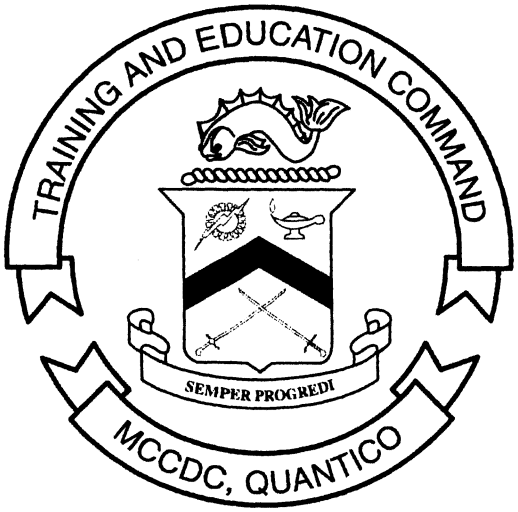
MCI 286G

MARINE CORPS INSTITUTE





FUNDAMENTALS OF DIGITAL LOGIC

MARINE BARRACKS
WASHINGTON, DC

UNITED STATES MARINE CORPS
MARINE CORPS INSTITUTE
WASHINGTON NAVY YARD
912 POOR STREET SE
WASHINGTON, DC 20391-5680

28.6g
16 Jan 98

MCI 28.6g, FUNDAMENTALS OF DIGITAL LOGIC

1. Purpose. MCI course 28.6g, Fundamentals Of Digital Logic, is published to provide instruction to Marine electronic repairmen.

2. Scope. This introductory course will serve as a primer for further digital logic study. MCI 28.6g discusses numbering systems, logic operations, and basic logic diagrams. This is the foundation of the common knowledge that the electric repairman uses to base his troubleshooting decisions on.

3. Applicability. This course is intended for instructional purposes only. It is designed for use by Marines in the ranks of Pvt through SSgt.

4. Recommendations. Comments and recommendations on the contents of the course are invited and will aid in subsequent course revisions. Please complete the course evaluation questionnaire located at the end of the text and return it to:

Director (Support Team)
Marine Corps Institute
Washington Navy Yard
912 Poor Street SE
Washington, DC 20391-5680

G. White
Lieutenant Colonel, U.S. Marine Corps
Deputy Director

ERRATUM CHANGE PAGE TO COURSE MATERIAL

1. Purpose. The purpose of this change is to give the student current instructions regarding the instructions for the Review Lesson Examination page.

2. Action. Change the instructions found on page R-1 of this book to read as follows:

"The purpose of the review lesson examination is to prepare you for your final examination. We recommend that you try to complete your review lesson examination without referring to the text, but for those items (questions) you are unsure of, restudy the text. When you finish your review lesson and are satisfied with your responses, check your responses against the answers provided at the end of this review lesson examination.

Select the ONE answer that BEST completes the statement or that answers the item. For multiple choice items, circle your response. For matching items, place the letter of your response in the space provided."

3. This page is to be filed directly behind the Promulgation Letter of this course.

FUNDAMENTALS OF DIGITAL LOGIC

CONTENTS

CONTENTS continued

Student Information

Number and Title	MCI 286G FUNDAMENTALS OF DIGITAL LOGIC
Study Hours	10
Course Materials	Text
Review Agency	Marine Corps Communication-Electronics School Marine Corps Air Ground Combat Center Twentynine Palms, California 92278-5020
Reserve Retirement Credits (RRC)	3
ACE	Not applicable to civilian training/education
Assistance	For administrative assistance, have your training officer or NCO log on to the MCI home page at www.mci.usmc.mil. Marines CONUS may call toll free 1-800-MCI-USMC. Marines worldwide may call commercial (202) 685-7596 or DSN 325-7596.

Study Guide

Congratulations

Congratulations on your enrollment in a distance education course from the Distance Learning and Technologies Department (DLTD) of the Marine Corps Institute (MCI). Since 1920, the Marine Corps Institute has been helping tens of thousands of hard-charging Marines, like you, improve their technical job performance skills through distance learning. By enrolling in this course, you have shown a desire to improve the skills you have and master new skills to enhance your job performance. The distance learning course you have chosen, MCI course 286G, *Fundamentals of Digital Logic*, is the foundation course of digital electronics. It is designed to be a no-tears introductory course for Marines tasked with electronic troubleshooting duties.

Your Personal Characteristics

- **YOU ARE PROPERLY MOTIVATED.** You have made a positive decision to get training on your own. Self-motivation is perhaps the most important force in learning or achieving anything. Doing whatever is necessary to learn is motivation. You have it!

- **YOU SEEK TO IMPROVE YOURSELF.** You are enrolled to improve those skills you already possess, and to learn new skills. When you improve yourself, you improve the Corps!

- **YOU HAVE THE INITIATIVE TO ACT.** By acting on your own, you have shown you are a self-starter, willing to reach out for opportunities to learn and grow.

- **YOU ACCEPT CHALLENGES.** You have self-confidence and believe in your ability to acquire knowledge and skills. You have the self-confidence to set goals and the ability to achieve them, enabling you to meet every challenge.

- **YOU ARE ABLE TO SET AND ACCOMPLISH PRACTICAL GOALS.** You are willing to commit time, effort, and the resources necessary to set and accomplish your goals. These professional traits will help you successfully complete this distance learning course.

Continued on next page

Study Guide, Continued

Beginning Your Course

Before you actually begin this course of study, read the student information page. If you find any course materials missing, notify your training officer or training NCO. If you have all the required materials, you are ready to begin.

To begin your course of study, familiarize yourself with the structure of the course text. One way to do this is to read the table of contents. Notice the table of contents covers specific areas of study and the order in which they ar presented. You will find the text divided into several study units. Each study unit is comprised of two or more lessons, lesson exercises, and finally, a stud unit exercise.

Leafing Through the Text

Leaf through the text and look at the course. Read a few lesson exercise questions to get an idea of the type of material in the course. If the course ha additional study aids, such as a handbook or plotting board, familiarize yourself with them.

The First Study Unit

Turn to the first page of study unit 1. On this page, you will find an introduction to the study unit and generally the first study unit lesson. Study unit lessons contain learning objectives, lesson text, and exercises.

Reading the Learning Objectives

Learning objectives describe in concise terms what the successful learner, you, will be able to do as a result of mastering the content of the lesson text. Read the objectives for each lesson and then read the lesson text. As you rea the lesson text, make notes on the points you feel are important.

Completing the Exercises

To determine your mastery of the learning objectives and text, complete the exercises developed for you. Exercises are located at the end of each lesson, and at the end of each study unit. Without referring to the text, complete the exercise questions and then check your responses against those provided.

Continued on next pa

Continuing to March

Continue on to the next lesson, repeating the above process until you have completed all lessons in the study unit. Follow the same procedures for each study unit in the course.

Preparing for the Final Exam

To prepare for your final exam, you must review what you learned in the course. The following suggestions will help make the review interesting and challenging.

- **CHALLENGE YOURSELF.** Try to recall the entire learning sequence without referring to the text. Can you do it? Now look back at the text to see if you have left anything out. This review should be interesting. Undoubtedly, you'll find you were not able to recall everything. But with a little effort, you'll be able to recall a great deal of the information.

- **USE UNUSED MINUTES.** Use your spare moments to review. Read your notes or a part of a study unit, rework exercise items, review again; you can do many of these things during the unused minutes of every day.

- **APPLY WHAT YOU HAVE LEARNED.** It is always best to use the skill or knowledge you've learned as soon as possible. If it isn't possible to actually use the skill or knowledge, at least try to imagine a situation in which you would apply this learning. For example make up and solve your own problems. Or, better still, make up and solve problems that use most of the elements of a study unit.

- **USE THE "SHAKEDOWN CRUISE" TECHNIQUE.** Ask another Marine to lend a hand by asking you questions about the course. Choose a particular study unit and let your buddy "fire away." This technique can be interesting and challenging for both of you!

- **MAKE REVIEWS FUN AND BENEFICIAL.** Reviews are good habits that enhance learning. They don't have to be long and tedious. In fact, some learners find short reviews conducted more often prove more beneficial.

Continued on next page

Tackling the Final Exam

When you have completed your study of the course material and are confident with the results attained on your study unit exercises, take the sealed envelope marked **"FINAL EXAM"** to your unit training NCO or training officer. Your training NCO or officer will administer the final examination and return the examination and the answer sheet to MCI for grading. Before taking your final examination, read the directions on the DP-37 answer sheet carefully.

Completing Your Course

The sooner you complete your course, the sooner you can better yourself by applying what you've learned! HOWEVER--you do have 2 years from the date of enrollment to complete this course.

Graduating!

As a graduate of this distance education course and as a dedicated Marine, your job performance skills will improve, benefiting you, your unit, and the Marine Corps.

Semper Fidelis!

STUDY UNIT 1

BINARY NUMBERING SYSTEM

Introduction. A numbering system provides the framework in which we represent numerical quantities as well as perform mathematical computations. The numbering system we are most familiar with is a positional numbering system, the decimal system. This system uses the Arabic numerals 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. These numerals are also known as symbols or digits. While some systems may use more than just these ten symbols and others less, the system we will study in this course, the binary system, uses only two symbols. The binary numbering system is the heart of digital logic. You must understand the binary numbering system if you are to successfully troubleshoot digital electronic equipment.

Lesson 1. NUMBERING SYSTEM BASICS

LEARNING OBJECTIVES

1. Identify in writing the three components of a positional numbering system.
2. Given the base of a numbering system, select the maximum number of digits that system contains.
3. Determine the base of any given number in a positional numbering system.
4. State in writing the base of a numbering system that uses two digits.
5. Given the base of any three-digit number, determine the place value for each digit.

1101. Base, Point, and Place Value

The positional numbering system you encounter daily is the base ten numbering system. What makes this system the base ten system? What makes this system a positional system? As you learn to answer these questions, you'll learn the basics of all numbering systems and be able to apply this knowledge to help you understand the binary numbering system. Let's take a closer look at numbers. Basically (at least for the purposes of this course) there are three parts to every number: base or radix, point, and place value. The following explains these three parts.

a. Base. The base or radix is the logical place to begin to explain a numbering system--as this is what a positional numbering system is built on.

(1) The base of a number determines the number of symbols in that system. For example, the decimal system, base ten, uses ten digits: 1, 2, 3, 4, 5, 6, 7, 8, 9, and 0. The octal numbering system, base eight, uses eight digits: 1, 2, 3, 4, 5, 6, 7, and 0. How many digits are used in the binary, base two, numbering system? That's right, two--1 and 0. Remember, digits, symbols, and numerals have the same meaning in this context.

(2) Within a numbering system, no single digit can represent a quantity equivalent to or higher than its base. Notice that there is no digit higher than 7 in base eight, no digit higher than 9 in base 10, etc. This will be explained later in the text.

(3) The base of a number when shown is written as a subscript to that number. For example, the number 18_{10} is read "one eight base ten." Normally the base will not be shown for a base 10 number. If no base is shown, assume base 10. This number 72_9 would be read "seven two base nine." Okay, now it's your turn. How would you read the number 23_6? If you said, "two three base six", you're right. We'll look at one more and then go on. How would you read the number 514? Couldn't fool you? Yes, this number is read "five one four base ten" or simply five hundred fourteen. Remember, if no base is stated, assume base ten.

(4) The base also allows you to know the place value for each digit, another concept which you will learn in this lesson.

Okay, now it's time to see what you've learned--what does the base tell you about a number?

Your answer should have included the following points:

- The base determines the number of digits used in that system.
- No single digit represents a quantity higher than or equivalent to the base.
- The base allows you to determine place value for each digit.

b. Point. The point of a numbering system divides whole numbers from fractions and takes the name of the system that it is in. As an example, the point in the decimal system is known as the decimal point, something you're sure to be familiar with. Looking at the decimal (base ten) number 134.75, whole numbers are always to the left of the point (in this case--decimal point) and fractions are always to the right of the point. In the binary number 101.11_2 the same relationship exists--whole numbers to the left, and fractions to the right of the binary point. Examples of points for other systems are the ternary point (base three), the octal point (base eight), and so on.

c. Place value. Place value describes the space (or location) that a number resides in. Just as every numeral has a value, every position in which a digit is placed also has a value.

(1) Let's look at the decimal number 105.0. Starting at the decimal point and going left, the first digit of this number tells you how many ones are in that number, the second how many tens, the third how many hundreds. The digit to the right of the decimal tells you how many tenths the number contains. Notice that the 0 serves as a place holder; that is, the 0 has no numerical value but simply allows the 1 and the 5 to maintain their proper place. Exploring further, you can see that each place is a power of the base. This means that (for base ten) starting at the decimal point and going left--the first place is 10^0 or ones, the second place is 10^1 or tens, the third place 10^2 or one hundreds, while the place to the right of the decimal point is 10^{-1} or tenths (fig 1-1).

10^4	10^3	10^2	10^1	10^0	10^{-1}
10000	1000	100	10	1	.1

Fig 1-1. Place values for base ten.

A quick review of exponents (powers): The exponent is written as a superscript to the number (N^{power}). The power tells you how many times to multiply the number by itself. As an example, N^3 means N x N x N and is read N raised to the third power or simply N to the third power. N^2 means N x N and is read N to the second power. Further, any number raised to the zero power equals 1 ($N^0 = 1$) and any number raised to the first power equals that number ($N^1 = N$). Negative exponents in this form N^{-x} mean $1/N^x$. As an example, 5^{-2} (read five to the negative second power), equals $1/5^2$ or 1/25 (1/{5 x 5}).

This type of relationship, which exists for all numbering systems, can be seen in table 1-1. As an example, using the point as reference and going left, the place values for the first few places in base three are $1(3^0)$, $3(3^1)$, $9(3^2)$, and $27(3^3)$; in base five $1(5^0)$, $5(5^1)$, $25(5^2)$, and $125(5^3)$. In table 1-1 you can also see the place values for other numbering systems with base 2 (binary numbering system) highlighted.

(2) The pattern found to the left of the point is also found to the right. However, place values to the right of the point are fractional values. The difference is simply that powers of the base to the right of the point are negative powers. This is illustrated in the number 25.817. The place values, starting at the decimal point and going left, would be 10^0 and 10^1; then from the point and to the right, 10^{-1}, 10^{-2}, and 10^{-3}.

10^1	10^0	10^{-1}	10^{-2}	10^{-3}
2	5	.8	1	7

Accordingly, in the number 111.01_2 the digits to the left of the binary point represent whole numbers while those to the right of the point represent fractions.

What is the place value that each of these digits holds (in order from left to right)?

If you said 2^2, 2^1, 2^0, 2^{-1}, and 2^{-2}, you're right.

Table 1-1 shows examples of place values in other bases.

Table 1-1. Place Values For Bases Two Through Ten

Binary: (Base Two)

base^{power}	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}
equivalent value	**16**	**8**	**4**	**2**	**1**	**.5**	**.25**

Ternary: (Base Three)

base^{power}	3^4	3^3	3^2	3^1	3^0	3^{-1}	3^{-2}
equivalent value	81	27	9	3	1	.33	.11

Quaternary: (Base Four)

base^{power}	4^4	4^3	4^2	4^1	4^0	4^{-1}	4^{-2}
equivalent value	256	64	16	4	1	.25	.06

Quinary: (Base Five)

base^{power}	5^4	5^3	5^2	5^1	5^0	5^{-1}	5^{-2}
equivalent value	625	125	25	5	1	.2	.04

Senary: (Base Six)

base^{power}	6^4	6^3	6^2	6^1	6^0	6^{-1}	6^{-2}
equivalent value	1296	216	36	6	1	.17	.03

Septenary: (Base Seven)

base^{power}	7^4	7^3	7^2	7^1	7^0	7^{-1}	7^{-2}
equivalent value	2401	343	49	7	1	.14	.02

Octonary: (Base Eight)

base^{power}	8^4	8^3	8^2	8^1	8^0	8^{-1}	8^{-2}
equivalent value	4096	512	64	8	1	.125	.016

Nonary: (Base Nine)

base^{power}	9^4	9^3	9^2	9^1	9^0	9^{-1}	9^{-2}
equivalent value	6561	729	81	9	1	.11	.012

Decimal: (Base Ten)

base^{power}	10^4	10^3	10^2	10^1	10^0	10^{-1}	10^{-2}
equivalent value	10000	1000	100	10	1	.1	.01

1102. Binary Numbers

Now that you know the basics of a positional numbering system, it's time to concentrate on binary numbers. A knowledge of other numbering systems may be helpful when troubleshooting some digital equipment, but a basic understanding of the fundamentals of binary will help you go a long way in understanding any system.

a. Common characteristics. Looking closely at the binary system, you see that it is no different in structure than the numbering systems we've already looked at. Just as there is no single digit in base ten to represent the quantity ten, there is no single digit in base two to represent the quantity two. All positional numbering systems share this same characteristic. So, in the binary system there are just ones and zeros. Any system with only two digits has to be easy!

b. Application. Let's look at a binary number to see if you understand them as well as you think. Using the binary number 1011.0 you should be able to apply the basics of what you've learned so far.

> Note: This number has no subscript, but for the purposes of this course, consider it a binary number. The dropping of the subscript is a convention used for ease of reading that will be employed in this text when numbers of different numbering systems are not mixed.

So, getting back to your number (1011.0)... we know the base is two; the place values are 2^0, 2^1, 2^2, 2^3, and 2^{-1}, and that the point is the binary point.

2^3	2^2	2^1	2^0	2^{-1}
1	0	1	1	.0

(1) Going further, it is easy to see the value of each digit. The highlighted digit in 1011.0 is in the 2^3 position, so to find the value of that digit we could multiply the digit by the place value of its position. Simply put, this highlighted digit 1011.0 would mean 1 x 2^3 or 8. The next digit 1011.0 means 0 x 2^2 or 0. The next digit, 1011.0, means 1 x 2^1 or 2. The last whole number digit, 1011.0, means 1 x 2^0 or 1. The digit representing a fractional value 1011.0 means 0 x 2^{-1} or 0.

Your turn again: What are the place values and the value of each of the digits in 11101.0_2? (Table 1-1 may help.)

Yours answers should resemble these:			
$1_{1}1101.0_2$	place value 16 (2^4)	digit value 1 x 2^4	or 16
$1_{1}101.0_2$	place value 8 (2^3)	digit value 1 x 2^3	or 8
$11_{1}01.0_2$	place value 4 (2^2)	digit value 1 x 2^2	or 4
$111_{0}1.0_2$	place value 2 (2^1)	digit value 0 x 2^1	or 0
$1110_{1}.0_2$	place value 1 (2^0)	digit value 1 x 2^0	or 1
11101.0_{2}	place value .5 (2^{-1})	digit value 0 x 2^{-1}	or 0

(2) Determining these values in the binary numbering system is relatively easy to do because of the small number of digits (ones or zeros) that you would find in each position. Of course this concept holds true for all positional numbering sytems; there are just larger numbers to deal with. This way of looking at numbers will be helpful when we try to convert from one system to another.

Lesson Summary. This lesson introduced you to positional numbering systems. The emphasis on base two will help, not only as you complete this course, but as you troubleshoot electronic equipment.

- -

Exercise: Complete items 1 through 6 by performing the action required. Check your responses against those listed at the end of this study unit.

1. The number 10_{10} may be read as one zero base

   a. two.
   b. eight.
   c. ten.
   d. sixteen.

2. What is the maximum number of different digits used in the base eight numbering system?

   a. Seven
   b. Eight
   c. Nine
   d. Sixteen

3. What numbering system uses only two digits?

   a. Binary
   b. Quinary
   c. Octal
   d. Decimal

4. The place value for each digit of 110_2 in order from left to right is

   a. eight, four, two.
   b. eight, four, one.
   c. four, two, one.
   d. four, two, zero.

5. The three parts to the binary numbering system are _____, _____, and _____.

6. In the binary system the _____ _____ separates fractions from whole numbers.

Lesson 2. BINARY MATHEMATICAL OPERATIONS

LEARNING OBJECTIVES

1. Given any binary number, select the decimal equivalent for that number.

2. Given any decimal number, select the binary equivalent for that number.

3. Given any single-digit two-term binary multiplication problem, state the solution in writing.

4. Given any single-digit two-term binary subtraction problem, state the solution in writing.

5. Given any single-digit two-term binary addition problem, state in writing the solution.

1201. Binary to Decimal Conversion

There are more than a few ways to convert binary to decimal. We will concentrate on the easiest method. Learning to convert numbers from one base to another is more than just a mathematical exercise. The purpose of conversion is to aid you in troubleshooting digital equipment.

a. Basics. First let's look at the binary number 152.0. If you said that can't be a binary number, again you're right. The largest digit in binary is 1, so, of course, 152.0 can't be a binary number. So, going over what we know of the binary system, we can see that the...

(1) ... base is two; as is the radix--remember base and radix are the same.

(2) ... point is the binary point and divides the fractional part of the number from the whole numbers.

(3) ... place value is a function of the powers of the base. Put another way, place value can be easily determined by using the point as a reference. Going left from the point, the first place value is the base raised to the zero power, the next place is the base raised to the first power, the next place is the base raised to the second power, and so on. Going right from the point, the first place is the base raised to the negative first power, the next place is the base raised to the negative second power, the third place would be the base raised to the negative third power, and so on.

b. Digit values. Now try the decimal number 152.0. Starting at the decimal point and going left, look at the place values 10^0, 10^1, and 10^2. There is one place to the right of the decimal point and that is 10^{-1}. These values are the same as ones, tens, and hundreds going left and tenths going right. Now look at the value of the digits. **THESE VALUES WOULD BE THE DIGIT MULTIPLIED BY THE PLACE VALUE THE DIGIT IS IN.** The following box illustrates this:

Digit		Place Value		Digit Value
1	x	10^2	or 100	100
5	x	10^1	or 10	50
2	x	10^0	or 1	2
0	x	10^{-1}	or .1	0

Now if you were to add the values of the digits, you'd find you had the value of the number. In other words, 100 + 50 + 2 + 0 = 152. Now we'll apply this to binary numbers.

c. Applications. Okay, let's look at the binary number 1101.0, and apply this same concept.

(1) A look at the place values from the binary point left shows a one (2^0), a two (2^1), a four (2^2), and an eight (2^3). From the binary point right we have .5 (2^{-1}). If you now look at the digit values (place value times digit) in this same order (binary point and left) you would have one, zero, four, eight, and zero. This is illustrated as follows:

Digit		Place Value			Digit Value
1	x	2^3	or	8	8
1	x	2^2	or	4	4
0	x	2^1	or	2	0
1	x	2^0	or	1	1
0	x	2^{-1}	or	.5	0

(2) If you again add the digit values, you'll find you have the value of the number, 8 + 4 + 0 + 1 + 0 = 13. Not only do you have the value of the number, but you've converted a binary number to a decimal number. As you can see, it's easy.

(3) If we use a chart similar to this one, this is seen somewhat easier.

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

The chart above is similar to the one in lesson one. If we put our number 1101.0 on the bottom of the chart in proper place value--

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125
	1	1	0	1	0		

8 + 4 + 0 + 1 + 0 = 13

--as shown it is easy to discern the decimal value of the binary number 1101.0. Looking at two more binary numbers 11111.1 and 1001.01, it's easy to see the decimal equivalents, once the binary numbers are placed on the following charts:

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125
1	1	1	1	1	1		

16 + 8 + 4 + 2 + 1 + .5 = 31.5

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125
	1	0	0	1	0	1	

8 + 0 + 0 + 1 + 0 + .25 = 9.25

(4) Now you try these two binary numbers, 10000.11 and 10101.01 on your own. You can use the two blank charts provided if you wish.

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

You should have gotten 16.75 and 21.25 as the decimal equivalents. If you used the charts to convert, they should have looked like the two that follow:

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125
1	0	0	0	0	1	1	

16 + 0 + 0 + 0 + 0 + .5 + .25 = 16.75

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125
1	0	1	0	1	0	1	

16 + 0 + 4 + 0 + 1 + 0 + .25 = 21.25

If you didn't get the answers above, you need to review the study unit up to this point. At any rate, try one more conversion: what is the decimal equivalent of this binary number 10111.101? Converting from binary to decimal is easy in this manner. The following perforated pages can be pulled out for use in conversions on the job. By the way, did you get 23.625 as the answer to that last problem?

(This page intentionally left blank.)

(5) Pull out conversion charts.

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

1202. Decimal to Binary Conversion

As with converting binary to decimal, there are different methods of converting decimal to binary. The method you will use will be straightforward, simple, and based on the skills you've acquired in this text.

a. Subtraction method. As you might expect, place value and digit value are the determinants in this conversion process as they were in the previous process. The subtraction method is easy to use because the base is so small in the binary numbering system. Again the use of the previously introduced handy chart will prove helpful. The conversion is accomplished in three steps.

(1) Steps. This is a simple three step-method:

STEP	SUBTRACTION METHOD
1	Determine the largest binary place value that can be subtracted from the decimal you want to convert.
2	Subtract the base two place value from the decimal number.
3	If there is a remainder, repeat the first two steps until the desired accuracy is reached.

It is during step one that the digits of your new number will be determined. Each successive place value that can be subtracted from the decimal number will be a 1 in the new number, and each place value that is skipped will be a 0.

(2) Examples. Examples of this method will make it easier to see. It's time again for that famous chart so you can see the place values easily.

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

Let's try lucky 13 as our first decimal number to convert to binary. Step one tells you to determine the largest binary place value you can subtract from the decimal number. The number 2^4, or 16, is too big to subtract from 13, but 2^3, or 8, the next place value will work. A 1 goes in the 2^3 or 8 place value column--this is the first digit of the binary number.

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125
	1						

Our next step is to subtract the selected number (large binary place value that could be subtracted), thus 13 - 8 = 5. The final step is to repeat the first two steps if there is a remainder. Since we have a remaind of 5 in this example, it's time to look for the largest place value that we can subtract from 5 (step one) and subtract it (step two). It looks as though 2^2 or 4 "ha the duty." A 1 goes in the 2^2 or 4 place value column--the second digit of the new number.

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125
	1	1					

Subtracting again gives us 5 - 4 = 1. Step three tells us what to do with a remainder--go to step one. The largest place value that can be subtracted from the decimal one is 2^0 or 1. Notice that the 2^1 or 2 place column is skipped, so a zero is placed there. However, the 2^0 or 1 place value column was used, so a 1 goes there.

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125
	1	1	0	1			

Then following step two we subtract, 1 - 1 = 0. Now because there is no remainder, the process is finished. Looking at your completed chart, you can see that you've converted decimal 13 to binary 1101_2.

How could you check this answer?

You could check your answer in two ways:

(a) Add the digit values of the binary number you've jus derived (8 + 4 + 0 + 1 = 13); this is taken directly off your chart.

(b) Look back to 1201c on pages 9 and 10 at the first binary to decimal conversion example you did.

Though this method is easy, one more example is in order before it's your turn. Let's try converting the decimal number 16.75 to its binary equivalent. Step one, find the largest binary place value you can subtract from 16.75, your decimal number. Looking at your chart, you should be able to see that the number you need is 16 (2^4). Now annotate a 1 in the 2^4 or 16 place value position.

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125
1							

Step two, subtract the selected number; 16.75 - 16 = .75. Step three, go to step one. Looking at what you have left, .75, you should be able to tell that the next place value you can subtract is 2^{-1} or .5. That means you skipped 2^3 or 8, 2^2 or 4, 2^1 or 2, and 2^0 or 1. Therefore, a 0 goes in those positions and a 1 goes in the 2^{-1} position.

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125
1	0	0	0	0	1		

Now you are ready for step two. Simple subtraction, .75 - .5 shows that you have .25 as a remainder. Step three tells you that if you have a remainder, go to step one. The next place value you can subtract is 2^{-2} (.25). Annotate your chart.

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125
1	0	0	0	0	1	1	

Go to step two, .5 - .5 = 0. No step three is needed because there was no remainder. Your binary number then is 10000.11_2. Let's check this answer by using the chart. First find, then add the binary digit values, 16 + 0 + 0 + 0 + 0 + .5 + .25 = 16.75.

b. Practice. Try converting 31.5, 12, and 14. If you need to, use the pull-out chart from 1201c(5) a few pages back to help you as you convert. If your answers don't match the ones shown, reread 1202. Your answers should be as follows:

$31.5 = 11111.1_2$ $\quad 12 = 1100.0_2$ $\quad 14 = 1110.0_2$

Yes, these examples were easy, but it's not the complexity of the problems that's important; it's the application of the principles.

1203. Binary Multiplication, Subtraction, Addition, and Division

Digital equipment performs mathematical operations differently from the way you do using paper and pencil. Digital equipment does all mathematical operations through addition of binary numbers. Done on paper these operations are long and tedious. Thus, the actual machine operations are outside the scope of this course. Nevertheless, you will explore the basic rules for the multiplication, subtraction, addition, and division of binary numbers. This is not a mathematical exercise--these rules will help you to better troubleshoot digital equipment.

Without a knowledge of these rules, perhaps the easiest way to perform these operations is to convert the numbers into decimal, do the operation, then convert the answer back to binary. But once you know the rules you no longer need to perform these extra steps. This knowledge is just as important to you as any tool in your repair kit. Now, in the interest of sharpening your troubleshooting skills, a discussion of the rules of binary mathematical operations follows.

a. Multiplication. Once you know the rules, multiplying binary numbers is easy. These are the rules:

$$0 \times 0 = 0$$
$$1 \times 0 = 0$$
$$0 \times 1 = 0$$
$$1 \times 1 = 1$$

This multiplication table should look very familiar to you. These values are the same as in the decimal system, so let's go on to the other operations.

b. Subtraction. In binary subtraction, the following basic rules apply:

$$0 - 0 = 0$$
$$1 - 0 = 1$$
$$1 - 1 = 0$$
$$0 - 1 = 1 \text{ borrow } 1$$

(1) These binary subtraction rules also should look familiar, until you get to that last equation, **0 - 1 = 1 borrow 1.** Binary subtraction in one instance does require you to borrow from the next highest place value column in order to complete a subtraction. That one instance is 0 - 1, and the rule covering this is 0 - 1 = 1 borrow 1. Now To see what is happening here, look at this decimal example:

$$40 - 21 =$$

To solve this problem, you would first attempt to subtract one from zero. But you can't do that, so you borrow from the next highest place value column. In this case, you borrow a ten from the tens place value column.

Now you would be able to subtract one from ten, leaving you a nine in the ones column of your answer. Next you would subtract twenty from what used to be forty--but is now thirty (since you borrowed ten from it)--leaving you a one in the tens column of your answer. Your computations might look something like this:

```
 3 10
 ∅0
-21
 19
```

(2) Now that you've seen the rules and a decimal example of borrowing--let's try two binary examples:

```
 101        100
-_11       -__1
```

In case you've forgotten binary place values, here's a chart to work with:

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
16	8	4	2	1	.5	.25	.125

In the first example, the first binary subtraction is one minus one. According to the rules, 1 - 1 = 0.

```
101
-11
```

This was done and a zero was placed in the ones place position of the answer.

```
101
-11
 0
```

The second subtraction of the first example is zero minus one. According to the rules 0 - 1 = 1 borrow 1.

```
101
-11
```

So, borrow one from the next higher place column (four's place or 2^2) and place a one in the two's column of the answer.

```
  1
 1∅1
-_11
  10
```

Let's look at what you did. The digit one you borrowed had a digit value of four. You then subtracted a one with a digit value of two from the one (digit value four) you borrowed. Your answer was a one with a digit value of two. Simply, four minus two equals two.

In the second example, your first binary subtraction is zero minus one (0 - 1).

```
 100
-  1
```

Again, according to the rules, 0 - 1 = 1 borrow 1, so you attempt to borrow one from the next higher place value (two's place). But because that column has a zero, you have to borrow from the next highest place (four's place).

```
    10
  1ø ø
-   1
    1
```

This done, you place a one in the one's place of your answer and go to the next subtraction.

This subtraction is one minus zero (1 - 0). The rule for this subtraction is 1 - 0 = 1. This done, you place a one in the two's column of your answer. The reason that this second subtraction procedure is one minus zero is because you went two places to borrow the one in your first subtraction. So what you have actually done is taken a four (borrowed a one with a digit value of four) and distributed it between the two's and one's columns. You put two (10_2) in the one's column so you could make your first subtraction. Because you still had two left over, you put a two (one with a digit value of two) in the two's column. If all the computations were to be shown, it would look like this:

```
  1 10
  1ø ø
-   1
   11
```

c. Addition. In binary addition, the following basic rules apply:

```
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0  carry 1
```

These rules are also familiar, at least until that last equation, 1 + 1 = 0 carry 1. In the decimal system one plus one equals two; however, that digit doesn't exist in the binary numbering system. A count higher than one means some value must be carried over to the next highest place value. Another decimal example:

19 + 21 =

To solve this problem you would first add one and nine and get ten. This is fine, but because you can't get ten, (two digits) in one place value column, you put a zero in the ones column of

your answer and carry the ten into the ten's place value column. Now add the ten you carried over along with the ten (from the nineteen) and the twenty (from the twenty-one) together to get forty. Lastly, put a four in the ten's column of your answer and go on your merry way. Your computations might look something like this:

```
  1
  19
+ 21
  40
```

A simple binary addition problem to illustrate this concept:

1 + 1 =

If you follow the rule of 1 + 1 = 0 carry 1, you'd put a zero in the one's column of your answer, and carry 1 over to the two's column. This carry would be brought down into the two's column of your answer. The computation might look something like this:

```
  1
  1
+ 1
  10
```

d. Division. Division in binary is a fairly straightforward process. The following rules for binary division apply:

0 ÷ 0 = 0
0 ÷ 1 = 0
1 ÷ 1 = 1

As in binary multiplication, there's nothing new or unusual to report concerning the rules for binary division.

Here's a chart to help you keep those rules straight.

MULTIPLICATION	SUBTRACTION	ADDITION	DIVISION
0 x 0 = 0 1 x 0 = 0 0 x 1 = 0 1 x 1 = 1	0 - 0 = 0 1 - 0 = 1 1 - 1 = 0 0 - 1 = 1 borrow 1	0 + 0 = 0 0 + 1 = 1 1 + 0 = 1 1 + 1 = 0 carry 1	0 ÷ 0 = 0 0 ÷ 1 = 0 1 ÷ 1 = 1

Lesson Summary. This lesson introduced you to binary mathematical operations. You should now have the skill and knowledge to convert between binary and decimal as well as to add, subtract, divide, and multiply binary numbers. These skills are the heart of digital logic.

Exercise: Complete items 1 through 7 by performing the action required. Check your responses against those listed at the end of this study unit.

1. What is the decimal equivalent of 1110.1_2?

   a. 13.5
   b. 14.5
   c. 15.1
   d. 16.125

2. What is the decimal equivalent of 10001.01_2?

   a. 14.125
   b. 16.25
   c. 17.25
   d. 19.25

3. What is the binary equivalent of 31.75?

   a. 1111.11_2
   b. 11001.011_2
   c. 11110.11_2
   d. 11111.11_2

4. What is the binary equivalent of 9.125?

   a. 1001.001_2
   b. 1010.001_2
   c. 1011.001_2
   d. 1011.011_2

5. In the space provided below write the rules for binary multiplication.

6. In the space provided below write the rules for binary subtraction.

7. In the space provided below write the rules for binary addition.

_____

_____

_____

_____

UNIT SUMMARY

By learning the basics of digital logic in this study unit, the binary numbering system, you have mastered the foundation for the following study units. Your next study unit will introduce you to the building blocks of digital logic, logic gates.

Lesson 1 Exercise Solutions

		Reference
1.	c.	1101a
2.	b.	1101a
3.	a.	1101a
4.	c.	1102b
5.	base, point, place value	1101
6.	binary point	1101b

Lesson 2 Exercise Solutions

		Reference
1.	b.	1201
2.	c.	1201
3.	d.	1202
4.	a.	1202
5.	0 x 0 = 0 1 x 0 = 0 0 x 1 = 0 1 x 1 = 1	1203a
6.	0 - 0 = 0 1 - 0 = 1 1 - 1 = 0 0 - 1 = 1 borrow 1	1203b
7.	0 + 0 = 0 0 + 1 = 1 1 + 0 = 1 1 + 1 = 0 carry 1	1203c

STUDY UNIT 2

LOGIC OPERATIONS

Introduction. In the previous study unit, you were introduced to the binary numbering system. You will use this knowledge to understand the logic operations and the logic gates that perform these logic operations.

The building blocks of digital electronic circuits are logic gates. Each gate performs a specific logic function. It is the arrangements of these simple gates that determine the difference between a computer and a radio. Understanding these logic gates is essential if you are to sucessfully repair digital electronic equipment. The gates themselves are few and simple, but the combinations of these simple building blocks may be staggeringly complex. Each type of gate is unique in the function that it performs. This study unit will give you the knowledge to identify these different gates as well as understand their different functions.

Lesson 1. BASIC LOGIC OPERATIONS

LEARNING OBJECTIVES

1. State in writing the three basic logic operations.
2. From a chart of algebraic logic symbols, select the **OR** symbol.
3. From a chart of schematic logic symbols, select an **OR** symbol.
4. Given a labeled **OR** gate, state in writing the equation describing that gate.
5. Given a labeled **OR** gate, state in writing the truth table derived from that gate.
6. From a chart of algebraic logic symbols, select the **AND** symbol.
7. From a chart of schematic logic symbols, select an **AND** symbol.
8. Given a labeled **AND** gate, state in writing the equation describing that gate.
9. Given a labeled **AND** gate, state in writing the truth table derived from that gate.

2101. The Three Basic Logic Operations

There are three basic logic operations: **AND**ing, **OR**ing, and complementing. These operations both singly and combined form the circuits of digital electronic equipment. The operations are performed by electronic switches known as logic gates. In this lesson, the three basic logic functions will be explained to you.

What are the basic logic operations?

Your answer should have been the following: **AND**ing, **OR**ing, and complementing

2102. Logic Levels

The electronic switches known as gates may have one or more inputs. Regardless of the number of inputs, these gates will have but one output. There are only two voltage levels associated with these inputs and outputs. The voltage levels found in most digital electronic equipment are five volts and zero volts. These two voltage levels are represented by ones and zeros and are called logic conditions, states, or levels. The more positive voltage is represented by a one, and the more negative voltage is represented by a zero. This set of conditions, where the higher voltage is represented by a one and the lower voltage by a zero, is known as positive logic.

During this lesson you will be using positive logic only. Further, the voltage represented by a one is known as the <u>true</u> condition, the <u>high</u> condition, or the <u>on</u> condition. The voltage represented by the zero is known as the <u>false</u> condition, the <u>low</u> condition, or the <u>off</u> condition. Let's review this important information:

* Logic gates are electronic switches that perform electronic logic operations.

* Logic gates may have more than one input, but only one output.

* Digital electronic equipment uses two voltage levels.

* The voltage levels are known as logic levels, logic states, or conditions.

* A high, on, true, or a one are associated with the higher voltage. (remember, positive logic only for now)

* A low, off, false, or a zero are associated with the lower voltage. (positive logic)

The condition or state of a logic gate is determined by its output level. Logic gates are either on or off, in a true condition or a false condition, in an active state or an inactive state, showing a logic high or a logic low. Again, level, state, and condition all mean the same thing--a description of the voltage level of a particular input or output. But, the logic level of the output of a gate is also a description of the state of that gate. Figure 2-1 summarizes these levels.

ONE	ZERO
HIGH	LOW
ON	OFF
ACTIVE	INACTIVE
TRUE	FALSE

Fig 2-1. Logic levels or states.

How is the state of a gate determined?

Right, the level of the output of the gate determines the state of that gate. That is, if the output of a certain gate is a high, then the state or condition of that gate is active, true, on, or high.

Now, let's discuss the actual logic functions and their associated gates.

2103. **OR** Function

The **OR** function acts like switches in parallel (fig 2-2).

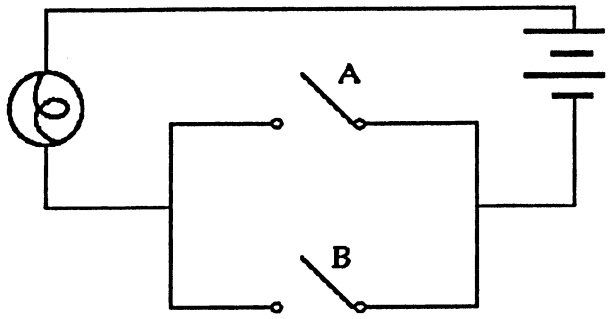


Fig 2-2. **OR** Function.

In this circuit the path for current flow is either through switch A or switch B. The **OR** function states simply that if any or all of its inputs are true, then the output is true. The output is false only when all the inputs are false. This will be explained to you as you complete this lesson.

a. Conditions. The **OR** function is represented algebraically by the + sign. This sign is read as or, not the familiar plus. To represent the **OR** function performed by the switches in figure 2-2, the equation A + B = C can be used. It would be read as A or B equals C. Let's look a little closer at what this equation is actually representing:

If switch A closes, the light will light.
If switch B closes, the light will light.
If both switch A and switch B close, the light will light.
If switch A and switch B are open, the light will not light.

What is the algebraic symbol for the OR function?

If you indicated this symbol +, you're right.
But how is it read?

Yes, this symbol is read--or.

b. Truth table. In digital logic, we let the ones and zeros do the talking. So if you were to go back to the equation (A + B = C) and give values to the variables, you would find only two possibilities for each variable.

Remember: Condition, state, and level have the same meaning. True, on, active, and high are associated with a one, while false, off, inactive, and low are associated with a zero.

(1) If a closed switch were the true condition, the open switch the false condition, and the light being lit the desired true condition--you could compose four equations:

1 + 0 = 1	switch A closed, B open	(true, false)
0 + 1 = 1	switch A open, B closed	(false, true)
1 + 1 = 1	switch A closed, B closed	(true, true)
0 + 0 = 0	switch A open, B open	(false, false)

Remember: These equations are read: one or zero equals one; zero or one equals one; one or one equals one; zero or zero equals zero.

The third equation should look strange to you. This is not a misprint. The algebra you have been working with since starting this study unit is called Boolean algebra. Boolean algebra is used to describe the relationships of digital logic. In Boolean algebra the + symbol does not have the same meaning it has in the ordinary algebra, with which you are no doubt more familiar. Here, in Boolean terms the + symbol means a logical sum.

(2) From these four equations, you can derive a truth table. A truth table is simply a summary of the equation with values inserted for the variables for a gate or circuit in chart form. A summary of the previous equation in chart form would look like this:

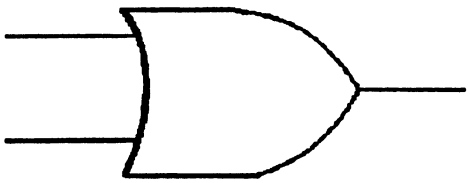
T = true F = false

A	B	C
T	F	T
F	T	T
T	T	T
F	F	F

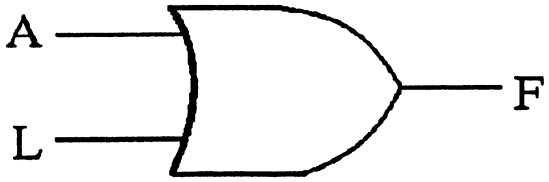
A	B	C
1	0	1
0	1	1
1	1	1
0	0	0

These charts are the truth tables for the **OR** function, with two inputs.

c. Gate. The last thing to learn is the gate itself. To the right is the schematic symbol for an **OR** gate.



Here is a labeled **OR** gate along with its equation and truth table:



A + L = F

A	L	F
1	0	1
0	1	1
1	1	1
0	0	0

Now that the gate, equation, and truth table are together, the relationship between the three are easily seen. A or L equals F tells you that F will only be true if A is true or if L is true. The truth table restates this relationship for each of the variables.

When is this gate active?	The gate is active when the output is true.
When is the output true?	The output is true when either input A or L is true.

2104. **AND** Function

The **AND** function acts like two switches in series (fig 2-3).

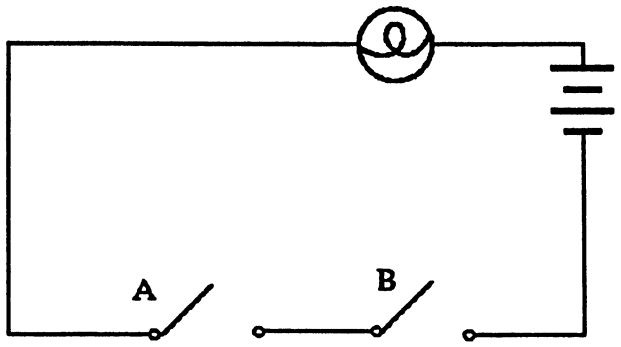


Fig 2-3. **AND** Function.

In this circuit the path for current flow is through both switch A and switch B. The **AND** function states that the output will be true only when all inputs are true (or in this case A and B must be true). As you go through this lesson, this concept will become clear to you.

a. Conditions. This function also has algebraic symbols. The symbols that indicate the **AND** function include the x symbol, the · symbol, and brackets ( ). This symbol is read as "and," not multiplied by or times. The **AND** function is also indicated by an absence of symbols as in this expression: AB = C. To algebraically represent the function performed by the circuit above, you could use this equation: A x B = C. This is read A and B equals C. Again, a closer look at what this equation represents in this circuit will be helpful:

If switch A alone closes, the light will not light.
If switch B alone closes, the light will not light.
If both switch A and switch B close, the light will light.
If both switch A and switch B open, the light will not light.

b. Truth table. You can derive four equations and a truth table from the previous statements describing the circuit in figure 2-3. The four equations would be as follows:

1 x 0 = 0	switch A closed, B open	(true, false)
0 x 1 = 0	switch A open, B closed	(false, true)
1 x 1 = 1	switch A closed, B closed	(true, true)
0 x 0 = 0	switch A open, B open	(false, false)

The truth tables would be as follows:

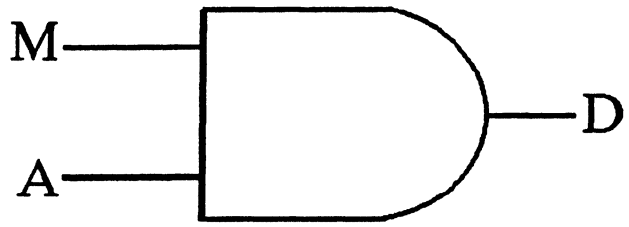
T = true F = false

A	B	C
T	F	F
F	T	F
T	T	T
F	F	F

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

These truth tables describe a two input **AND** function.

c. Gate. The best is left for last again. This is a labeled **AND** gate, its equation, and truth table.



M x A = D

M	A	D
1	0	0
0	1	0
1	1	1
0	0	0

2105. Complementation

Complementation is the inversion of a signal. Unlike the other two basic logic operations, complementation (also known as inversion or negation) is not performed by a gate. To complement is to reverse. The complement of a one is a zero. The complement of a zero is a one. So it follows that the complement of a true is a false and the complement of a false is a true. The complement of a high signal is a low signal. The complement of 100101_2 is 011010_2. As you can see, each one has become a zero and each zero has become a one.

a. Superior bar. Complementation is indicated by a superior bar. A superior bar is a line placed above the variable or number to be complemented. The complement of the number 100101_2 is $\overline{100101}_2$. Read this as not 100101_2 or as you've seen before, simply 011010_2. The complement of A is $\overline{A}$, which is read not A. If A were one, then $\overline{A}$ would be zero. The line above the A is a superior bar, an indicator of negation. The complement of the expression ABC is $\overline{ABC}$ which is read as not A and B and C. Note that this expression $\overline{ABC}$, isn't read as not A and not B and not C, which isn't the same as not A and B and C.

The proper way to read *this* expression, $\overline{A}\overline{B}\overline{C}$, however, is not A and not B and not C. As you learn about combinatorial gates and negated gates, the importance of properly reading these expressions will become clearer.

How is this expression, $\overline{A}+\overline{B}+\overline{C}$ read?
How is this expression, $\overline{A+B+C}$ read?

$\overline{A}+\overline{B}+\overline{C}$ is read as not A or not B or not C.
$\overline{A+B+C}$ is read as not A or B or C.

b. Double negate. To double negate is to complement twice. If $\overline{A}$ is complemented, the result would be $\overline{\overline{A}}$ (read as not not A). But logically (since there are only two states in digital logic), if it isn't not A it must be simply A. So $\overline{\overline{A}}$ is equal to A. As an example, if A = 1, then $\overline{A}$ = 0 (the complement of 1 is 0), and $\overline{\overline{A}}$ = 1 (the complement of 0 is 1).

c. Bubble/Inverter. This function (complementation), though not performed by a gate, will always be used in conjunction with a gate and often in conjunction with amplification (to restore signal strength). Schematically, the inverter is shown as a small circle or "bubble." Figure 2-4 illustrates inversion on the outputs of an amplifier and an **AND** gate. This "bubble" is known as a state indicator.

Note: The terms inverter and bubble are often used interchangeably for the schematic indication of inversion. You will see both terms throughout this course.

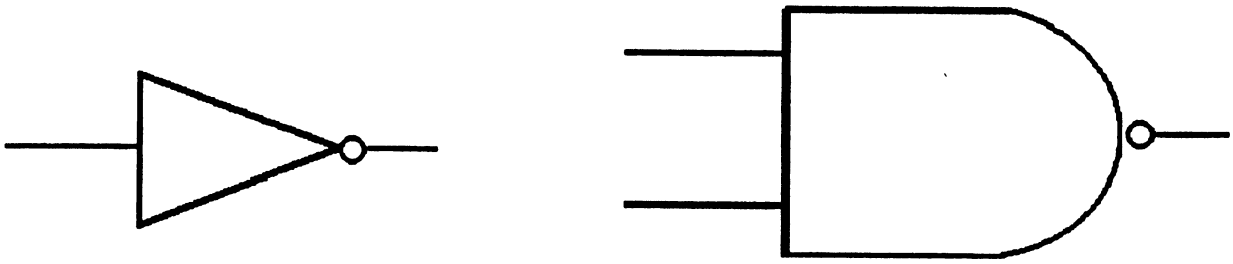


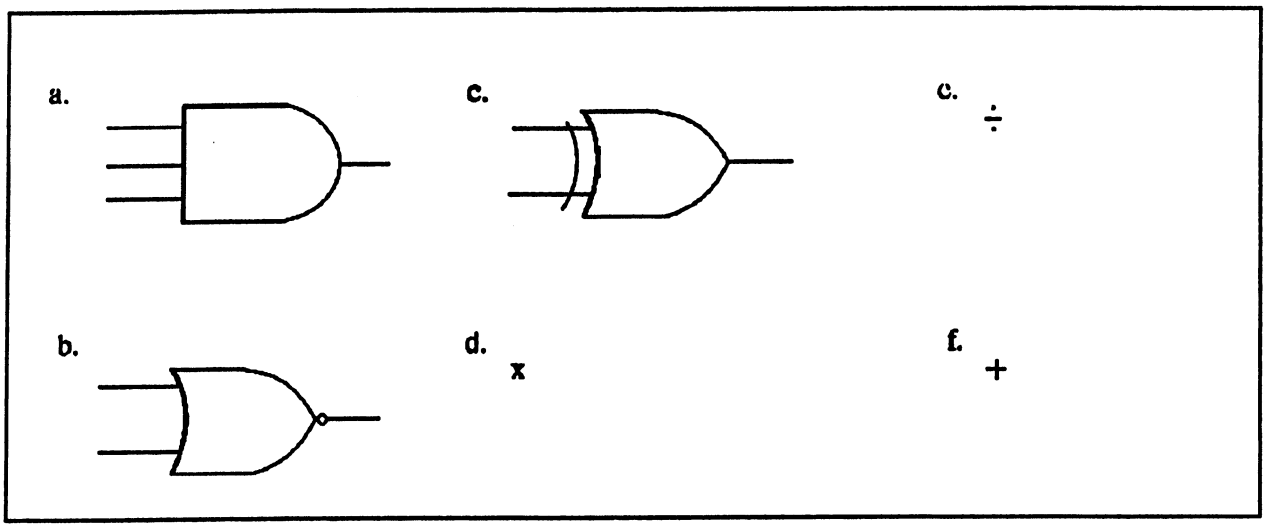
Fig 2-4. The "Bubble" as an Indicator of Complementation.

Lesson Summary. **AND**ing, **OR**ing, and complementation formed the foundation of this lesson, as well as the foundations of digital logic. The functions and gates you will learn about in the next lesson will all be a combination of these three functions.

Exercise: Complete items 1 through 12 by performing the action required. Check your responses against those listed at the end of this study unit.

1. The three basic logic operations are _____, _____, and _____.

Items 2 through 5 refer to the chart that follows. Review the chart, then answer the items.



2. Which algebraic symbol is associated with the **OR** function?

_____

3. Which algebraic symbol is associated with the **AND** function?

_____

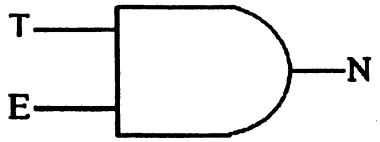
4. Which schematic symbol is associated with the **AND** function?

_____

5. Which schematic symbol is associated with the **OR** function?

_____

Items 6 and 7 refer to the following gate. Review the logic gate, then answer the items.



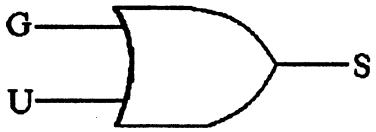
6. Complete the truth table for this gate.

T	E	N
0	0	
0	1	
1	0	
1	1	

7. What is the equation for this gate?

_____

Items 8 and 9 refer to the following gate. Review the gate, then answer the items.



8. Complete the truth table for this gate.

G	U	S
0	0	
0	1	
1	0	
1	1	

9. What is the equation for this gate?

_____

10. On, high, active, and one are associated with a

    _____.

11. Off, low, inactive, and zero are associated with a

    _____.

12. The logic state of a gate is determined by its _____ condition.

Lesson 2. COMBINATORIAL GATES

LEARNING OBJECTIVES

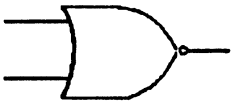
1. From a chart of schematic logic symbols, select a **NOR** symbol.
2. Given a labeled **NOR** gate, state in writing the equation describing that gate.
3. Given a labeled **NOR** gate, state in writing the truth table derived from that gate.
4. From a chart of schematic logic symbols, select a **NAND** symbol.
5. Given a labeled **NAND** gate, state in writing the equation describing that gate.
6. Given a labeled **NAND** gate, state in writing the truth table derived from that gate.
7. From a chart of schematic logic symbols, select an exclusive **OR (XOR)** symbol.
8. Given a labeled **XOR** gate, state in writing the equation describing that gate.
9. Given a labeled **XOR** gate, state in writing the truth table describing that gate.
10. From a chart of schematic logic symbols, select an exclusive **NOR (XNOR)** symbol.
11. Given a labeled **XNOR** gate, state in writing the truth table describing that gate.

2201. Combinatorial Functions

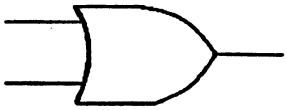
Combinatorial functions are so called because they combine the basic logic functions to create a new function. The three basic logic functions--**AND**ing, **OR**ing and complementation form the basis for the remaining functions you will learn. The two basic gates combined with complementation will be the focus of this lesson.

2202. **NOR** Function

The **NOR** function is a combination of the **OR** function with complementation. The **NOR** gate is an **OR** gate with a inverter on the output.



Take a look at the schematic symbol of a **NOR** gate on the left and an **OR** gate on the right to see the resemblance of a **NOR** to an **OR**.



a. Equivalent circuit. Look at the equivalent circuit (fig 2-5). You can see that only when both switches A and B are open (false), will the light illuminate (true).

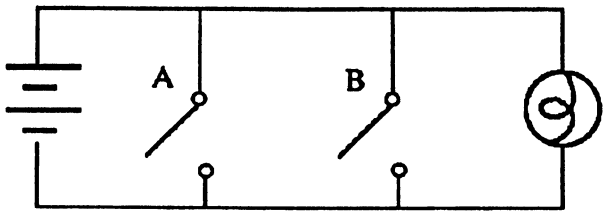


Fig 2-5. **NOR** Function.

Let's look closer at this circuit to see when the light will illuminate.

If switch A closes, the light will not light.
If switch B closes, the light will not light.
If both switch A and switch B close, the light will not light.
If switch A and switch B are open, the light will light.

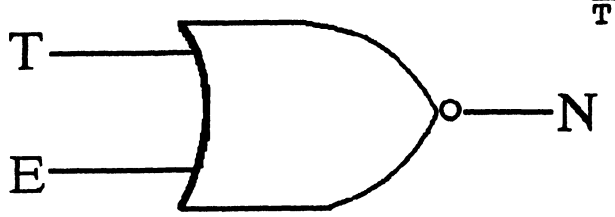
b. Conditions. To understand the **NOR** gate, you must first understand the **OR** gate. A high on any input of the **OR** gate will give you a high output. Similarly the **NOR** gate output would be the same as an **OR** gate output except the **NOR** gate output is inverted. Looking at the values in the truth table this should be a little clearer to you.

T = true F = false

A	B	C
T	F	F
F	T	F
T	T	F
F	F	T

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

The output values in these truth tables are the inverse of the **OR** gate's values from the previous lesson. The following is a labeled **NOR** gate along with the equation and the gate's truth table.



$\overline{T + E} = N$

T	E	N
0	0	1
0	1	0
1	0	0
1	1	0

This equation uses complemented variables, a concept that was introduced in the last lesson.

If you are confused with complementation, try this review:

* The complement of A is $\overline{A}$ (read as not A).
* The complement of $\overline{A}$ is simply A.
* The complement of 0 is 1.
* This expression $\overline{A+B+C}$ is read as not A or B or C.
* This expression $\overline{A}+\overline{B}+\overline{C}$ is read as not A or not B or not C.
* $\overline{\overline{A}}$ (read as not not A) is the same as A.

2203. **NAND** Function

The **NAND** function is a combination of the **AND** function and complementation. The **NAND** gate is an **AND** gate with an inverter on its output. The similarity of a **NAND** gate to an **AND** gate is seen in the schematic symbol for both of them (fig 2-6).

Fig 2-6. **NAND** Gate/**AND** Gate Schematic Symbols.

a. Equivalent circuit. If you look at this **NAND** gate equivalent circuit, you can see that if either switch is open the lamp will light.

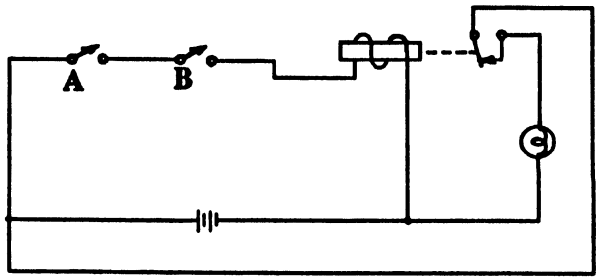


Fig 2-7. **NAND** Function.

b. Conditions. The truth table summarizes the possible conditions of this circuit.

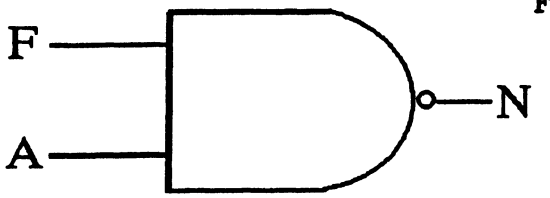
T = true F = false

A	B	C
T	F	T
F	T	T
T	T	F
F	F	T

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

Again, the output values found here are the inverse of the **AND** gate's values from the previous lesson.

The following illustrates the truth table, a labeled gate, and the equation for a two input **NAND** gate.

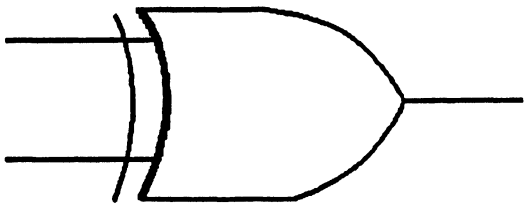


$$\overline{F \times A} = N$$

F	A	N
1	0	1
0	1	1
1	1	0
0	0	1

2204. Exclusive OR (XOR) Function

The exclusive or (**XOR**) function is a very useful logic function. The following is the schematic symbol for an **XOR** gate:



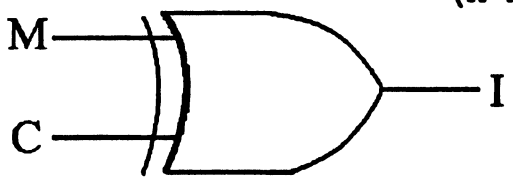
a. Conditions. The **XOR** gate outputs a high when its inputs are different. The output is low when the inputs are the same. Restated, the gate is inactive when the inputs are identical and active when the inputs are different.

> Remember: The condition or state of a logic gate is determined by its output. That is, if the output of a certain gate is a high, then the state or condition of that gate is active. If the output o a gate is a low, then that gate is inactive.

b. Equation. Let's look at a labeled **XOR** gate along with its truth table and equations.

$$M \oplus C = I$$

$$(\overline{M} \times C) + (M \times \overline{C}) = I$$



M	C	I
1	0	1
0	1	1
1	1	0
0	0	0

Notice that there are two equations to describe this gate--they both mean the same thing. The first equation introduces a new symbol, $\oplus$, which is the logical symbol for the **XOR** function. The above equation, $M \oplus C = I$, is read as M exclusive or C equals I. The second equation is in terms that you have already learned and needs no explanation.

2205. Exclusive **NOR** (**XNOR**) Function

Guess what? The **XNOR** function is similar to the **XOR** function. Guess what else? The **XNOR** function is a combination of complementation and the **XOR** function. In figure 2-7 you'll find a labeled **XNOR** gate.

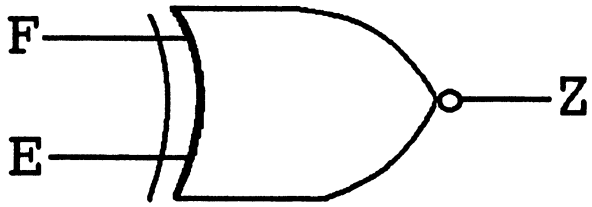
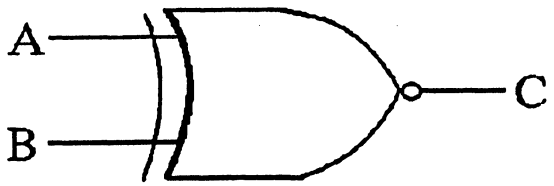


Fig 2-8. **XNOR** Gate Schematic Symbol.

a. Conditions. The **XNOR** gate has a high output when its inputs are the same. It is inactive when its inputs are different. The **XNOR** is similar to the **XOR** with the exception of an inverter on the output. This is becoming too easy, so to make it a little harder...in the box at the end of this sentence, write in the equation for the gate in the figure above.

The schematic logic symbol and truth tables for this gate (**XNOR**) follow:



T = true F = false

A	B	C
T	F	F
F	T	F
T	T	T
F	F	T

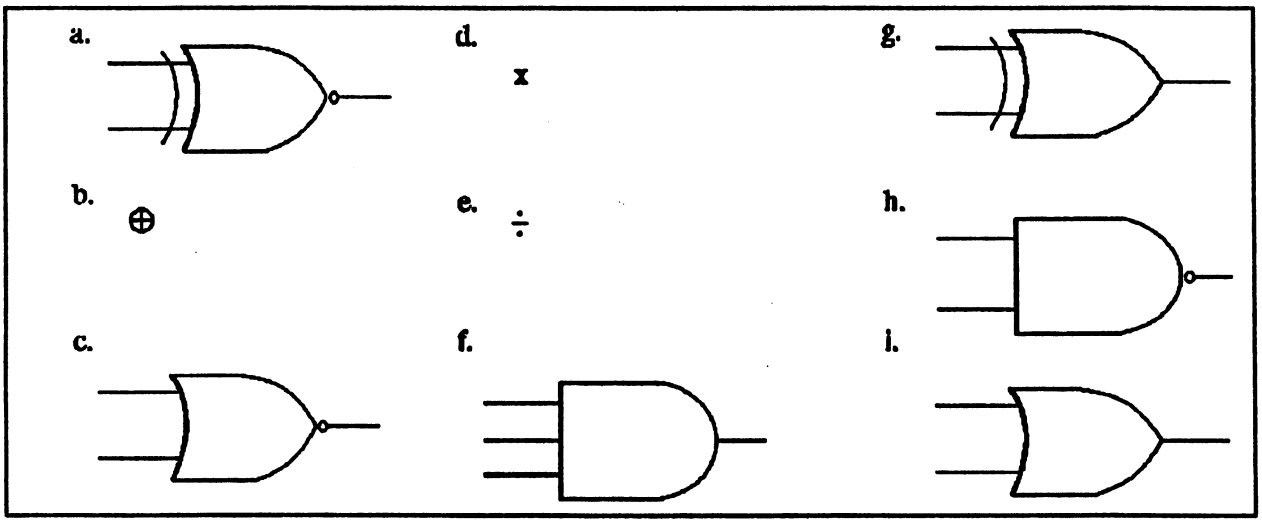
A	B	C
1	0	0
0	1	0
1	1	1
0	0	1

b. Equation. The equation (which you should know by now) for this gate is $(\overline{A} \times \overline{B}) + (A \times B) = C$. Why this equation? Well, for one thing it summarizes the truth table. But more importantly, it states what the **XNOR** gate does. If the inputs are the same, all high or all low--you get a high (one, true - active) out. If the inputs are different--you get a low (zero, false - inactive) out. But what about the equation for the gate in figure 2-8? That equation should be ... $(\overline{F} \times \overline{E}) + (F \times E) =$?

Lesson Summary. This lesson introduced you to logic functions that were combinations of the three basic functions. There's nothing too hard about any of these so far. In the next lesson, you'll see the bubble on the input of gates you already know.

Exercise: Complete items 1 through 13 by performing the action required. Check your responses against those listed at the end of this study unit.

Items 1 through 5 refer to the chart that follows. Review the chart, then answer the items.



1. Which schematic symbol is associated with the **NAND** function?

   _____

2. Which schematic symbol is associated with the **NOR** function?

   _____

3. Which schematic symbol is associated with the **XNOR** function?

   _____

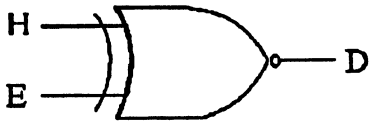
4. Which schematic symbol is associated with the **XOR** function?

   _____

5. Which algebraic symbol is associated with the **XOR** function?

   _____

Items 6 and 7 refer to the following gate. Review the logic gate, then answer the items.



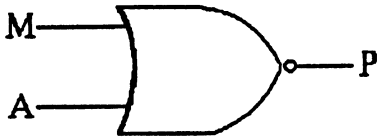
6. Complete the truth table for this gate.

H	E	D
0	0	
0	1	
1	0	
1	1	

7. What is the equation for this gate?

_____

Items 8 and 9 refer to the following gate. Review the gate, then answer the items.



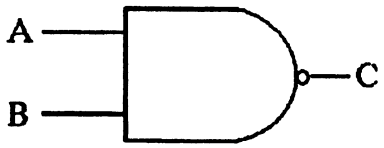
8. Complete the truth table for this gate.

M	A	P
0	0	
0	1	
1	0	
1	1	

9. What is the equation for this gate?

_____

Items 10 and 11 refer to the following gate. Review the logic gate, then answer the items.

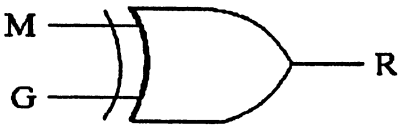


10. Complete the truth table for this gate.

A	B	C
0	0	
0	1	
1	0	
1	1	

11. What is the equation for this gate?

Items 12 and 13 refer to the following gate. Review the gate, then answer the items.



12. Complete the truth table for this gate.

M	G	R
0	0	
0	1	
1	0	
1	1	

13. What is the equation for this gate?

Lesson 3. INHIBITED GATES

LEARNING OBJECTIVES

1. Given a labeled inhibited logic gate, state in writing the equation derived from that gate.

2. Given a labeled inhibited gate, state in writing the truth table derived from that gate.

3. Given any logic gate, select an equivalent gate from a list of other logic gates.

2301. Inhibited Gates (Overview)

You've looked at two-input gates almost exclusively, but as you've learned, logic gates are not limited to two inputs. Additionaly, all gates seen so far have not had a state indicator (bubble) on the inputs. In this lesson you'll see gates with more than two inputs. You will also see gates with state indicators on the inputs; these are known as inhibited gates. The presence of the state indicator on the input of a gate indicates that a low activates that input.

a. Input. The logic gates of this lesson differ from the previous lesson's gates only in the respect of their inhibited inputs. None of the concepts introduced here are new ones. In this lesson, you will again combine the three basic logic functions.

b. Signal level. As you know, the state indicator is the symbol for complementation. The bubble on the input of a gate complements the input level prior to the signal entering the gate. One way to conceptualize this is shown in figure 2-9.

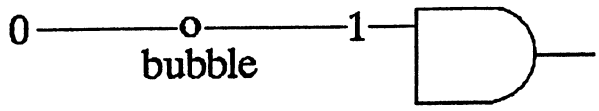


Fig 2-9. **AND** gate wth one inhibited input.

This figure shows an **AND** gate with one inhibited input. Although the signal on the inhibited input is a zero, the signal actually reaching the gate is a one because the original signal is complemented. With a one on the other input of the gate, the output is a one. So as you can see, the gate is not performing magic, just its normal function.

2302. Inhibited **AND**

Let's jump right in with a gate (fig 2-10), look at the equation and the truth table, and then see if we understand what's going on.

Remember: The presence or absence of the level indicator lets you know what the active condition inputs are.

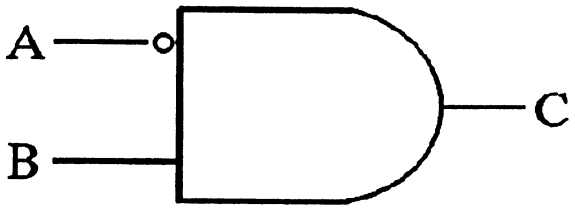


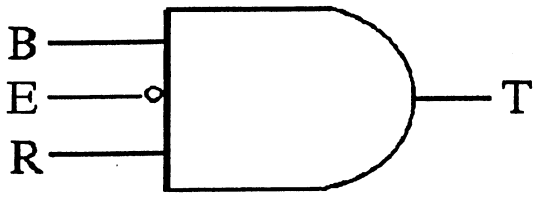
Fig 2-10. Inhibited **AND**.

a. Equation. As you've seen in previous lessons, the gate's equation is simply a statement of what makes that particular gate active. Now that a low will be the active condition on the inhibited input, the inhibited gate's equation will reflect this by including a complemented variable. The input conditions that make this gate active are zero (A) and one (B). The equation describing the active condition is $\bar{A}$ (not A) and B equals C ($\bar{A} \times B = C$).

b. Truth table. The truth table for this gate is shown below.

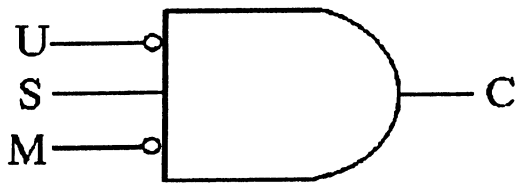
A	B	C
0	0	0
0	1	1
1	0	0
1	1	0

Two more exammples follow:



$B\bar{E}R = T$

B	E	R	T
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0



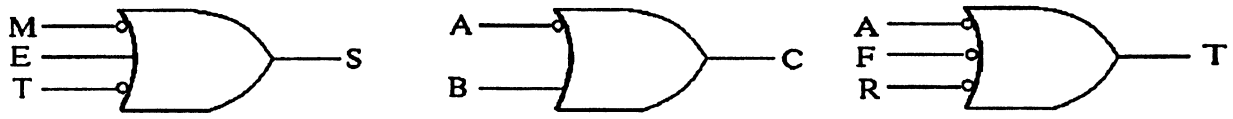
$\bar{U}S\bar{M} = C$

U	S	M	C
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

c. Conditions. Though these are three input gates, the **AND** function is still performed; that, is only when all inputs satisfy the active condition will the gate be active.

2303. Inhibited OR

These gates aren't very difficult now that you know the basics. Let's take a look at a few.



The first gate's equation is: $\overline{M} + E + \overline{T} = S$

The second gate's truth table is as follows:

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

The truth table and equation for the third gate are as follows:

$\overline{A} + \overline{F} + \overline{R} = T$

A	F	R	T
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	0

What is the truth table for the first gate? Use the truth table below to complete your answer.

Gate # 1

M	E	T	S
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

What is the equation for the second gate? Use the box below for your answer.

Gate # 2

Your answers should look like this:

Gate # 2

$\overline{A} + B = C$

Gate # 1

M	E	T	S
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

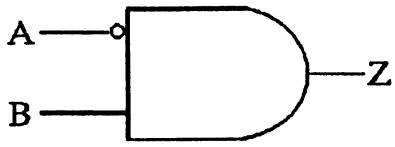
Why these answers? Let's look at gate # 1 first. We know this gate will be active (output a one) whenever any of its inputs are active. For a normal input, that would mean the presence of a one; for an inhibited input, that means the presence of a zero. Gate # 1 has two inhibited inputs M and T. This means that whenever M is low or E is high or T is low, the output S will be high. The truth table restates these conditions for gate # 1.

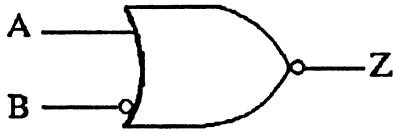
Now let's look at gate # 2. The equation for any gate states the conditions that make that gate active. The equation for gate # 2 reflects the fact that one input is inhibited (A) and that the presence of a zero on this input is the active input. Restated, the equation says if A is low or if B is high, then the output C will be high.

2304. Equivalent Gates

The term "equivalent gates" means two or more different gates that perform the same function (same truth table). You may be wondering, "Why bother learning something like this?" As a repairman, you will be confronted with these types of things. Gates come prepackaged on chips--they are not manufactured as individual gates. Circuit designers have to work within the constraints imposed by the chips available as they create equipment. If the circuit designers arrive at a point where the **NOR** function is called for but the **NOR** gate is unavailabe on the chip(s) they have to use, then they must use a different gate or combination of gates to meet the specifications of their design. The following are some equivalent gates you'll be likely to encounter.

a. Set one. The gates are different, the equations are different--but the outcome and the truth table are the same.

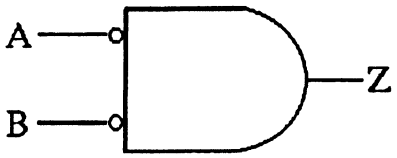


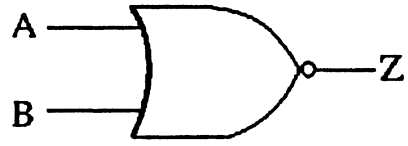


A	B	Z
0	0	0
0	1	1
1	0	0
1	1	0

This is what is meant by equivalent gates. In this first set of equivalent gates, you can see that there is but one set of variables that will satisfy the true condition.

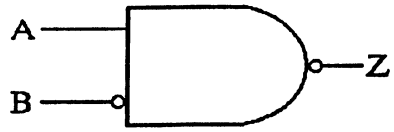
b. Set two. Again, these gates have only a single set of variables that make the gates true. Try all the variables to see if you truly understand these gates.

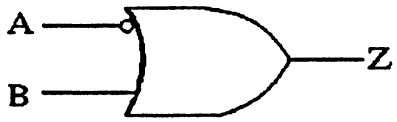




A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

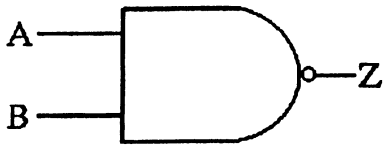
c. Set three. Nothing mind boggling going on with these two gates either. As you can see again, two different gates, same truth table.

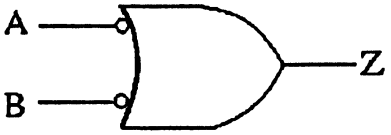




A	B	Z
0	0	1
0	1	1
1	0	0
1	1	1

d. Set four. You should now be confident enough in your own ability to fill in the truth table. So using the gates shown here, complete this truth table.





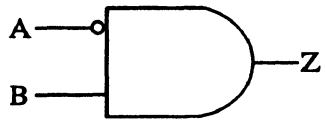
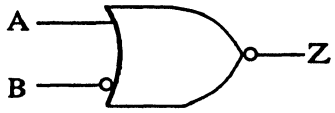
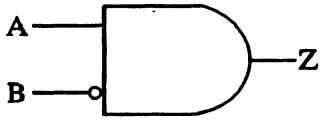
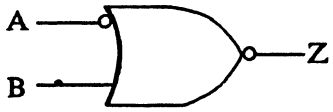
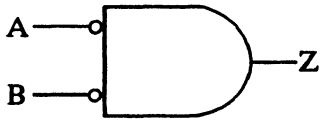
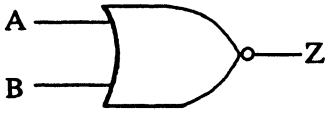
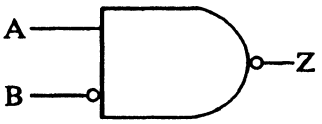
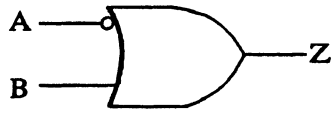
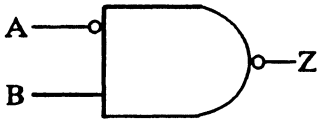
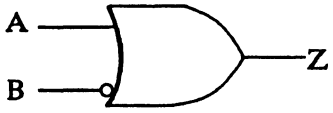
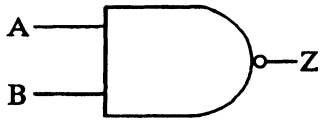
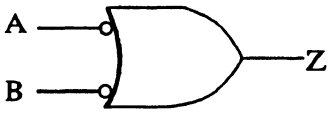
A	B	Z
0	0	
0	1	
1	0	
1	1	

Your completed truth table should look like this:

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

(This page intentionally left blank.)

COMMON GATES	A	B	Z
	0	0	0
	0	1	0
	1	0	0
$AB = Z$	1	1	1
	0	0	0
	0	1	1
	1	0	1
$A + B = Z$	1	1	1
	0	0	0
	0	1	1
	1	0	1
$(\overline{A} \times B) + (A \times \overline{B})$ $A \oplus B = Z$	1	1	0
	0	0	1
	0	1	1
	1	0	1
$\overline{AB} = Z$	1	1	0
	0	0	1
	0	1	0
	1	0	0
$\overline{A + B} = Z$	1	1	0
	0	0	1
	0	1	0
	1	0	0
$(\overline{A \times B}) + (A \times B)$	1	1	1

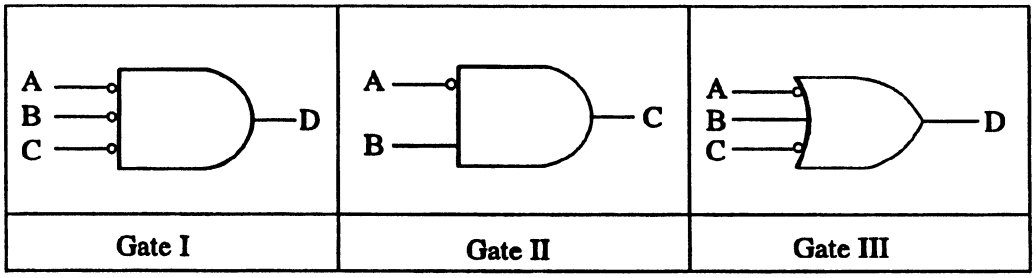
EQUIVALENT GATES		A	B	Z
		0	0	0
		0	1	1
		1	0	0
		1	1	0
		0	0	0
		0	1	0
		1	0	1
		1	1	0
		0	0	1
		0	1	0
		1	0	0
		1	1	0
		0	0	1
		0	1	1
		1	0	0
		1	1	1
		0	0	1
		0	1	0
		1	0	1
		1	1	1
		0	0	1
		0	1	1
		1	0	1
		1	1	0

Lesson Summary. The inhibited gates presented in this lesson are the pieces to the puzzle of digital logic. All the lessons that follow will help you put the pieces together. Understanding the concept of equivalent gates can prove to be a valuable tool to the electronic repairman. This lesson introduced you to this concept.

- -

Exercise: Complete items 1 through 6 by performing the action required. Check your responses against those listed at the end of this study unit.

Items 1 through 4 refer to the chart of logic gates below. Review the gates, then answer the items.



1. Which is the truth table for gate I?

a.

A	B	C	D
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	1

b.

A	B	C	D
0	0	0	0
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	1

c.

A	B	C	D
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	0

d.

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

2. What is the equation for gate I?

a. $\overline{A} \times \overline{B} \times \overline{C} = D$
b. $\overline{ABC} = D$
c. $\overline{A} + \overline{B} + \overline{C} = D$
d. $\overline{A + B + C} = D$

3. What is the equation for gate II?

a. $\overline{A} + B = \overline{C}$
b. $\overline{A} + B = \overline{C}$
c. $\overline{A} \times B = \overline{C}$
d. $\overline{A} \times B = \overline{C}$

4. Which is the truth table for gate III?

a.

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

b.

A	B	C	D
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	0

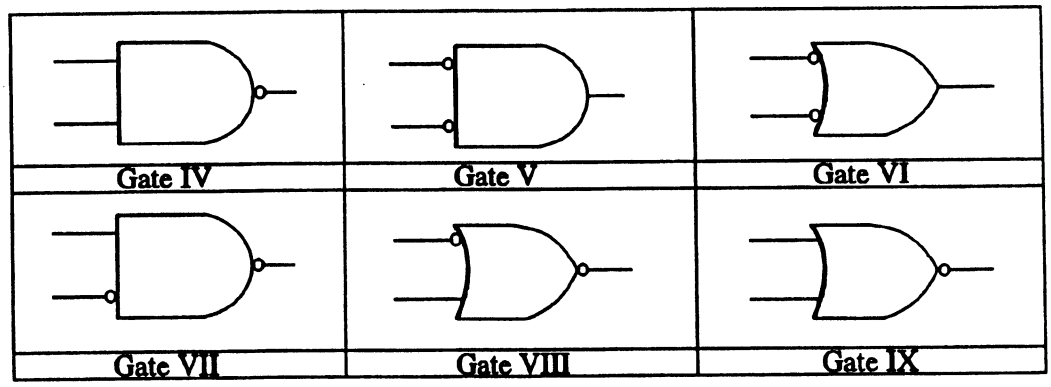
c.

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

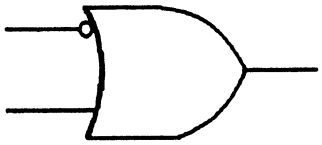
d.

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

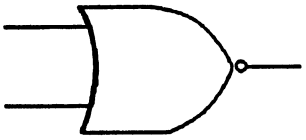
Items 5 and 6 refer to the chart of logic gates below. Review the gates, then answer the items.



5. Which gate(s) in the chart have the same truth table as the following gate?



6. Which gate(s) in the chart have the same truth table as the following gate?



UNIT SUMMARY

Gates, gates, and more gates--this is the stuff of digital logic. Now that you've mastered the building blocks, it's time to make something with these gates. The study units that follow will allow you to do just that as you explore some basic circuits.

Lesson 1 Exercise Solutions

		Reference
1.	**AND**ing, **OR**ing, and complementing (in any order)	2101
2.	f.	2104a
3.	d.	2105a
4.	a.	2105c
5.	b.	2104c
6.	T E N 0 0 0 0 1 0 1 0 0 1 1 1	2105b
7.	T x E = N	2105
8.	G U S 0 0 0 0 1 1 1 0 1 1 1 1	2104b
9.	G + U = S	2104
10.	Higher voltage (true)	2102
11.	Lower voltage (false)	2102
12.	Output	2102

Lesson 2 Exercise Solutions

		Reference
1.	h.	2203
2.	c.	2202
3.	a.	2205a
4.	g.	2204
5.	b.	2204b
6.	H E D 0 0 1 0 1 0 1 0 0 1 1 1	2205
7.	(H x E) + (H x E) = D	2205
8.	M A P 0 0 0 0 1 0 1 0 0 1 1 1	2202b
9.	M + A = P	2202b
10.	A B C 0 0 1 0 1 1 1 0 1 1 1 0	2203b
11.	A x B = C	2203b

12.

M	G	R
0	0	0
0	1	1
1	0	1
1	1	0

2204b

13. $M \oplus G = R$ or $(\overline{M} \times G) + (M \times \overline{G}) = R$ 2204b

Lesson 3 Exercise Solutions

		Reference
1.	d.	2302b
2.	a.	2302a
3.	c.	2302a
4.	d.	2303
5.	VII.	2303
6.	V.	2304

STUDY UNIT 3

COMMON MULTIGATE CIRCUITS

Introduction. It's time now to combine logic gates into functioning circuits. Your ability to understand the operation of the more common circuits will help you go a long way towards becoming an ace troubleshooter.

In this study unit, we'll begin with some definitions, then we'll focus on the most common type of digital logic circuits called bistable devices; these include the flip-flop and the latch. These simple circuits serve as the basis for many of the other circuits found in digital electronic equipment. Finally, after the flip-flops, we'll look at a few other circuits.

Lesson 1. Timing, Triggering, Clocking, and Bistable Devices

LEARNING OBJECTIVES

1. Given a logic gate, input waveforms, and a list of output waveforms, select the output waveform corresponding to the given input waveforms.

2. From a list of definitions, select the definition for triggering.

3. From a list of definitions, select the definition for clocking.

4. From a list of definitions, select a definition for a bistable device.

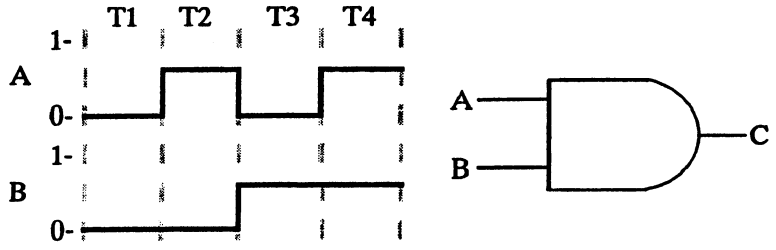
To operate properly, most digital circuits rely on sequenced events. This means that ***when*** a signal is input into a logic circuit it is often as important as the signal itself. Digital devices with multiple inputs require that those input signals are sufficiently strong and properly shaped; in addition, they must be of proper duration and in a specific sequence. Three terms are associated with this concept of sequenced events--timing, triggering, and clocking. This lesson will present the concept of sequenced events by exploring these three terms.

3101. Definitions

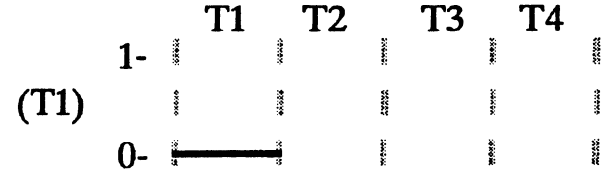
a. Timing. As previously emphasized, the sequence of events is very important in digital circuits. Synonymous with this sequencing is timing. To fully understand digital logic circuits, you must have a firm grasp of timing. Timing is no more than a way of representing a sequence of events. In the case of digital devices, timing represents what is happening in the device from moment to moment as a signal is applied. We will explore timing through the use of timing diagrams.

Note: Are you having difficulty remembering the various gates and their associated truth tables? Remove and use the perforated pages at the end of study unit 2, para. 2305, as a study aid. These pages of logic gates and truth tables will help as you tackle this study unit.

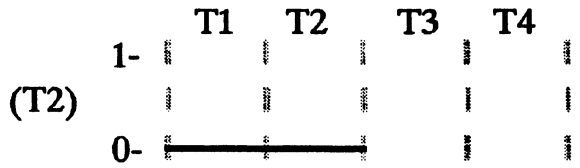
(1) Timing diagrams. A timing diagram is a graphic description of an event divided into equal time periods. To use a timing diagram you must consider four points--time period, input, device, and output. The input, the effect of the input on the device, and the output (if any) are analyzed for each time period. Let's develop a timing diagram for an **AND** gate to bring our four points together. Notice that the inputs are given and are divided into time periods T1, T2, T3, and T4. From this gate and these inputs, we will construct an output waveform by analyzing the device and inputs for each time period.



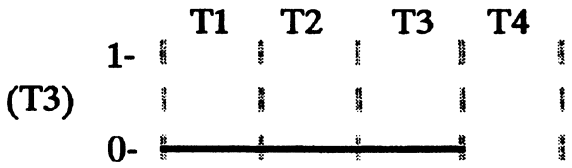
(a) T1. At T1, the input on A is zero as is the input on B. This would make the output at C zero as well. Let's start constructing our output waveform:



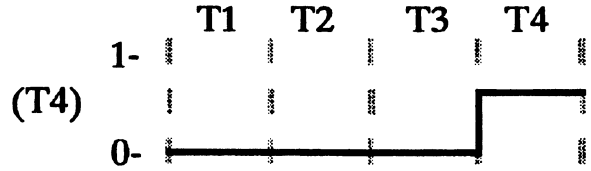
(b) T2. At T2, A is one and B is zero. Consequently, our output at C is zero. Continuing our output waveform:



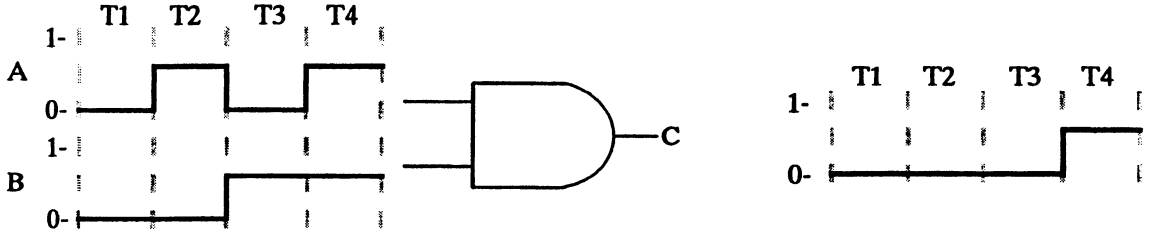
(c) T3. At T3, A is zero and B is one. This means that C is zero as well. Now, add to our output waveform:



(d) T4. At T4, A is one and B is one also. This makes C, our output, one. Now, finishing the waveform:

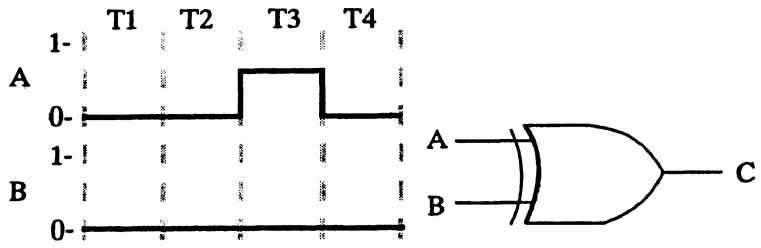


Let's take a graphical look at what we've done:

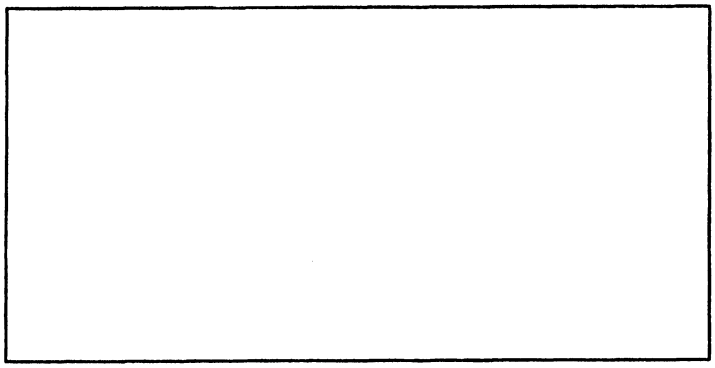


As you can tell, there's really nothing new about the values here. In fact, if you made a truth table with the data presented, you'd see that the truth table would describe the output conditions of a two input **AND** gate.

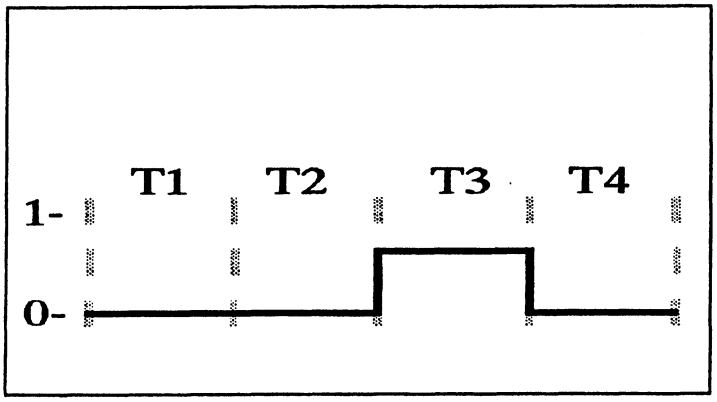
(2) Practice. Here's one more gate. I'll give you the input waveforms, but the output waveform I'll leave to you.



Here's a place for you to put your answer:



Your waveform should look like this:

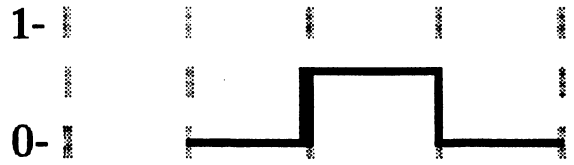


Here is a quick explanation of why your waveform should look like this. First, this is an **XOR** gate, which means the output will be one when the inputs are different (active state), and the output will be zero when the inputs are the same. At T1 the inputs are both lows, consequently the output will be zero. At T2 the inputs are again both zero, thus the output is a low again. T3 has inputs of a high on A and a low on B, which results in an output of one. Finally, at T4 we have lows on both inputs; again, this won't activate this gate so the output is a low. This is how the waveform above was derived.

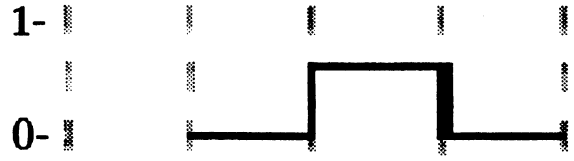
b. Triggering. The act of changing a digital electronic device from one condition to another is called triggering. In other words, triggering is a description of the signal used to change the state of a digital device. There are two basic types of triggering, edge and level. A device may be triggered by either the beginning or the end of a signal, a condition known as edge triggering. A level-triggered circuit, on the other hand, would react to the state of the signal--a one or a zero.

(1) Edge triggering. Edge triggering describes the condition where a circuit reacts immediately to the pulse. Edge triggering can be positive or negative.

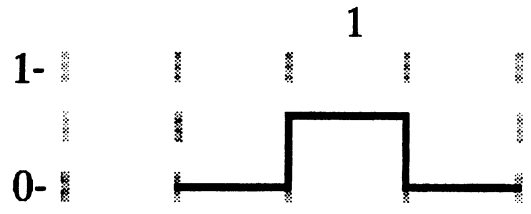
(a) Positive edge triggering simply means that the circuit reacts to the positive or leading edge of a pulse (a change from a zero to a one).

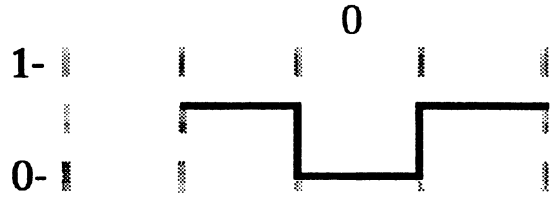


(b) Negative edge triggering means that the circuit reacts to the negative or lagging edge of the pulse (a change from a one to a zero).



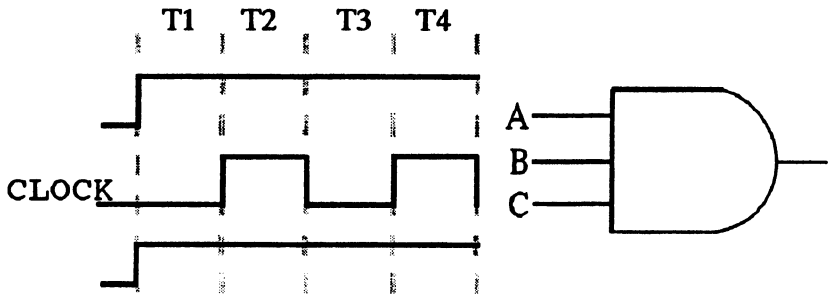
(2) Level triggering. Level triggering describes the condition where the state of the pulse (1 or 0) carries out the transfer of information or completes an action.





c. Clocking. Clocking determines when a circuit will react to that trigger or input. In other words, instead of reacting immediately to a change of input conditions, a circuit will wait for the clock pulse before accepting an input. As we will discover, in clocked logic circuits the clock pulse acts to control, initiate, or complete what is happening in that circuit. It does this by enabling or disabling the inputs at the proper time(s). In essence, clocking will either initiate or complete what is happening in a circuit. Just as with triggering, there are two possible types of clock pulses--edge and level.

An example of clocking can be seen in the timing diagram below:



The input on A and C are kept at a constant logic high. Therefore, the clock input at B will determine when and what will be felt at D. In this case, when the clock is high, the output will be high; when the clock is low, the output will be low.

d. Bistable devices. The most common digital logic circuits are bistable. Bistable devices have two stable states. The two stable states give these devices the ability to remember. That is, once these devices are set (given a brief input on their input lines) to a certain condition, the devices remain in (remember) that condition until they are reset. During each stable state the output of the device will remain constant. In the next lesson we will focus upon two bistable devices, the latch and the flip-flop. As you will see, understanding timing, triggering, and clocking is essential if you are to understand these basic circuits. These bistable devices are the heart of most digital circuits.

Lesson Summary. In this lesson we looked at timing, triggering, and clocking. In addition, the term bistable was introduced. Although lesson one offered only a brief inspection of these concepts, the lessons that follow will reinforce these ideas.

Exercise: Complete items 1 through 4 by performing the action required. Check your responses against those listed at the end of this study unit.

1. Triggering is best described by which one of the following?

   a. The act of changing a digital electronic device from one state to another
   b. A signal that determines when a device will change from one condition to another
   c. A description of waveform inputs and outputs to a digital electronic device
   d. That part of the output signal that corresponds to the input signal

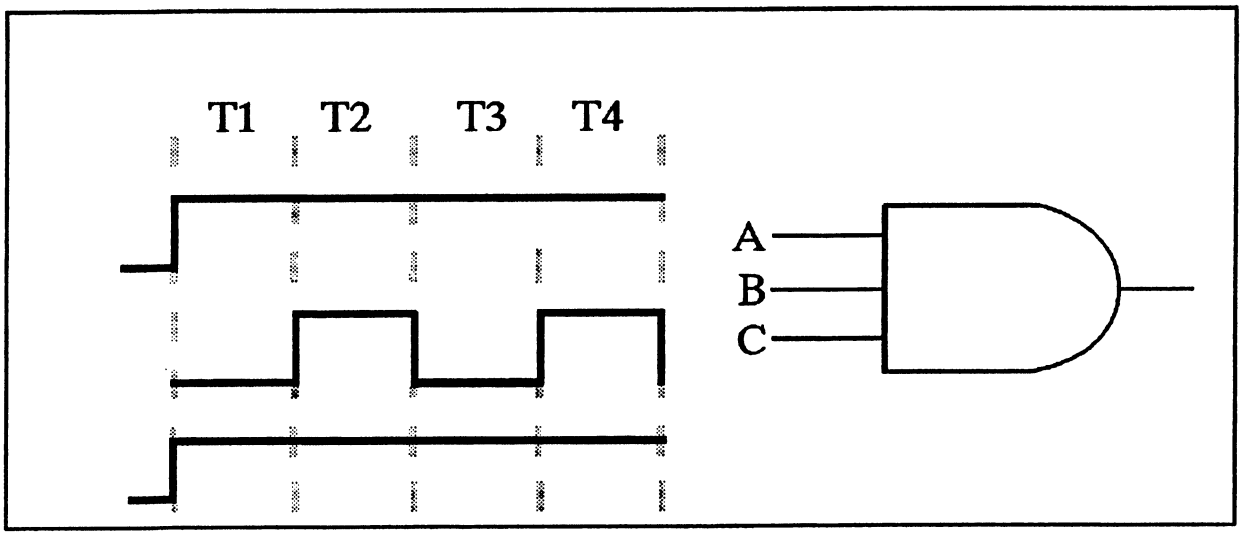
2. Clocking can best be described as

   a. a signal that determines when a device will change from one condition to another.
   b. the act of changing a digital device from one condition to another.
   c. a description of the cycle of operation for a digital device.
   d. a signal that monitors a digital device or circuit.

3. A bistable device can best be described as a digital electronic device

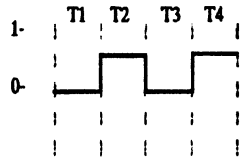
   a. with two or more stable states.
   b. with two stable states.
   c. that has two stable states, with the output remaining constant in one state until the device receives an input to change that state.
   d. that has two or more stable states, with the output remaining constant during each state.

For question 4, refer to the following diagram.

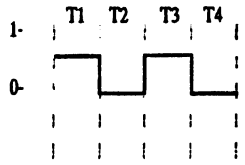


4. Select the proper output waveform, given this gate and these input waveforms.

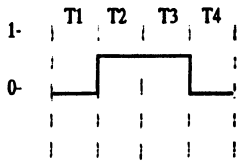
a.



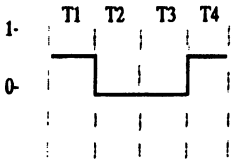
b.



c.



d.



Lesson 2. Latches and Flip-Flops

LEARNING OBJECTIVES

1. Given inputs to any RS latch, determine the outputs.
2. Given inputs to any RS flip-flop, determine the outputs.
3. From a list of definitions, select the definition of preset/clear.
4. From a list of definitions, select the definition of race.
5. Match logic symbols with the name of the bistable device it represents.
6. Given a logic symbol for a bistable device, identify the elements of the symbol.
7. Given any flip-flop or latch, select from a list the truth table derived from that flip-flop or latch.
8. Given input and clock conditions for any flip-flop, state the sequence of logic levels from input to output.

In this lesson, you'll examine latches and flip-flops. These easy-to-learn devices are at the core of all digital devices. Though numerous, they all share the following characteristics:

- They have the ability to store (remember) and pass on information (data).
- They have at least one input for data.
- They have two complimentary outputs which are designated as Q and $\overline{Q}$ (not Q).
- The device condition is determined by the state of Q.
- When the input conditions make Q equal to 1, the device is considered set.
- When the input conditions make Q equal to 0, the device is considered reset.
- They have the ability to be set (Q = 1) by pulsing the preset input, regardless of the state of the clock or data inputs.
- They have the ability to be reset (Q = 0) by pulsing the clear input, regardless of the state of the clock or data inputs.

The foundation for this lesson is a basic understanding of these characteristics. Each of these characteristics will be explained to you as we progress through this lesson. Though the list looks long, all the pieces of this puzzle will fall into place very easily as you continue this lesson. Let's begin our journey through storage devices with two simple latches.

3201. Set/Reset (RS) Latch

As usual, we'll start small and work our way into bigger and better things. A latch is a simple bistable device. The two latches we'll start with are very easy. A latch is also known as a rudimentary memory or bistable multivibrator. These latches will demonstrate the first six common characteristics.

> Although often called flip-flops, latches technically are not flip-flops. The difference is that a latch can be triggered at any time, while a flip-flop requires a clock pulse along with the trigger (data) input.

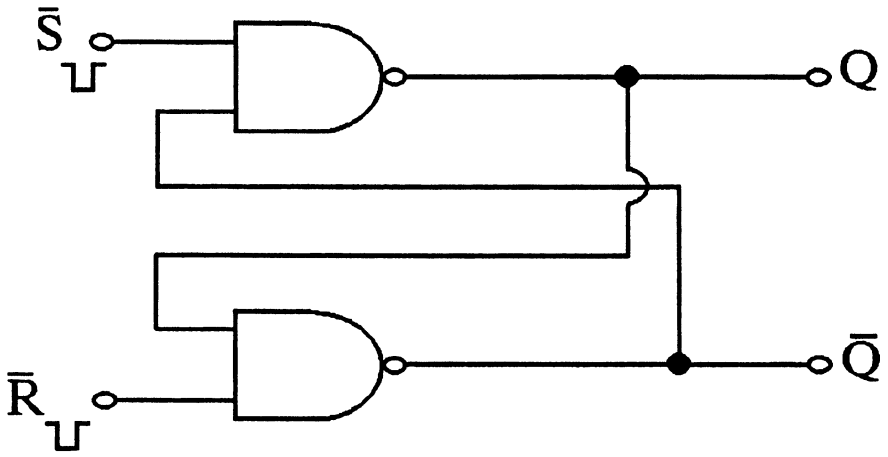
a. Input/Output. An RS latch has two data inputs and two outputs. The data inputs are labeled, Set (S) and Reset (R); the outputs are labeled, Q and $\overline{Q}$ (not Q). All latches (as well as flip-flops) should have outputs that complement each other (remember complements). If a set of data inputs attempts to produce equal outputs, that condition will be called "disallowed." The disallowed state will be explained later in this section.

> Remember: The complement of 1 is 0. The complement of A is $\overline{A}$ (not A). Flip-flops and latches have at least one input for data and have complimentary outputs.

b. Set/Reset. These simple memory devices can be Set (Q = 1) or Reset (Q = 0) by placing a brief input on the Set and Reset lines. The device then holds or stores that information once the input is taken away.

> Remember: All latches and flip-flops can store information. The state of Q determines the device condition.

c. Application. Let's look at two simple Set/Reset (RS) latches. The following are two simple latches, one composed of **NAND** gates (fig 3-1) and the other of **NOR** gates (fig 3-2).



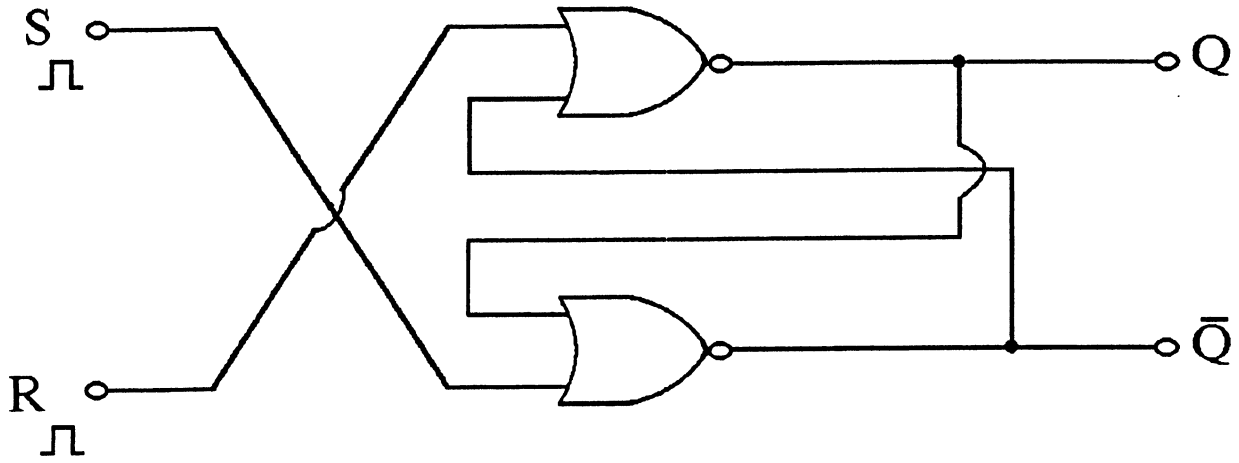
$\bar{S}$	$\bar{R}$	Q	$\bar{Q}$
1	1	NO CHANGE	
0	1	1	0
1	0	0	1
0	0	DISALLOWED STATE (BOTH 1'S)	

Fig 3-1. RS latch composed of **NAND** gates.

As you can see, you're still not finished with truth tables. As well, these simple circuits will be the foundation for more complex circuits which follow. Let's now explore flip-flops.

(1) Look at the truth table and circuit in figure 3-1. The latch in this figure responds to a low input on either its $\bar{S}$ or $\bar{R}$ inputs. Notice in the truth table that if both inputs are high there is no change in the output condition. Also notice the disallowed condition of two lows on the inputs.

(2) Looking at the circuit in figure 3-2, we see that the **NOR** circuit reacts to highs on the inputs. In this circuit, two lows produce no change in output. The disallowed condition for this circuit is the condition of having two highs as inputs.



S	R	Q	$\overline{Q}$
0	0	NO CHANGE	
1	0	1	0
0	1	0	1
1	1	DISALLOWED STATE (BOTH 0'S)	

Fig 3-2. RS latch composed of **NOR** gates.

d. Symbols. Figure 3-3 shows two logic symbols; the first is the **NAND** gate latch and the second is the **NOR** gate latch.

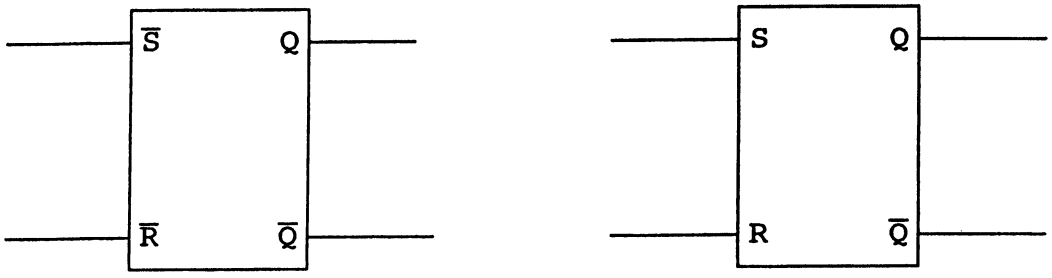


Fig 3-3. RS latch logic symbols.

e. Race. We also must consider what would happen when an attempt is made to change a latch to the disallowed condition. A latch's output, like a logic gate, is dependent upon the inputs. But unlike a gate, in a latch the sequence in which the inputs are applied is very important.

When a latch is changed to the disallowed condition, as when both data inputs are driven low (**NAND** gate latch), or when both data inputs are driven high (**NOR** gate latch), it is impossible to change both inputs at precisely the same instant. In essence, it becomes a "race" to see which input will control the output. This condition is known as race.

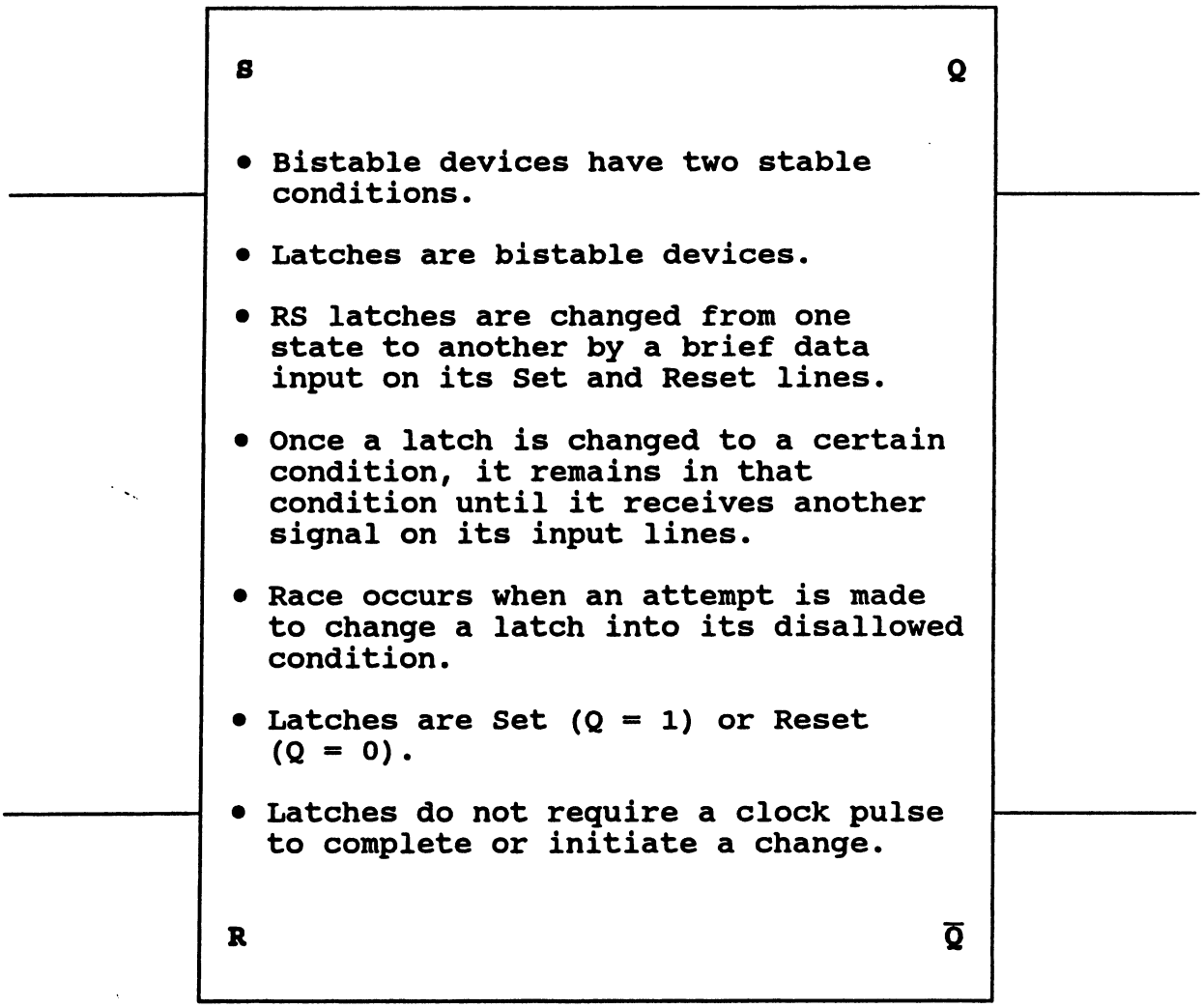
The outputs during these conditions are unpredictable because there is no way of predicting which input will change first.

Let's analyze our two previous latches.

(1) In the latch of figure 3-1 (**NAND** gates), if $\overline{S}$ goes low first, then the latch will output a high on Q and a low on $\overline{Q}$. But if $\overline{R}$ goes low before $\overline{S}$, then there will be a low on Q and a high on $\overline{Q}$.

(2) In the latch of figure 3-2 (**NOR** gates), if S goes high before R, then the latch will have a high on Q and a low on $\overline{Q}$. If the opposite happens and R goes high first, the latch will output a low on Q and a high on $\overline{Q}$.

Let's review:



3202. RS Flip-flop

What differentiates a latch from a flip-flop?

You are right, if your answer indicated these differences:

* Latches are unclocked; that is, they can be triggered at any time.

* Flip-flops require a clock pulse along with a trigger to change states.

Now, let's add a clocking (Do you remember clocking?) circuit to an RS latch to get an RS flip-flop.

a. Analysis. Figure 3-4 shows a typical RS flip-flop. Notice that this is the same circuit as figure 3-2, the only difference is that a clocking circuit has been added.

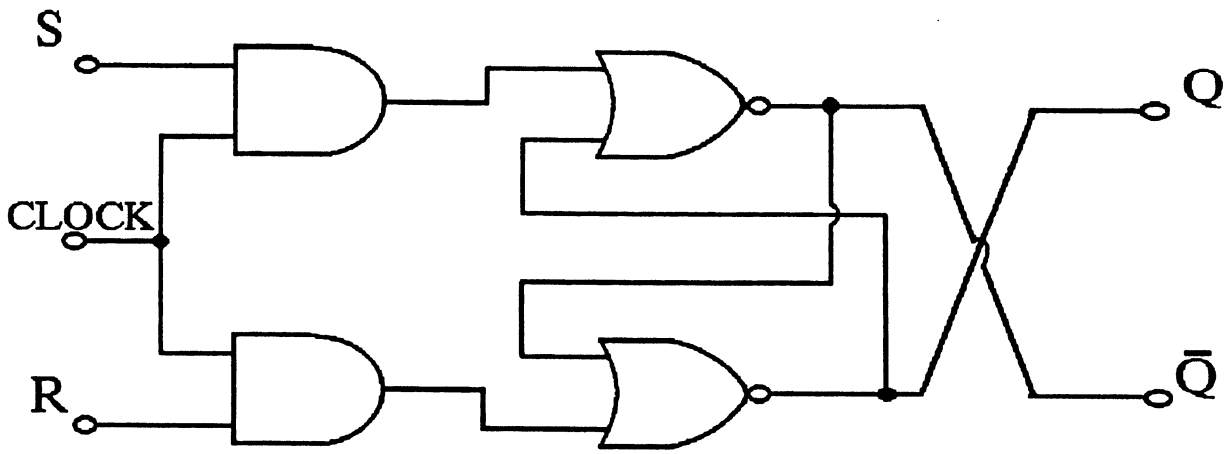


Fig 3-4. RS Flip-flop.

b. Review. Looking back on the latch from fig 3-2, we saw that the circuit responded to a high on either the S or R inputs. A high on the S input would drive Q high; a high on the R input would drive $\bar{Q}$ high. These conditions still apply, but now we also need a clock pulse to change our device.

Let's look at this (figure 3-4) closely. A high on the clock, along with a high on the Set input, drives Q high; whereas a high on the clock, along with a high on the Reset, drives $\bar{Q}$ high.

c. Clock. The clock in this circuit does two things for us. First, it enables or disables both **AND** gates simultaneously. Secondly, it provides a "window" to change (input data) the flip-flop, and in this particular instance that "window" is "open" when there is a high on the clock line. Restated, unless there is a high on the clock line, this flip-flop cannot change states.

d. Preset/Clear. What happens when a digital circut is first turned on? What is the initial condition of the various flip-flops and latches? Not sure, are you? Well, I don't know either. There is no way to be sure what those numerous little memories are storing. But that can be fixed. This can be done by adding to our circuit preset and clear inputs. These inputs will allow you to Set or Reset the condition of the various devices without an input on the clock line or the data input line(s).

Preset and clear inputs may be added to any latch or flip-flop. What does this mean? This means that when a circuit is turned on--if all its preset inputs were to be pulsed to the active condition--then all the circuit's latches and flip-flops would be **initialized** with Q outputs logic high. This way you don't have to start off in the dark.

What would happen in a circuit, if all the clear inputs were pulsed to the active level?

Glad to see you are paying attention! You are right, all Q outputs would be initialized logic low.

e. Circuit. Figure 3-5 shows the same flip-flop we've been working with, only now with preset and clear inputs added. A high on the preset input will make Q logic high. A high on the clear input will make Q logic low.

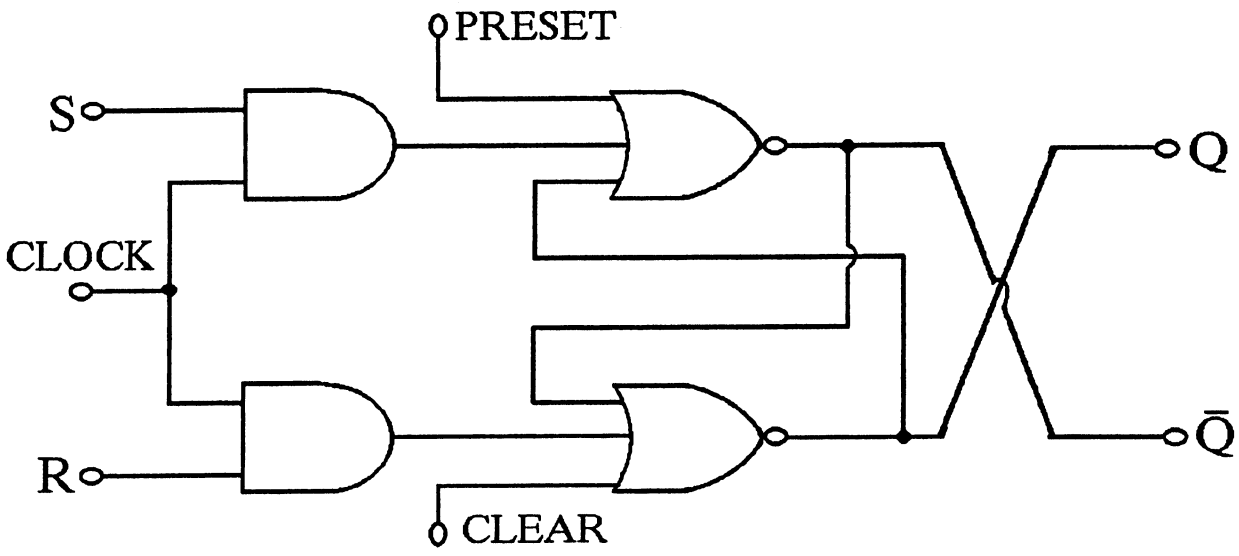


Fig 3-5. RS Flip-flop with clear and preset inputs.

3203. D Latch and D Flip-flop

The latches and flip-flops presented so far are all subject to race conditions. Now it's time to look at circuits whose race problems have been eliminated. In our first example, this is done quite easily with the help of an inverter (remember the bubble).

What is race?

Good!
An attempt to drive the outputs into the disallowed condition; that is, an attempt to drive Q and $\overline{Q}$ to the same level.

a. D Latch. The D stands for data or delay. As you can see in figure 3-6, it is not possible to drive this latch into the disallowed condition. The single split data input has killed the chance of both R and S being the same state. If there is a high on the data line, then S is high and R is low. Consequently, Q is high. A low on the data line drives Q low. In other words, the latch is Set with a high on the data line and Reset with a low on the data line.

Remember: Set is the same as saying Q = 1.
Reset is the same as saying Q = 0.

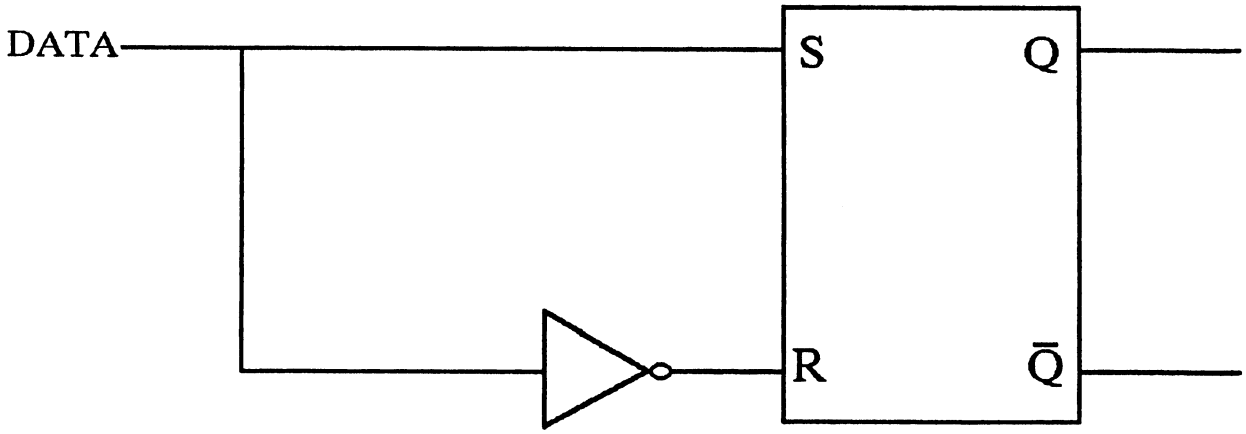
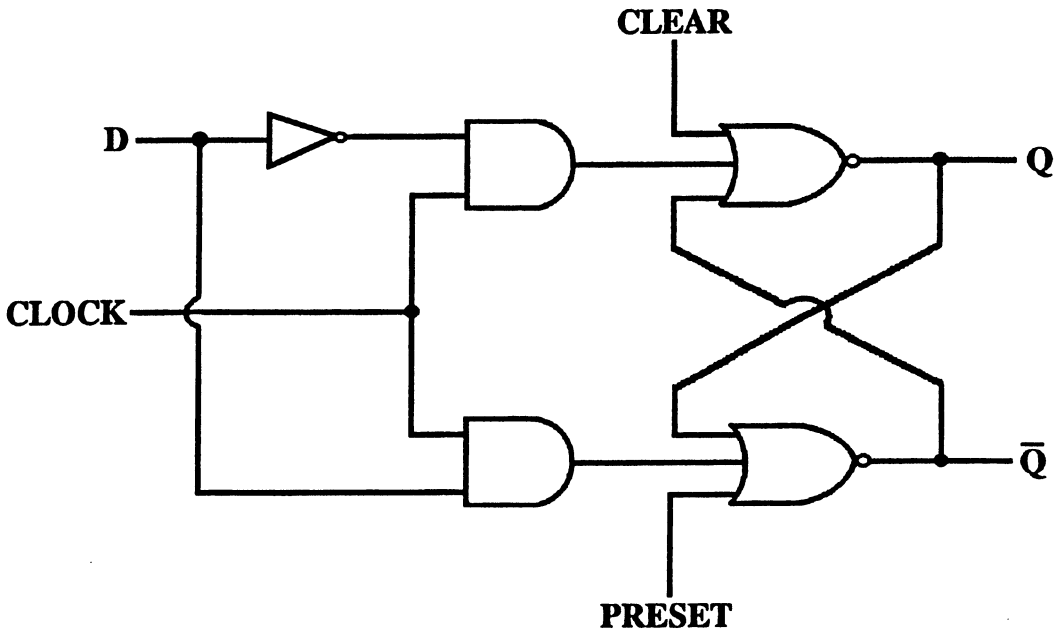


Fig 3-6. D Latch logic symbol.

b. D Flip-flop. The D flip-flop (figure 3-7) also makes use of a single split input to eliminate the problems of race. With the exception of the inverter, this circuit should look familiar. Notice that the truth table reflects that this flip-flop has no disallowed condition.



D	CLOCK	Q	$\overline{Q}$
0	0	NO CHANGE	
0	1	0	1
1	0	NO CHANGE	
1	1	1	0

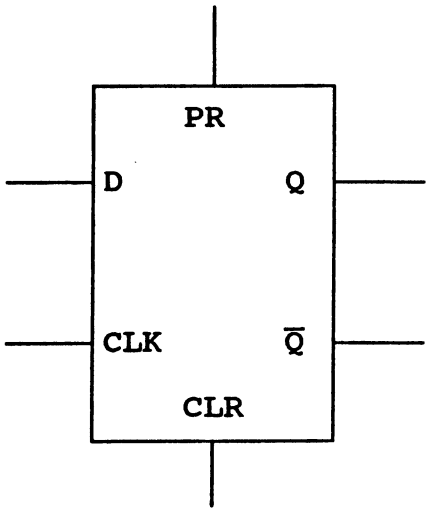
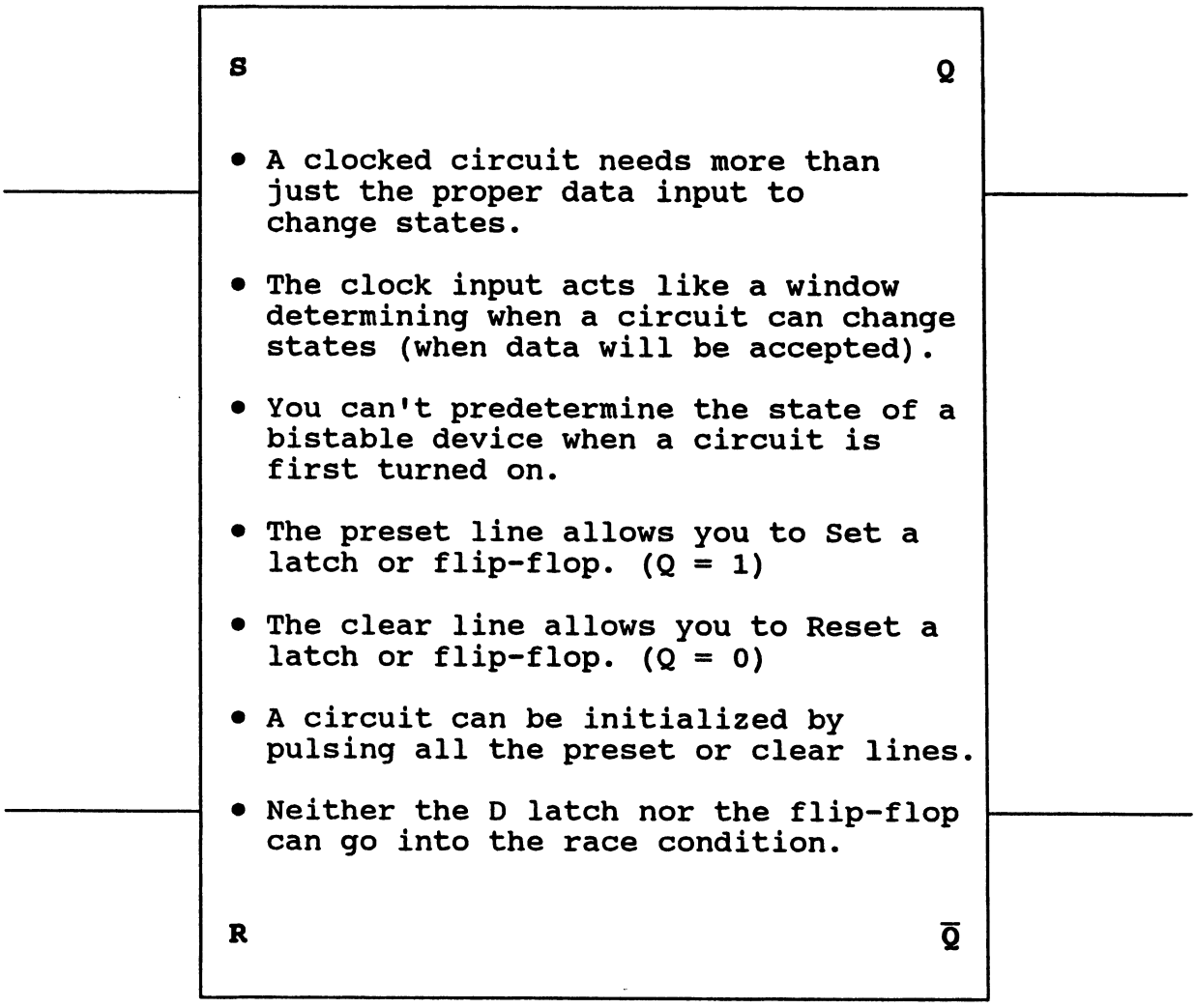


Fig 3-7. D Flip-flop, truth table, and logic symbol.

Okay, before going any further let's look back at some of the things we've seen so far.



3204. JK Flip-flop

The JK flip-flop is the most widely used flip-flop in digital equipment. Like the D flip-flop, it is not subject to race conditions with the attendant undefined output. The JK flip-flop, unlike any of the other flip-flops presented so far, has four possible input combinations: no change, set, reset, and change (toggle). We'll look at this closer in the following discussion.

a. Circuit. Figure 3-8 shows a simple JK flip-flop. In this circuit the R and S data inputs are replaced by J and K with J equal to S and R equal to K. Perhaps the first difference you'll notice in this flip-flop is how the outputs are fed back to the inputs. This accounts for the uniqueness of this flip-flop, as well as the elimination of race and it's ability to toggle (change).

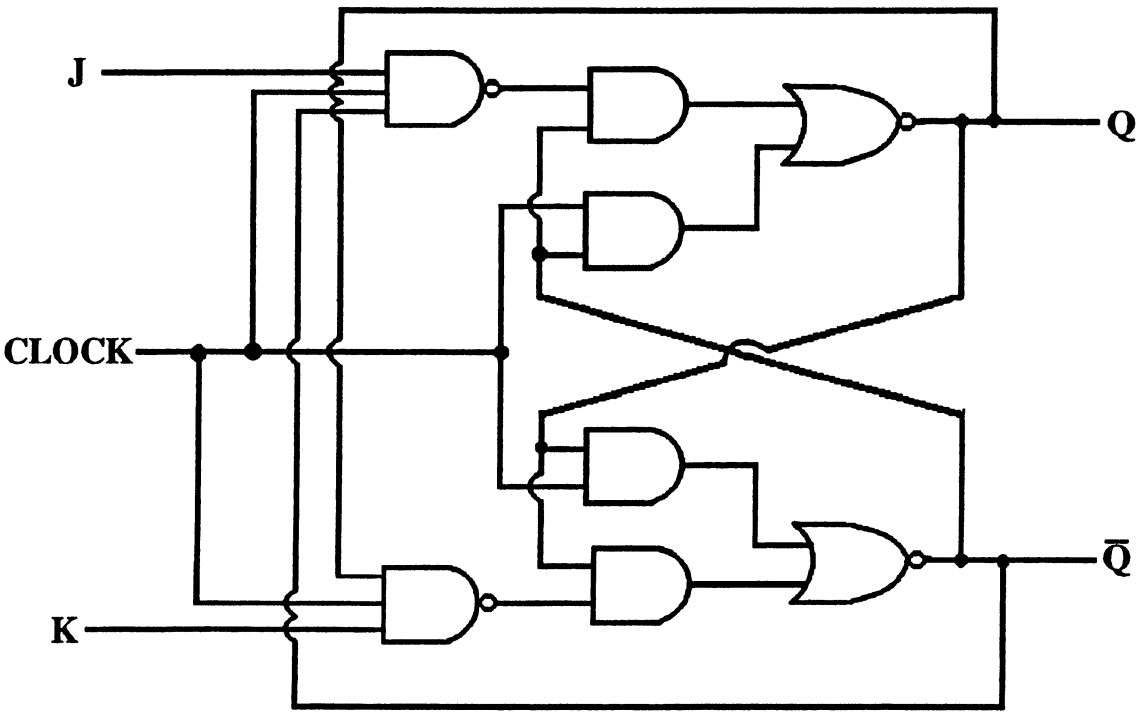


Fig 3-8. JK flip-flop.

b. Truth table. The input conditions shown in this truth table are for J and K status at the time of the clock pulse. The output conditions shown in this truth table are a reflection of Q's state immediately following the clock pulse. In other words, this truth table does not show clockless conditions.

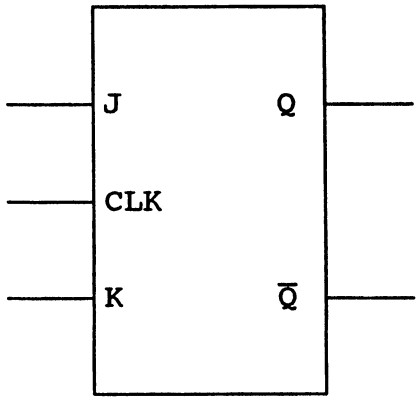
J	K	OUTPUT Q
0	0	NO CHANGE
1	0	0
0	1	1
1	1	CHANGES TO OPPOSITE

$\overline{Q}$

So tell me what's going on with $\overline{Q}$ during the sequence in the truth table above? Put your answer in the mini truth table above.

Your answer should have been, in order from top to bottom, no change, 1, 0, and changes to opposite. You knew this because Q and $\overline{Q}$ always complement each other.

c. Logic symbol. The logic symbol for this circuit (JK flip-flop) follows.

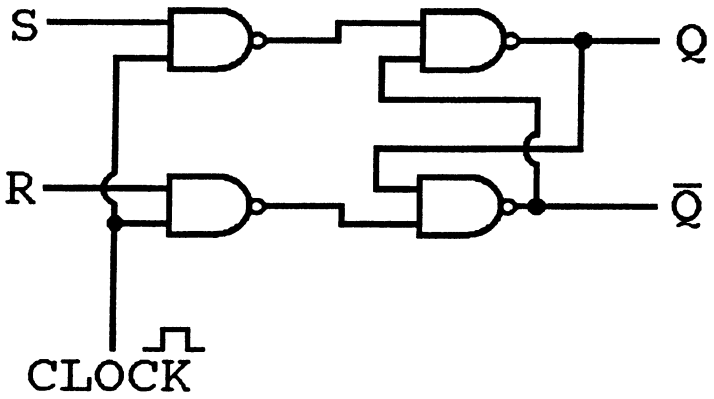


3205. Master-slave Flip-flops

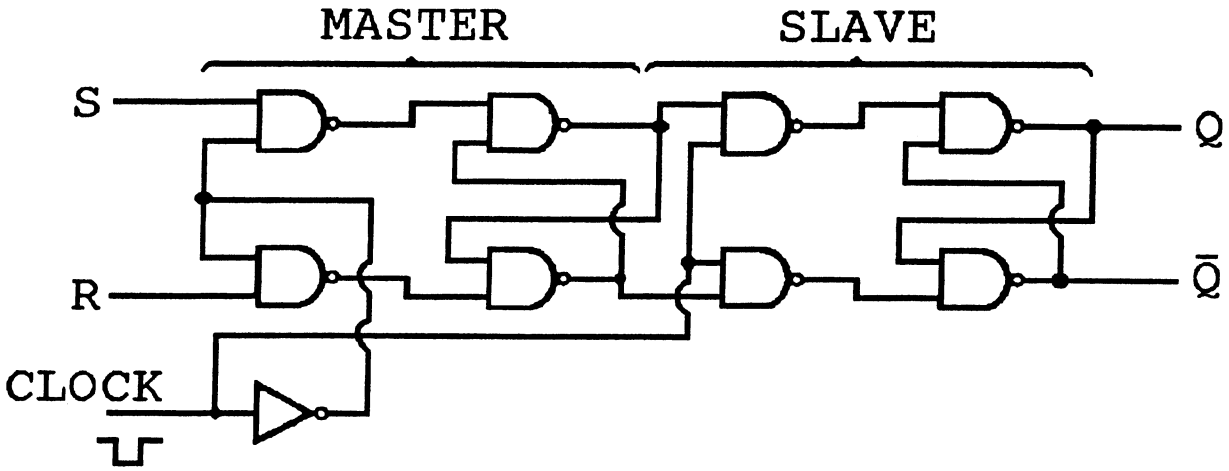
The flip-flops presented so far have been straightforward and simple. Now it's time to look at somewhat more complicated flip-flops. The complexity is not to make your job harder, it is only to make these circuits function better.
The master-slave relationship does two important things:

- eliminates the possibility of race
- allows flip-flops to be cascaded (connected together input to output)

Before we actually look at a master-slave flip-flop, let's start with a simple clocked RS flip-flop, then add a second stage to make it a master-slave configuration.

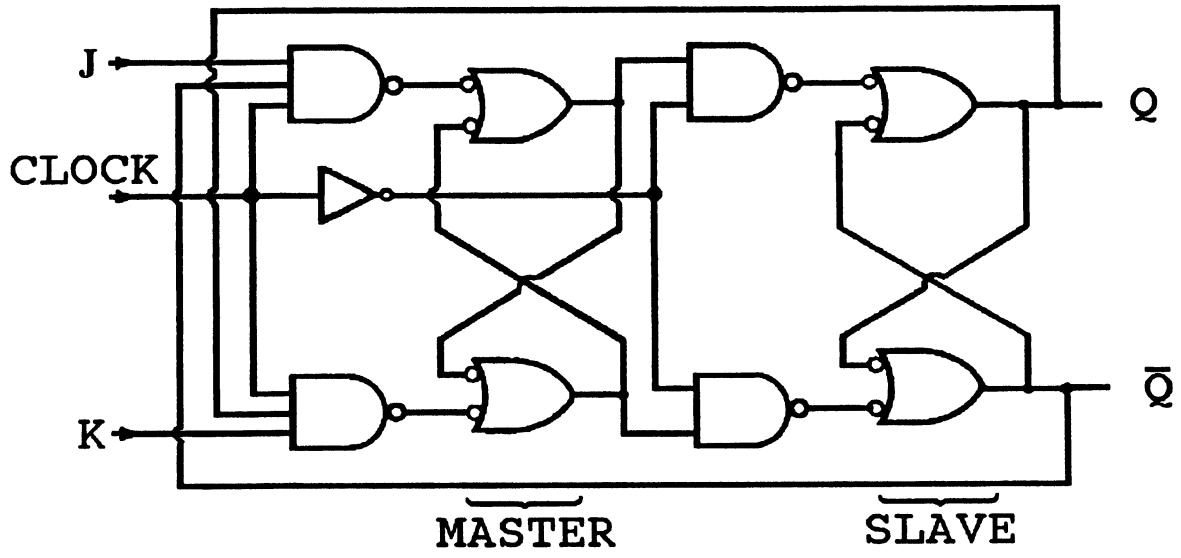


If we cascade another flip-flop in front of this and invert the clock line to the first stage, what we will have is a basic master-slave flip-flop:



Now, what will happen in this circuit is that data will be accepted into the master portion of the circuit when the clock is low. It will then wait there until the clock is high. Now the data can be transferred to the slave portion of the flip-flop.

Let's look at a JK master-slave flip-flop.

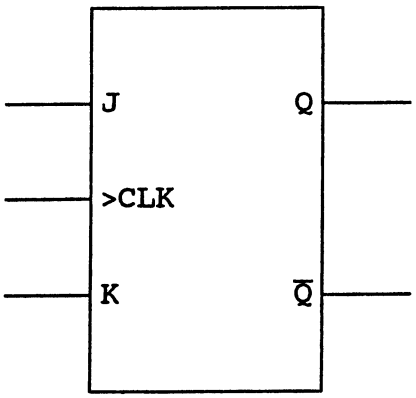


In this JK flip-flop you can see the same type of relationship exists as did in the basic RS master slave flip-flop. In this edge-triggerd flip-flop, the positive edge of the clock enables the master portion of the flip-flop. Data can be accepted at this time and the master portion will either set or clear. At the same time that the positive edge of the clock is felt on the master, its complement is felt on the slave input, thus isolating the slave portion. The positive edge (it goes through an inverter) of the clock then enables the slave portion of the flip-flop. When this happens, the data from the master is transferred to the slave portion of this flip-flop.

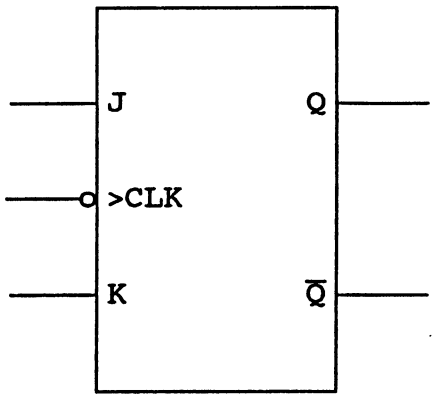
3206. Logic Symbol Elements

On the next page (table 3-1), you can find some of the common logic symbols associated with flip-flops. These logic symbols show different types of triggerng on the clock as well as the preset and clear inputs. In the logic symbols, notice that this symbol > indicates edge, while again this symbol o indicates a low as the active condition. Though not shown here, these symbols apply to the data inputs also, whether they are J and K, S and R, or D.

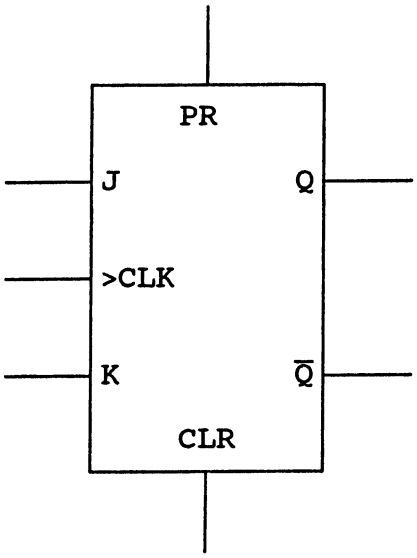
Table 3-1. Logic Symbols



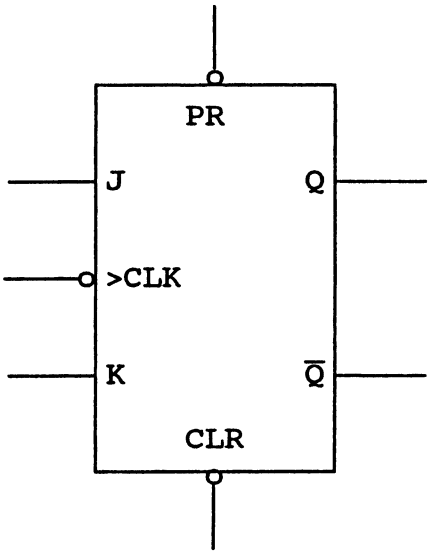
Positive edge trigger



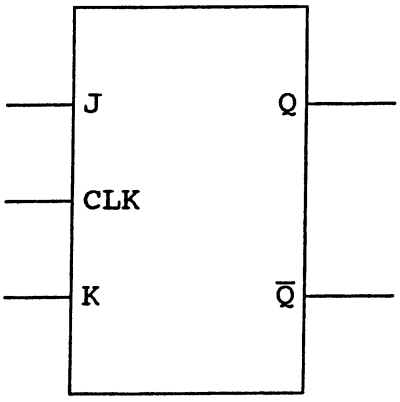
Negative edge trigger



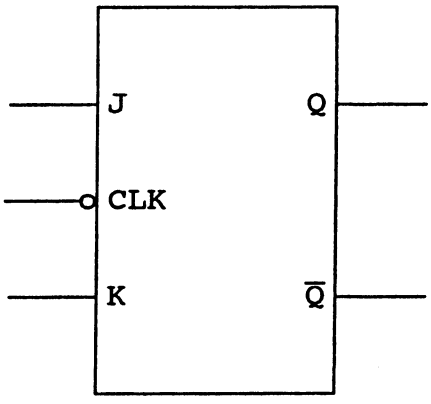
High level preset and clear



Low level preset and clear



High level trigger

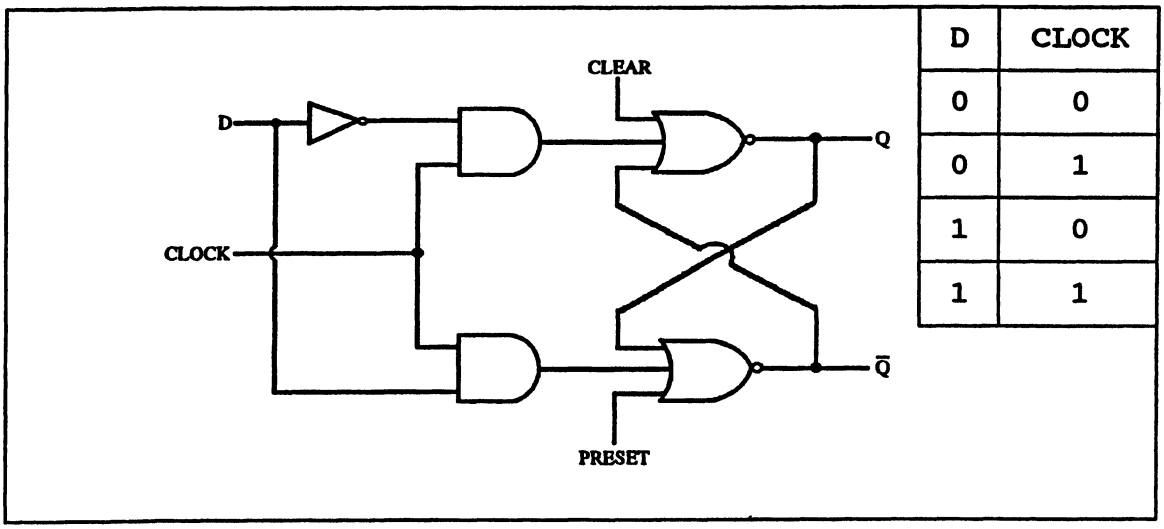


Low level trigger

Lesson Summary. Flip-flops are found extensively throughout digital equipment. A thorough knowledge of these circuits is important to the electronic repairman--for it is your understanding of these little circuits that will make the complex ones easier to troubleshoot and repair.

Exercise: Complete items 1 through 12 by performing the action required. Check your responses against those listed at the end of this study unit.

Refer to the following figure for item 1.



D	CLOCK
0	0
0	1
1	0
1	1

1. Select the proper values for Q and not $\overline{Q}$.

a.

Q	$\overline{Q}$
no change	
0	1
no change	
1	0

c.

Q	$\overline{Q}$
1	0
0	1
1	0
0	1

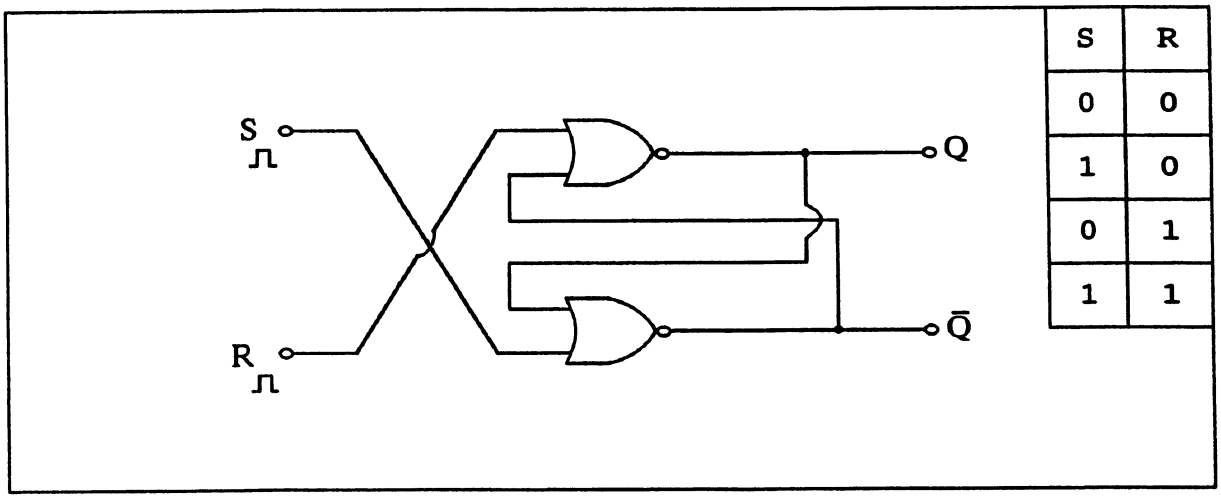
b.

Q	$\overline{Q}$
no change	
0	1
1	0
disallowed state	

d.

Q	$\overline{Q}$
no change	
1	0
no change	
0	1

Refer to this figure for question 2.



S	R
0	0
1	0
0	1
1	1

2. Select the proper values for Q and not $\bar{Q}$.

a.

Q	$\bar{Q}$
no change	
1	0
0	1
disallowed state (both 0's)	

b.

Q	$\bar{Q}$
disallowed state	
1	0
0	1
no change	

c.

Q	$\bar{Q}$
no change	
0	1
1	0
disallowed state (both 1's)	

d.

Q	$\bar{Q}$
no change	
0	1
1	0
no change	

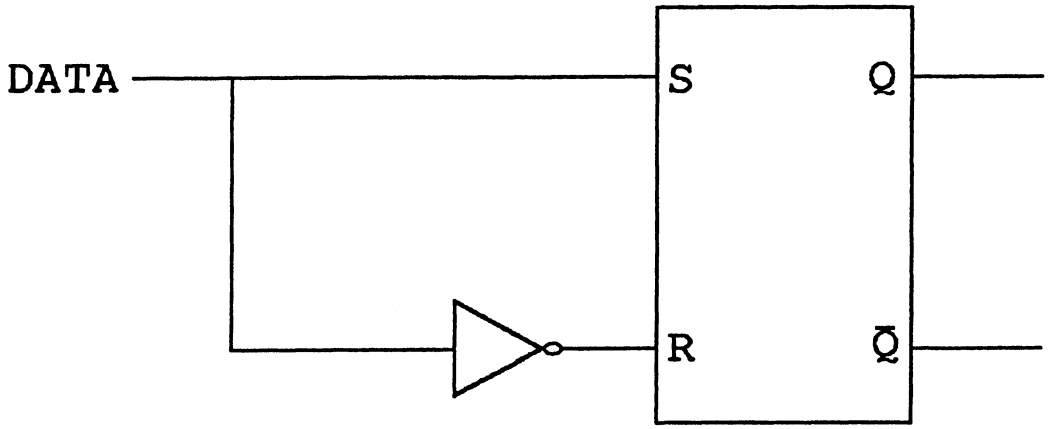
3. Preset and clear, in relation to bistable devices, can best be described as

   a. signals that allow a flip-flop to be set or reset without a signal on either the clock or data input lines.
   b. the two states of a flip-flop or latch.
   c. a condition where $Q = 0$.
   d. a condition where $\overline{Q} = 0$.

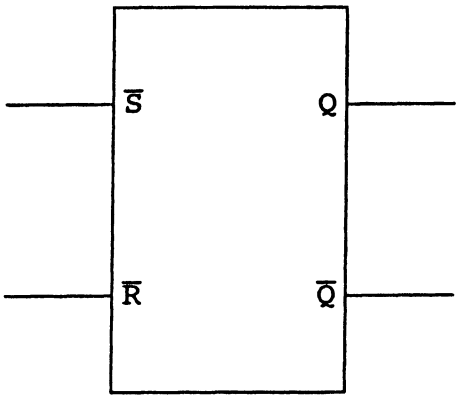
4. Race can best be described as a condition

   a. in which the output of a bistable device cannot be predicted because the device has been driven into the disallowed state.
   b. generated by a flip-flop changing states too rapidly.
   c. in which a bistable device fails to maintain its memory.
   d. in which the output of a bistable device locks at zero because the device has been driven into the disallowed state.

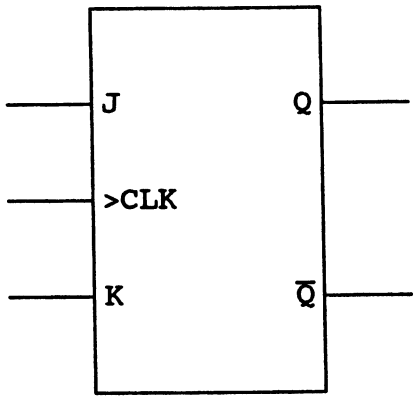
5. Shown below is the logic symbol for a(n) _____.



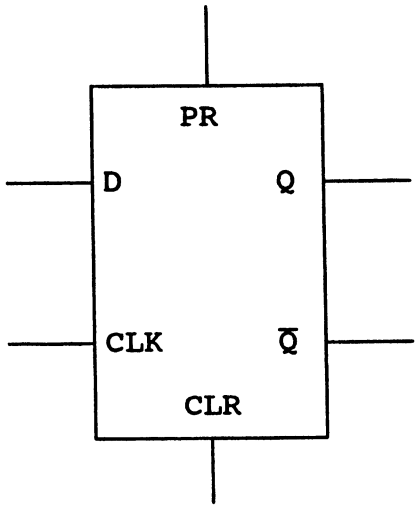
6. Shown below is the logic symbol for a(n) _____.



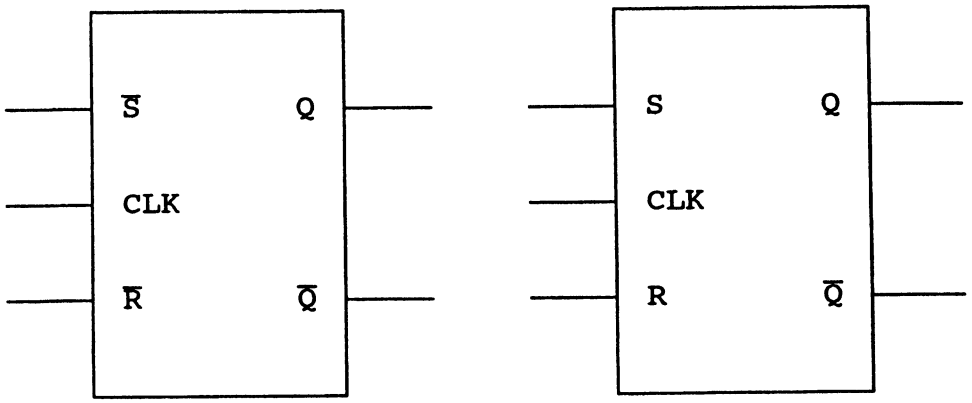
7. Shown below is the logic symbol for a(n) _____.



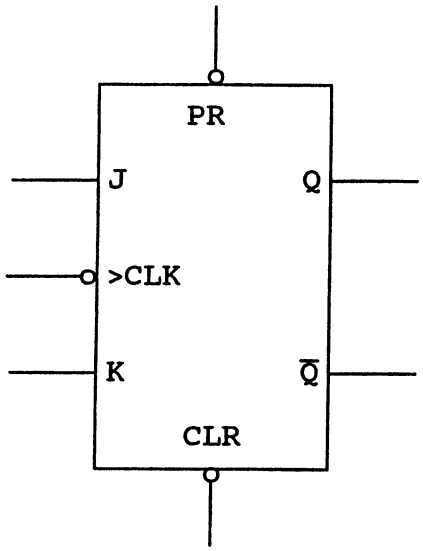
8. Shown below is the logic symbol for a(n) _____.



9. Shown below are the logic symbols for _____.

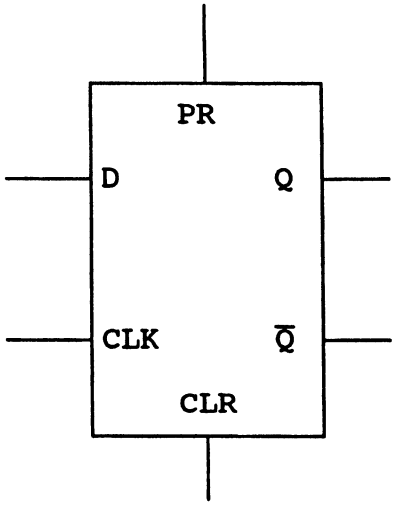


10. Which statement is true about this logic symbol?



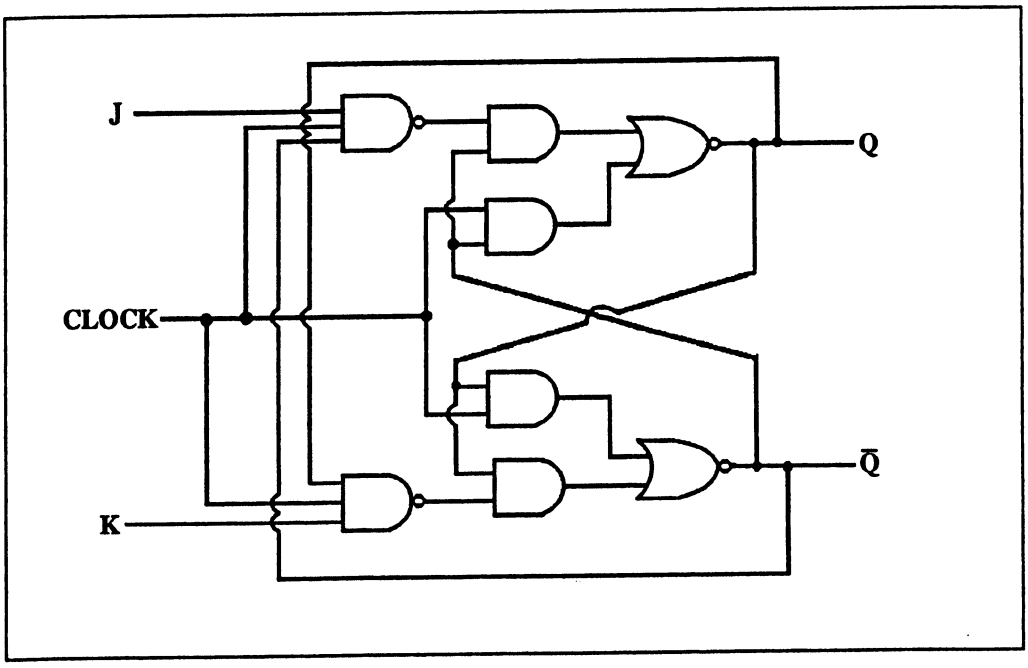
a. The J and K inputs are high level triggered, the clock is negative edge triggered, and the preset and clear are negative edge triggered.
b. The J and K inputs are high level triggered, the clock is negative edge triggered, and the preset and clear are low level triggered.
c. This flip-flop is not widely used.
d. In order to preset or clear, there must be a low on the clock line.

11. Which statement is true about this logic symbol?



a. The disallowed state is initiated by lows on the input lines.
b. The disallowed state is initiated by highs on the input lines.
c. The only time there is a change in the output is when the clock is 0.
d. There is no disallowed state because one input is inverted.

Refer to this figure for question 12.



12. Which truth table below is derived from this flip-flop?

a.

J	K	OUTPUT Q
0	0	no change
1	0	0
0	1	1
1	1	changes to opposite

b.

J	K	OUTPUT Q
0	0	no change
1	0	1
0	1	0
1	1	changes to opposite

c.

J	K	OUTPUT Q
0	0	changes to opposite
1	0	0
0	1	1
1	1	no change

d.

J	K	OUTPUT Q
0	0	changes to opposite
1	0	1
0	1	0
1	1	no change

Lesson 3. Adders, Subtractors, Counters, and Registers

LEARNING OBJECTIVES

1. Construct a truth table for a digital adder.

2. Given the starting contents, the number of pulses, and the time period, determine the output of a counter.

3. Given any counter, determine its configuration.

4. Given any counter, identify the type.

5. Given the starting contents, the number of input pulses, and the time period, determine the contents of a register.

6. Given any register, determine its format.

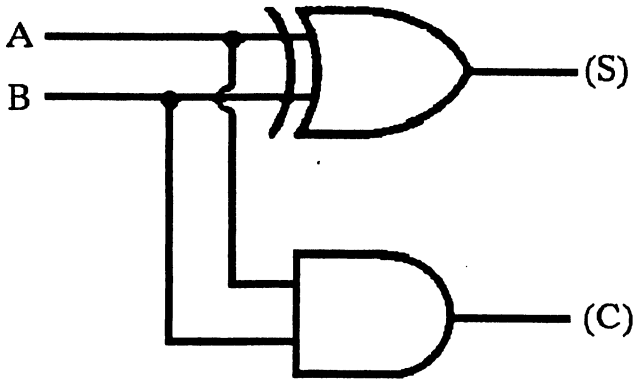
7. Given any register, identify its type.

In this lesson, we'll look at circuits that store, add, subtract, divide, and count--in binary, of course. The majority of the circuits we will look at take advantage of the flip-flop's ability to store logic levels. The lesson will begin with a few simple non-flip-flop circuits, giving you a short break from latches and flip-flops. But, immediately after that and through to the end of the lesson, latches and flip-flops will be central to the circuits presented.

3301. Adders and Subtractors

As promised, in these first circuits, you will notice that there are no flip-flops in sight.

a. Half adder. Figure 3-9 shows a logic diagram and truth table for a half adder. This adder has two inputs, A and B. It also has two outputs, S (sum) and C (carry). As you can see, the inputs to the two gates are in parallel.



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

Fig 3-9. Half adder and truth table.

An examination of the truth table reveals the following:

(1) The sum (S) output is high when A and B are opposite logic levels.

(2) The sum output is low when A and B are the same logic level.

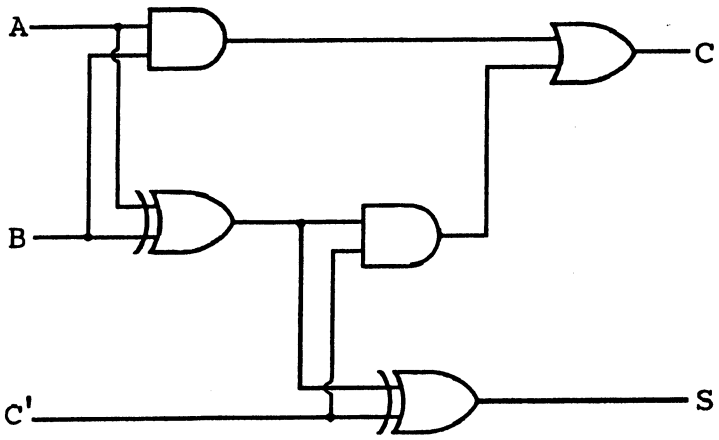
(3) The carry (C) output is high only when A and B are both high.

(4) The outputs of this circuit follow the rules for binary addition. The rules (as you're sure to remember), along with the sum and carry outputs, are shown below.

Rules	Sum	Carry
0 + 0 = 0	0	0
0 + 1 = 1	1	0
1 + 0 = 1	1	0
1 + 1 = 0 carry 1	0	1

The dual inputs of this adder limit it to adding only two bits (two binary digits) at one time.

b. Full adder. Figure 3-10 depicts another adder. This circuit is nothing more than two half adders with their carry outputs ORed together. This adder not only can add the information on the A and B data inputs, but it is also able to add data on the C' (C prime) data input, which is a "carry in" from a preceding stage.



A	B	C'	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fig 3-10. Full adder and truth table.

c. Subtractor. Figures 3-11 and 3-12 show a half subtractor and a full subtractor. These circuits follow the rules for binary subtraction, as shown below.

Rules

0 - 0 = 0
1 - 0 = 1
1 - 1 = 0
0 - 1 = 1 borrow 1

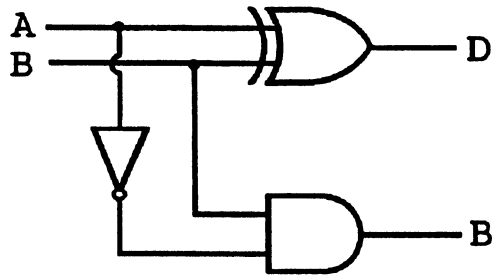


Fig 3-11. Half subtractor

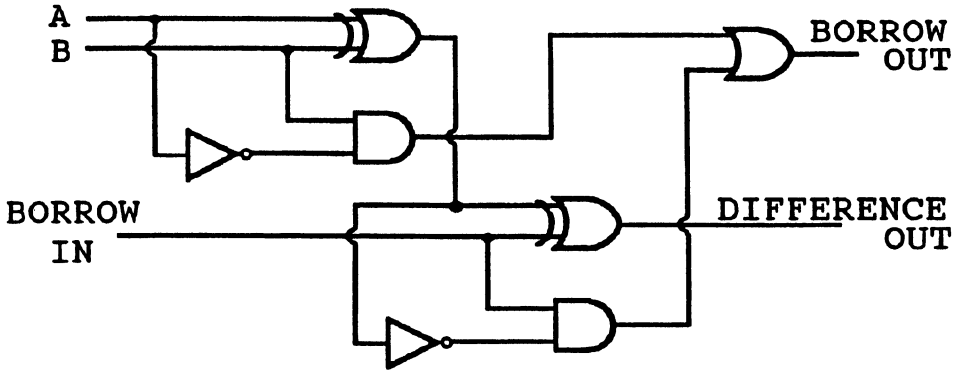


Fig 3-12. Full subtractor.

Notice that the full subtractor has one more input than the half subtractor. This additional input is the "borrow in" input. This configuration is more versatile than the half subtractor because it allows the full subtractor to be used in conjunction with other subtractors.

Digital machines, for the most part, don't use these type circuits to subtract. As you learned in section 1203, digital machines perform mathematical operations through addition. These two subtractor circuits were presented here because in the course of your work you may come in contact with them.

Adders can also perform subtraction. This is done with the help of a controlled inverter circuit which uses processes known as twos and ones complementation.

3302. Counters.

A counter is a circuit that counts input pulses. These circuits are also known as dividers or accumulators. Counters output a pulse for every one or more input pulses. Because they can be configured to output a pulse for any number of input pulses, counters can be used to divide as well as count. These devices are basically flip-flops or latches cascaded together. Counters are structured in two basic configurations, asynchronous and synchronous.

- Asynchronous counters are also known as serial or ripple-carry counters. Each latch or flip-flop in these counters uses the previous stage's output as an input. Because of this dependency (that is, the change of state of the latch or flip-flop depends on the output of the previous latch or flip-flop), the action of the latches or flip-flops is sequential in a serial counter.

- Synchronous counters are also known as parallel counters. The flip-flops or latches in these counters are configured so that they may be triggered simultaneously. That is, each flip-flop or latch in the counter can change states simultaneously .

a. T-latch counter. Figure 3-13 shows a logic diagram for a serial counter. This counter is composed of T-latches.

A T-latch is a very simple device. It has one data input that is labeled T, and, of course, two outputs labeled Q and $\overline{Q}$. These latches are positive edge triggered and simply toggle (change to the opposite state) when they receive the correct signal on the T input. The T-latches in the counter of your example (fig 3-13) also have preset inputs.

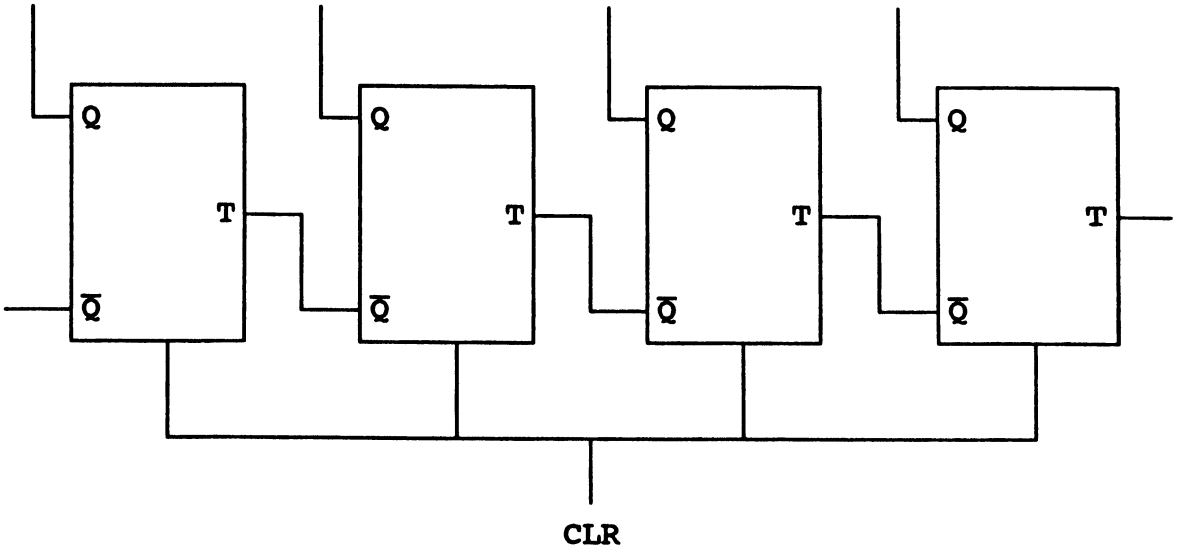


Fig 3-13. T-latch serial counter.

You should notice, though, that the position of the inputs and the outputs are reversed on the latches (fig 3-13). This is not the normal position for the inputs and outputs. Normally the inputs would be on the left and the outputs would be on the right. Only in this example will the latches be presented in this fashion. This reversal has been done to help you read the number that this counter will produce as you explore counters for the first time.

Let's clear this counter by pulsing the clear line.

What will this do?

Your answer should have been, "This input will make Q zero and $\overline{Q}$ one."

If you are confused about preset and clear, review section 3202c.

Figure 3-14 shows our counter after being cleared. The importance of preset and clear will be clearer to you after this example. Notice that each latch is storing a zero (low).

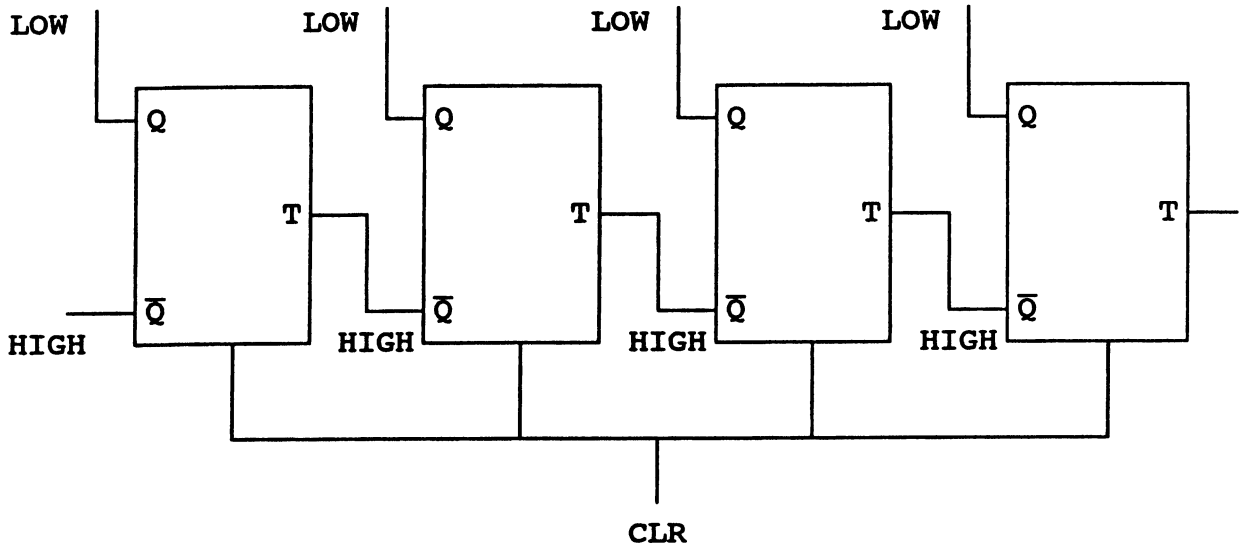


Fig 3-14. Counter after being cleared.

Now, let's pulse the input. Figure 3-15 shows the counter after the first input pulse.

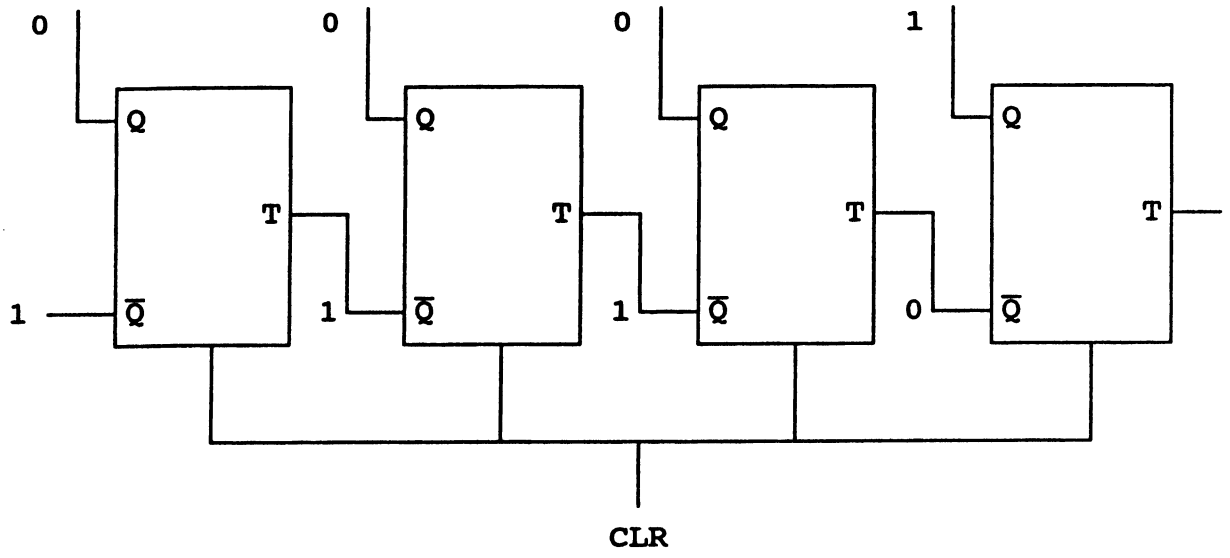


Fig 3-15. Counter after first input pulse.

Let's look at the state of our counter now. One latch is storing a one and the rest are still storing zeros. The counter is showing a count of 0001. Why did only one latch toggle? Because these latches react to pulses transitioning from low to high (positive edge trigger), not high to low. Let's send a second pulse and examine the counter again (fig 3-16).

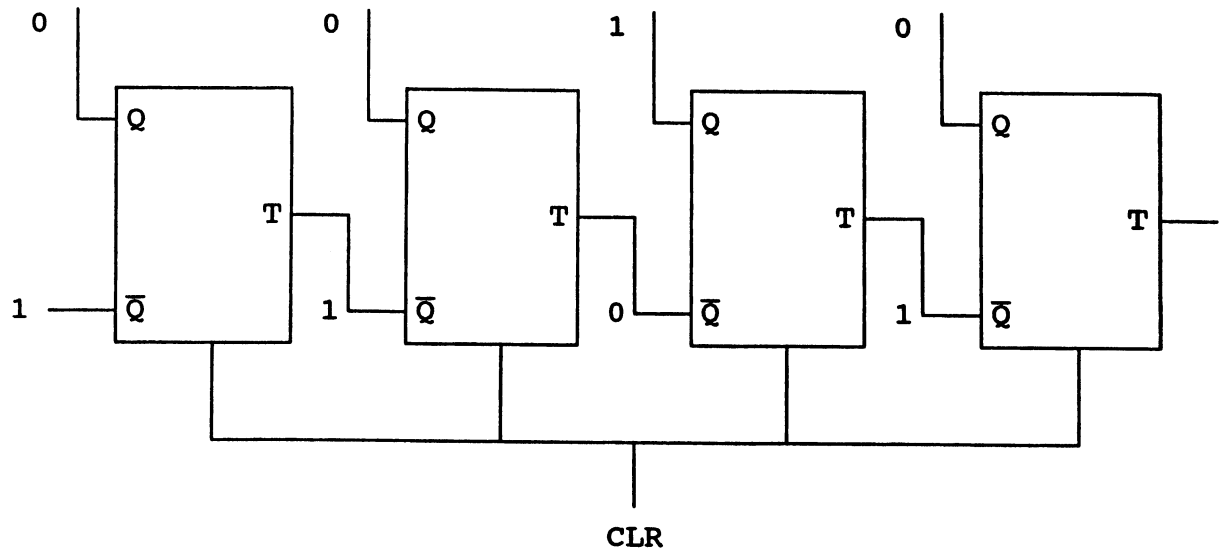


Fig 3-16. Counter after second input pulse.

Our counter is now showing a count of 0010 or 2. What happened? The latch closest (the first latch) to the input toggled from one to zero. What did you expect? The first latch's $\overline{Q}$ output (the input to the next latch) went from from zero to one, which made the second latch toggle. Now, let's pulse our counter for a third time, but this time you explain the sequence of events.

After the third pulse, the counter will hold a count of _____ .

The outputs of the first latch are:
Q is _____ and $\overline{Q}$ is_____.

The second latch did/did not toggle.
The outputs of the second latch are:
Q is _____ and $\overline{Q}$ is _____.

The third latch did/did not toggle.
The outputs of the third latch are:
Q is _____ and $\overline{Q}$ is _____.

The fourth latch did/did not toggle.
The outputs of the fourth latch are:
Q is _____ and $\overline{Q}$ is _____.

After the third input pulse, our counter displays a 0011 or 3. The pulse on the input line causes the first latch to toggle. Now, the output of the first latch is Q high and $\overline{Q}$ low.

The second latch doesn't toggle because its input (the $\overline{Q}$ output from the first latch) went from a one to a zero which won't cause it to toggle. Consequently, its outputs remain Q high and $\overline{Q}$ low.

The third and fourth latches don't toggle either because neither latch gets an input from the stages that precede them. Consequently, these outputs remain Q low and $\overline{Q}$ high. Figure 3-17 shows our counter after the third data input. This T-latch counter is easy and staightforward. Now, on to other counters.

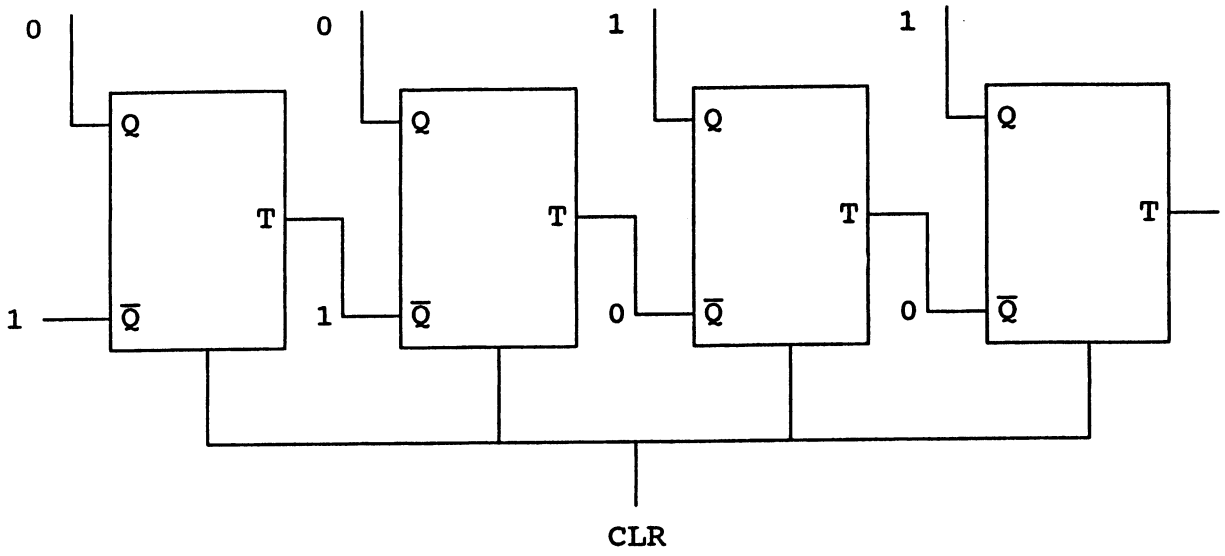


Fig 3-17. Counter after the third input pulse.

The first counter was presented as a teaching tool, designed to promote your understanding of a counter's operation. The number produced was read from left to right. The counters that follow will be shown as you would find them in a logic diagram or schematic. Consequently, the MSB (most significant bit) will be to the right and the LSB (least significant bit) will be to the left. What this means to you is that all numbers the counters produce are read from right to left.

The MSB means the digit with the highest value.
The LSB means the digit with the lowest value.

MSB	LSB
11 1001	11 100100

b. Four-bit binary counter. Figure 3-18 shows a logic diagram for a four-bit binary counter. There are two things you should note.

(1) The clock input is paralleled into each gate. This allows the flip-flops to be changed simultaneously. This configuration is known as a synchronous counter.

(2) The **AND** gates could be omitted and the counter would still count as intended. The **AND** gates in this circuit are used to speed the action of this counter.

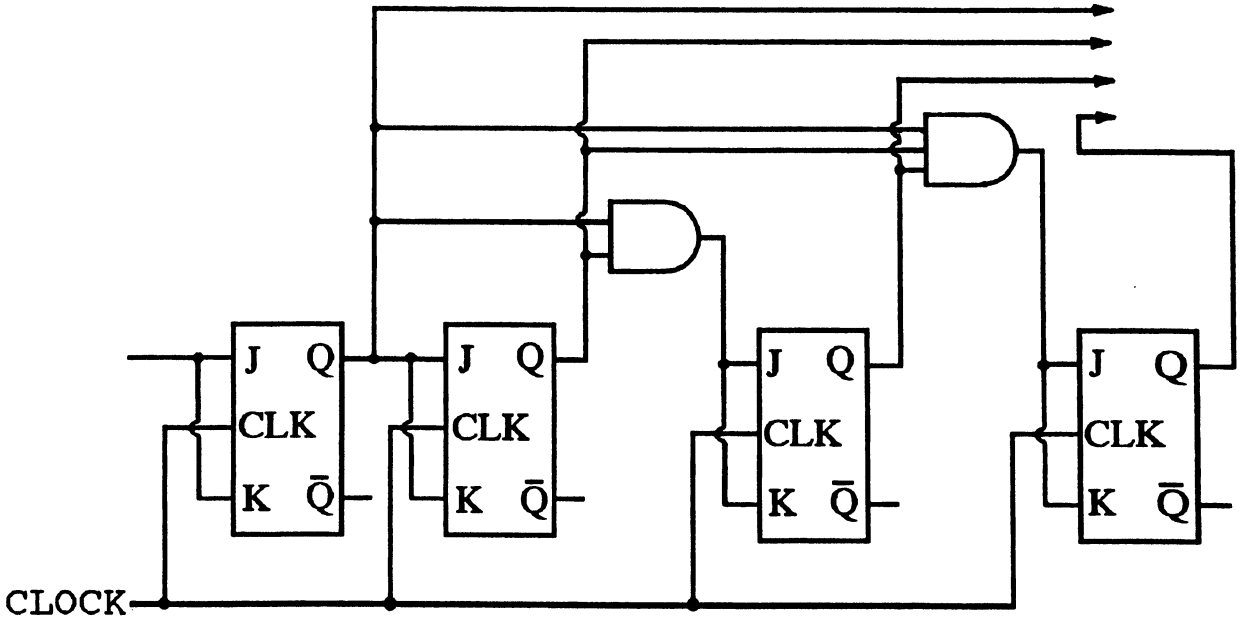


Fig 3-18. Four-bit binary counter.

If the **AND** gates were omitted from this circuit, would it still be a synchronous counter?

Good, you're thinking. The circuit without the **AND** gates would be asynchronous.

c. Asynchronous binary counter. In the binary counter shown in figure 3-19, the J and K inputs are held logic high; this allows the flip-flops to toggle as they receive inputs on the clock line. This counter is very similar to the first counter in this lesson.

Let's trace a pulse through this counter. Wait. Before we can do that, it is important to note three things: 1) these flip-flops are negatively edge triggered, 2) each output is a weighted output, and 3) we will assume that all the flip-flops are reset.

Okay, now let's trace signals through the counter.

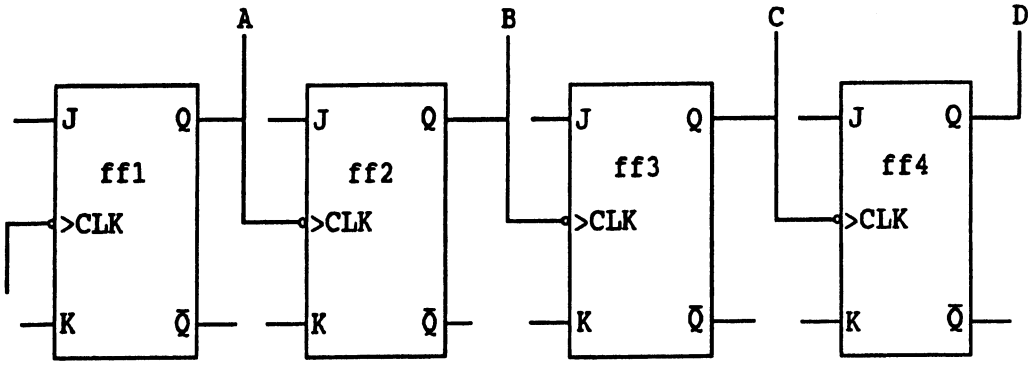


Fig 3-19. Asynchronous binary counter.

The first pulse will cause ff1 to toggle. So Q of ff1 will go from 0 to 1. This means that a 1 (high) will be felt at A.

What happens to ff2?

Nothing! A transition from high to low triggers these flip-flops.

A second pulse will toggle ff1 again, making Q of ff1 go from 1 to 0 (high to low). So of course ff2 will toggle making Q of ff2 go from low to high. Now there is a zero on A and a one on B.

There is no need to trace more signals through this counter because you already understand the basic concept.

Before we leave this counter, lets look at what is meant by a weighted output. This means that each output when high has a specific value. In this case A = 2^0 or 1, B = 2^1 or 2, C = 2^2 or 4, and D = 2^3 or 8.

d. <u>Decade counter</u> Wouldn't it be nice to have a counter count in decimal instead of binary? A counter can be configured to do just that. Figure 3-20 shows an uncomplicated circuit that counts to nine then resets.

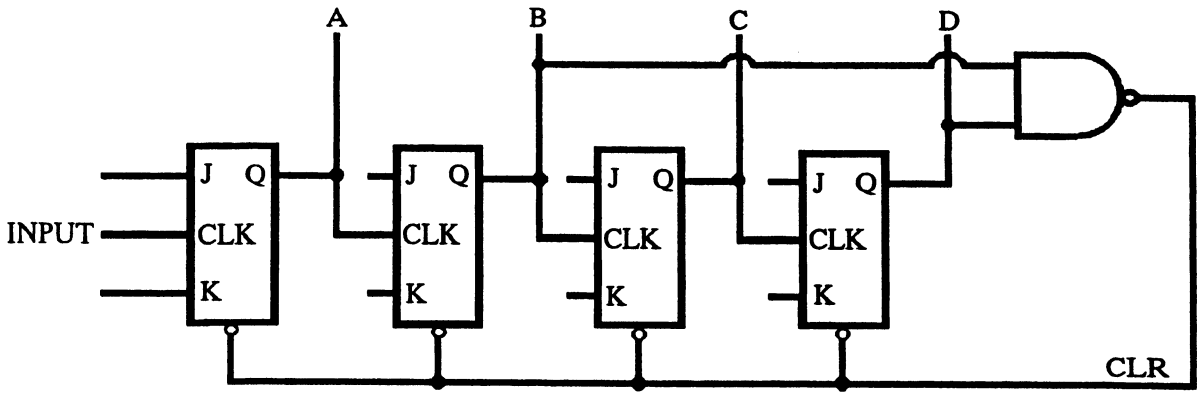


Fig 3-20. Decade counter.

This counter is similar to the counter in figure 3-19, with the exception of adding a **NAND** gate and clear inputs to the flip-flops. This counter counts to 1001 or 9 exactly as the previous counter did. When the counter is at this point, A is high and D is high. Now, with the next input, A goes from high to low which toggles the next flip-flop and drives B high. With B and D high, the **NAND** gate will output a low which will clear the counter (that is, all Q's become zero). The high to low transition on D caused by the counter being cleared would cause the next stage to increment. If these counters were cascaded, the output at D would also be the input to the next counter.

3303. Registers

Registers are used to store information, but unlike a flip-flop, a register can store more than one bit of information at a time. Registers are also used to act as buffers, to generate sequences, to divide, and to convert information from serial to parallel format and vice versa. Format, as it applies to registers, refers to the method of inputting and outputting information. Register format will be either serial or parallel.

- Serial format describes a configuration that allows information to be transferred in and/or out--**one bit at a time only.**

- Parallel format describes a configuration that allows information to be loaded or unloaded in a single operation--**more than one bit at a time.**

We will look at a simple register composed of latches, then we'll go on to the common shift registers.

a. Latch-type register. The register shown in figure 3-21 may appear at first glance to be complicated, but a closer inspection will allow you to see that this circuit is very uncomplicated.

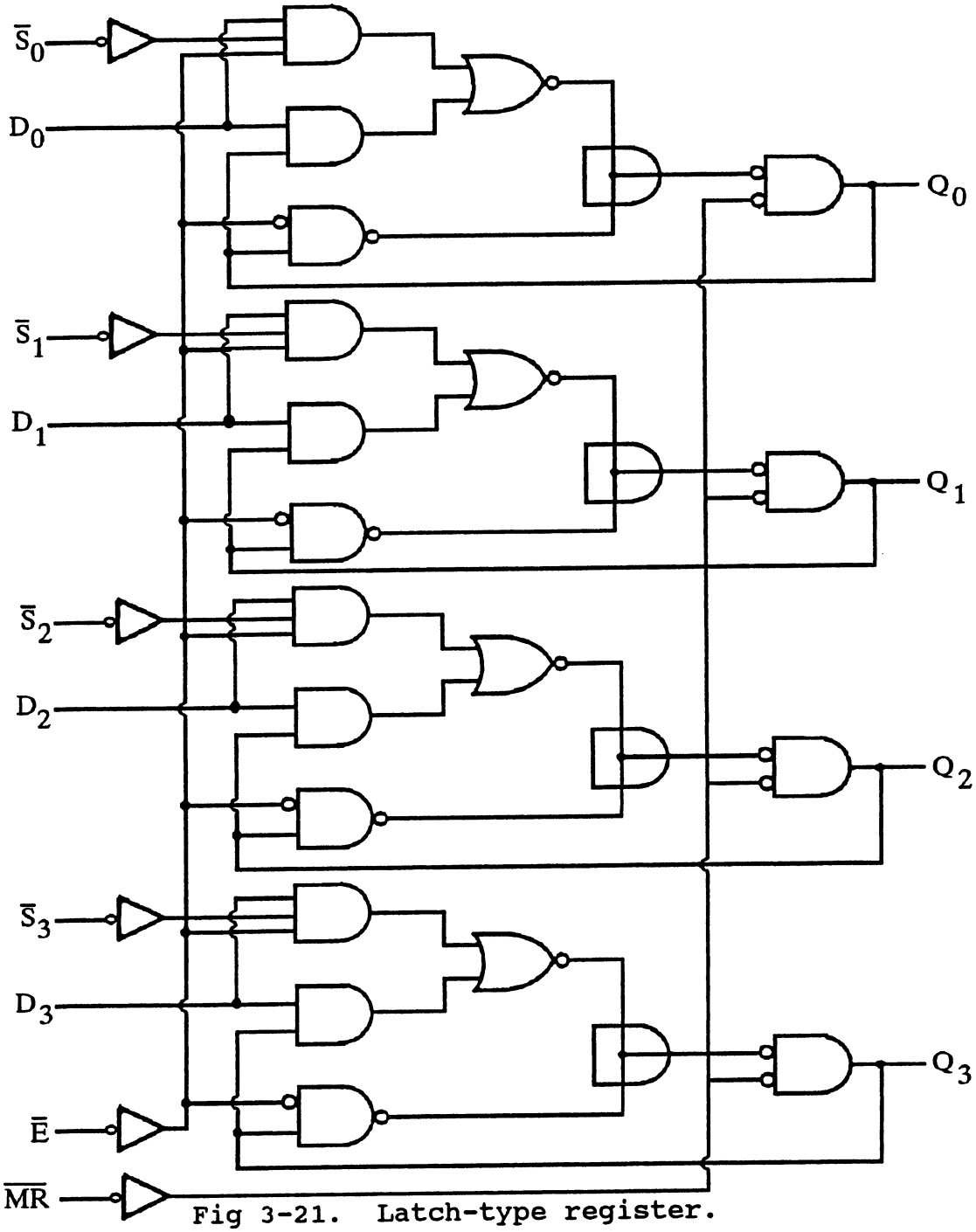


Fig 3-21. Latch-type register.

This four-bit or one nibble (four bits equals one nibble) register is composed of D latches. The data inputs to this circuit are D_0, D_1, D_2, and D_3. $\overline{S}_1$ through $\overline{S}_3$ are the set inputs, while Q_0 through Q_3 are the outputs. Possibly, the only new items to you in this circuit should be $\overline{E}$ and $\overline{MR}$. $\overline{E}$ is an enable input which, when active (low), allows data to be input to the circuit. $\overline{MR}$ is a master reset input which, when active (low), will reset (clear) the output to 0 (that is, all Q's become 0).

Is this a parallel or a serial register?

It is a parallel register because it can accept and/or output more than one bit (in this case, four bits) of information at a time.

b. Four-bit shift register. Figure 3-22 is a logic diagram for a shift register. This register uses D flip-flops. The register's data input is also the first flip-flop's data input. The output of each flip-flop is fed to the data input of the successive flip-flop.

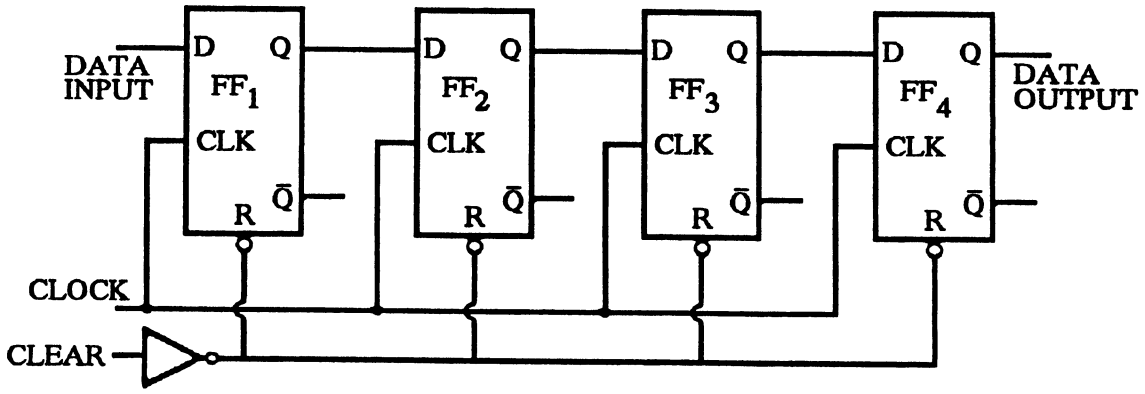


Fig 3-22. Four-bit shift register.

Let's trace a data bit from the register's input to its output:

Remember: These flip-flops cannot change unless an active clock pulse is present at the clock input.

(1) During the first clock pulse, a logic low is input and stored in the first flip-flop.

(2) During the second clock pulse, this 0 bit is shifted, or stepped, to the next flip-flop.

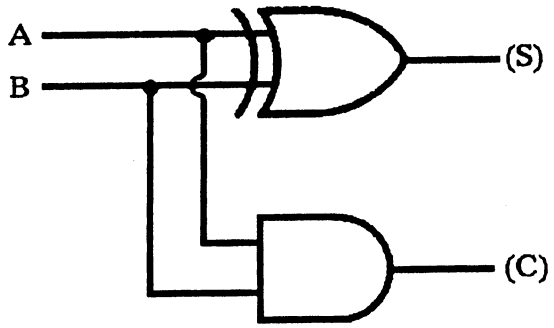
(3) During the next clock pulse, the 0 bit is shifted to the third flip-flop.

(4) During the fourth clock pulse, the data is shifted to the last flip-flop where the output is now available as the register's output.

Lesson Summary. Understanding the basic circuits presented in this lesson will strengthen your knowledge of digital logic. In this lesson we combined flip-flops and latches into working circuits. These circuits, as well as variations of these circuits, are the constants of digital logic.

Exercise: Complete items 1 through 8 by performing the action required. Check your responses against those listed at the end of this study unit.

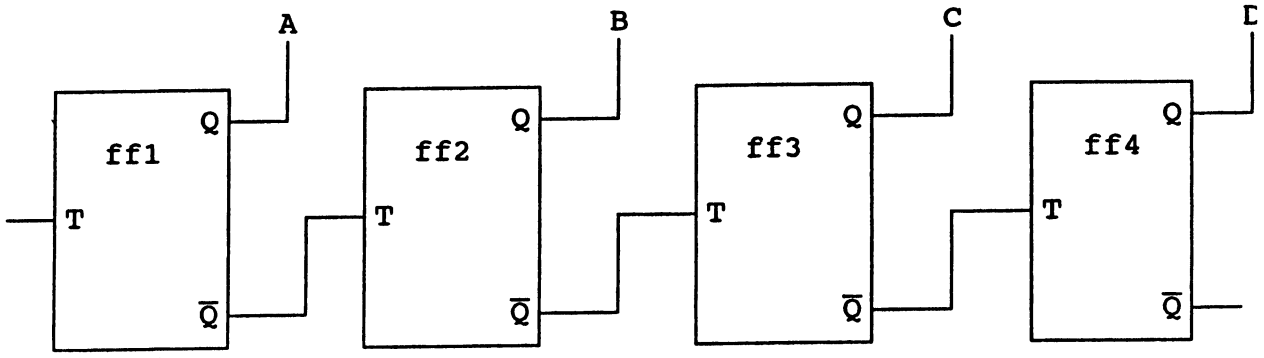
Refer to the following figure for item 1.



1. Complete the truth table for this circuit.

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

For items 2 and 3, refer to the following figure.

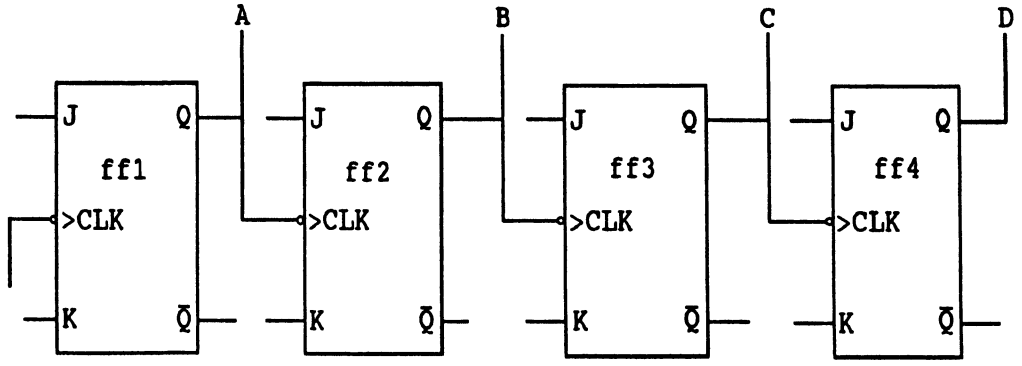


2. This is a logic diagram for a _____ counter.

3. This is a(n) _____ counter.

a. synchronous
b. parallel
c. asynchronous
d. 5-bit

For items 4 and 5, refer to the following figure. Assume that J and K are held logic high and the device is cleared immediately prior to the conditions stated in each item.

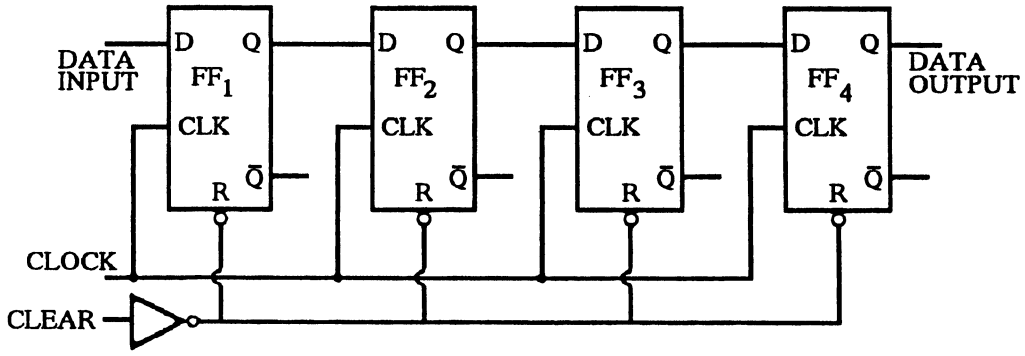


4. After five input pulses, this device is storing the number_____ .

5. After six input pulses, what are the logic levels felt at each of the following points?

A _____
B _____
C _____
D _____

For items 6 through 8 refer to the following figure.



6. This is a logic diagram for a(n) _____ register.

7. The format of this register is _____.

8. It takes _____ input pulses before this register produces an output.

UNIT SUMMARY

This study unit taught common digital logic circuits as well as the concepts needed to understand them. The circuits developed in this study unit will help you learn the troubleshooting techniques of the next study unit.

Lesson 1 Exercise Solutions

		Reference
1.	a.	3101b
2.	a.	3101c
3.	c.	3101d
4.	c.	3101a

Lesson 2 Exercise Solutions

		Reference
1.	a.	3203b
2.	a.	3201c
3.	a.	3202d
4.	a	3201c
5.	D latch	3203a
6.	RS latch	3201d
7.	JK flip-flop	3204c
8.	D flip-flop	3203b
9.	RS flip-flops	3202
10.	b.	3206
11.	d.	3203b
12.	a.	3204b

Lesson 3 Exercise Solutions

		Reference
1.	S C 0 0 1 0 1 0 0 1	3301a
2.	T-latch counter	3302a
3.	c.	3302
4.	5	3302c
5.	A is low, B is high, C is high, and D is low.	3302c
6.	Shift	3303b
7.	Serial	3303
8.	4	3303b

STUDY UNIT 4

INTERPRETING DIGITAL LOGIC CIRCUITS

Introduction. Finally, it's time to draw on the knowledge you've acquired in the previous study units and apply it. The key to successfully completing this section is a thorough knowledge of the principles of study unit 1, the gates of study unit 2, and the circuits presented in study unit 3. The electronic repairman is tasked with bringing his skills and knowledge to bear on problems confronted on the job, just as you will be in this study unit.

Interpeting logic diagrams, the first stage of troubleshooting is based upon a very simple premise: a knowledge of how a circuit should operate normally. You can't determine what's wrong if you don't know how it works. This study unit is the first step on the road to understanding digital logic circuits. Study unit 4 will nurture that understanding by using timing diagrams, truth tables, and equations to make schematics readable.

Lesson 1. THE TOOLS OF CIRCUIT EVALUATION

LEARNING OBJECTIVES

1. Given a digital electronic schematic, construct a truth table describing that circuit.

2. Given a digital electronic schematic, develop an equation describing that circuit.

3. Given a digital electronic schematic, and a timing diagram describing the input waveforms, construct a timing diagram describing the output waveform(s).

4. Given a digital electronic schematic, select from a list of timing diagrams, the timing diagram describing that circuit.

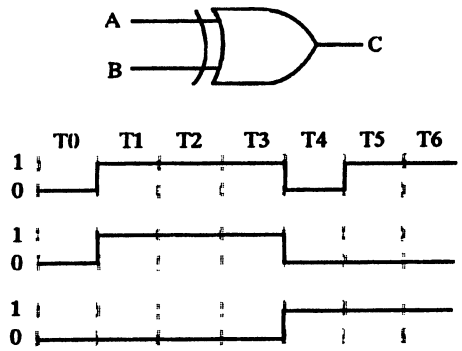
4101. Equations, Truth Tables, and Timing Diagrams

This lesson will introduce you to the combination of truth tables, equations, and timing diagrams. These three tools either singly or in combination will help you understand digital logic circuits. As usual, we'll start with fairly easy problems, then progress to more complicated circuit problems.

a. Overview. Combining a timing diagram, a truth table, and an equation allows you to completely evaluate a digital device. This combination forms a complete picture of the circuit. It allows you to see the relationship between the device and its input and output waveforms. You won't always be able to develop all three for a given device or circuit nor will you always want to. You need use only what will help you understand a circuit.

But even one of these troubleshooting "tools" will help you decipher various circuit problems. We will look at several familiar gates and circuits from this perspective.

(1) **XOR** gate. Figure 4-1 illustrates the truth table, equation, and a timing diagram for an **XOR** gate.



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

$$(\bar{A} \times B) + (A \times \bar{B}) = C$$

Fig 4-1. **XOR** gate.

Notice, that in the figure above:

* The equation is a synopsis of the truth table.
* The truth table is a statement of all possible conditions of the gate.
* The timing diagram is a visual representation of the relationship of input to output waveforms.

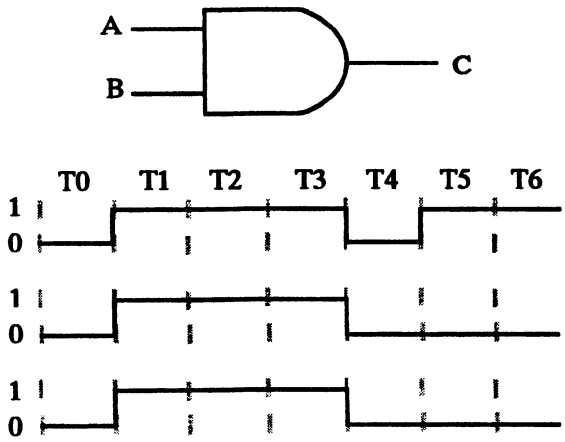
For all devices, the truth table, equation, and timing diagram singly or combined help you understand an electronic circuit.

(a) Equation. The equation is a succinct statement of the device/circuit. When an equation is derived for a more complex device, it puts that device in a clearer perspective. This is true because it expresses the mathematical relationship between input and output.

(b) Truth table. In most cases the truth table will list all possible input/output combinations for a circuit. This tabular listing of values represents the parameters of the circuit it is associated with.

(c) Timing diagram. This diagram is the most significant aid to troubleshooting. A timing diagram for one gate doesn't seem important. But--because it can give you a quick visual indication of the relationship between input and output it becomes an important tool when troubleshooting larger devices/circuits. The timing diagram uses the values of the truth table and the relationship of the equation to create a picture of a circuit's function. As this lesson progresses this will become clearer to you.

(2) **AND** gate. Here we see the equation, truth table, and a timing diagram for an **AND** gate. Again, none of this by itself is new, but the combination may be novel.

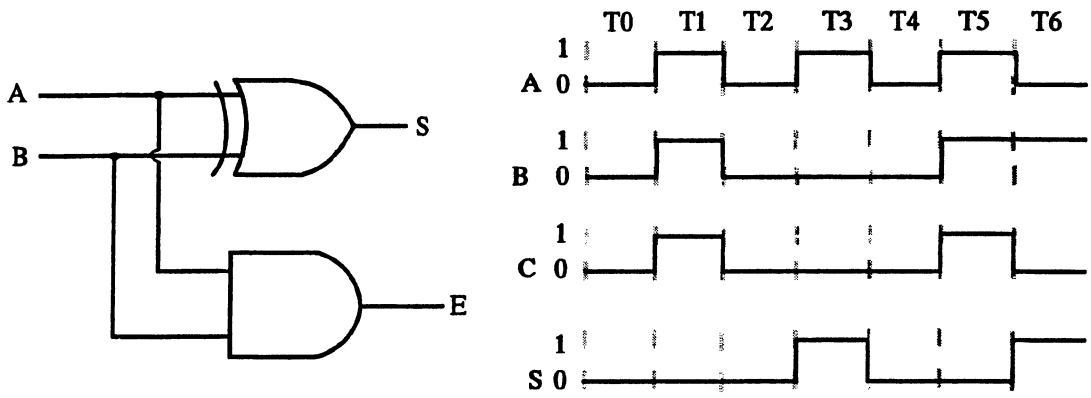


A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

A x B = C

Fig 4-2. **AND** gate.

(3) Half adder. Combining the two gates previously presented into this configuration will give you this familiar half adder.



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

$A \times B = C$

$(\overline{A} \times B) + (A \times \overline{B}) = C$

Fig 4-3. Half adder.

Again we have a timing diagram, a truth table and two equations to help make sense out of the circuit above. There are two equations because there are two outputs.

4102. Equations

In study unit 2 we briefly looked at equations for various gates. Equations can be derived for circuits as well as gates. We will look at circuit-derived equations briefly in the following section. A more in-depth discussion of this subject is outside the scope of this course.

a. Two AND gate circuit. I'm sure you've committed to memory the equations for all the gates in this course up to this point. (If not, use the pull-out sheets from study unit 2 as a reference.) Now it's time to put a few gates together and come up with an equation describing that circuit. Let's try the following circuit (fig 4-4).

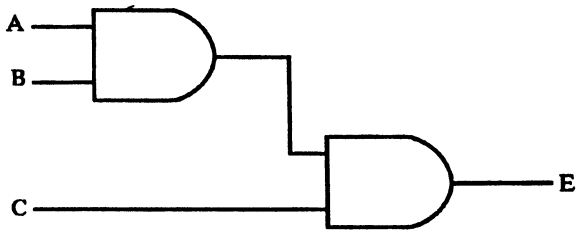


Fig 4-4. Two **AND** gate circuit.

Remember: A gate's equation is an expression of what makes that gate active.

(1) To start with, consider any gate's equation. A logic equation should do three things: describe the active condition, account for the inputs, and express the output in terms of the input. Let's clarify this. Look at the **AND** gate in the first example (figure 4-2). The equation for this gate is A x B = some value. This simply means that when A and B are present the gate is active, that A and B are the gate's inputs, and the output of this gate is the logical product of the "**AND**ing" of A and B. Normally we would have a single letter representing the output (but here we'd lose track of our inputs). In this case the output is also the input to the next stage. So instead of using a single letter to describe this intermediate output we'll use the expression AB.

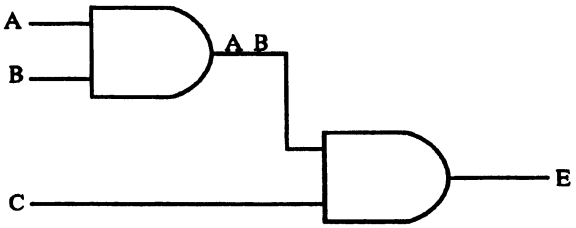


Fig 4-5. Two **AND** gate circuit.

(2) Now for the last gate.

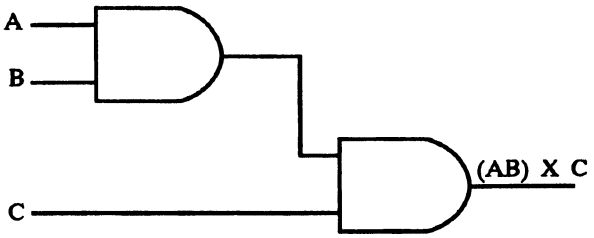


Fig 4-6. Two **AND** gate circuit.

Since the output of the first gate is one input and C is the other, we wind up with an expression which will describe the complete diagram: (AB) x C = E. Notice that the equation:

* Describes the circuit's active condition.
* Accounts for all inputs.
* Expresses the output in terms of the inputs.

b. **AND-OR circuit**. Let's try another diagram.

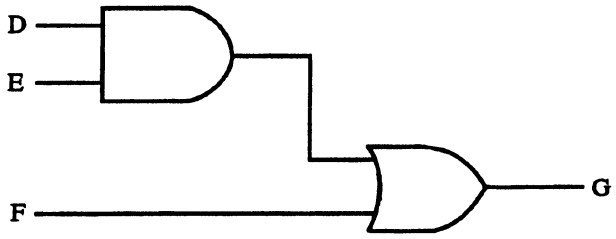


Fig 4-7. **AND-OR** circuit.

(1) The first expression, D x E, is similar to the expression in the last example (figure 4-4).

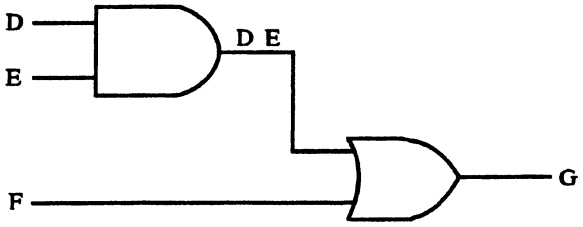


Fig 4-8. **AND-OR** circuit.

(2) Now we **OR** the output of the first gate with F to obtain an expression that describes this diagram: DE + F = G.

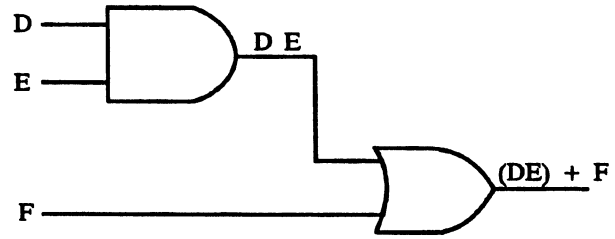


Fig 4-9. **AND-OR** circuit.

c. Three gate circuit. One more logic diagram; one more equation.

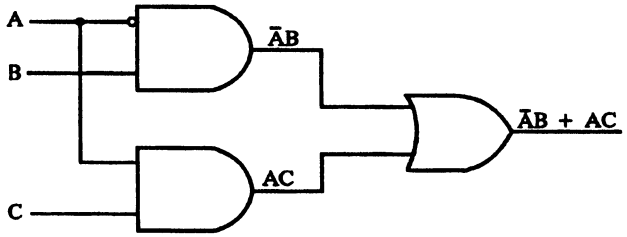


Fig 4-10. Three gate circuit.

At this point, you've learned enough so that there's no need to step through each stage of developing the equation. The outputs of both **AND** gates are **OR**ed together giving us the equation $\bar{A}B + AC$.

4103. Truth Tables

A truth table is also a valuable tool in determining the functioning of a circuit. But contrary to the equation's conciseness, a truth table is expansive. As did the equation, the truth table shows the circuit's active condition, accounts for all inputs, and relates output to input. The truth table goes further in that it lists all possible input/output combinations (in most cases). This listing of values allows you to relate the function of one circuit to another. However, there are times when all values won't be displayed. You'll see these special cases when we analyze flip-flops.

a. Truth table values. The values in a truth table for a circuit under study can often help you relate the circuit under study to a circuit you're already familiar with.

(1) In other words, a truth table can help you answer these questions:

Is this circuit performing the same function as some other device or circuit that I already know?
Could this circuit be replaced by another device or circuit?

Truth tables, as you've already seen, can list values for a circuit as well as an individual gate. Let's develop truth tables for circuits for which we've previously derived equations.

(2) Examples:

A	B	C	AB	E
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	1

The truth table to the left lists all the possible values for the circuit in figure 4-4. The importance of a truth table as an aid to understanding a circuit is apparent from even this relatively straightforward example. Not only does it provide a working combination of values, but it also helps you conceptualize the function of the circuit. Looking at this truth table, I'm sure you can see that the logical function it represents can also be performed by a single gate, that of a ... well, you tell me the gate.

That's right! It's a ...

D	E	F	DE	G
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

...three input **AND** gate.

This second truth table similarly lists all possible values for the circuit in figure 4-7.

(3) The last column of each truth table could easily be replaced by the left side of the circuit's equation. In fact, some texts present truth tables in this fashion. As an example, the top rows for each of our previous truth tables would look like these:

A	B	C	AB	ABC

D	E	F	DE	DE + F

The first truth table shows that A and B are **AND**ed together, and this logical product is then **AND**ed with C to produce the output E (or ABC). The second truth table shows that D and E are **AND**ed together, and this logical product is then "**OR**ed" with F to produce the output G (or DE + F). Of course the values in each column would be the same. Presented in this fashion, the truth table shows the mathematical relationship of input to output. It also allows you to view a circuit in stages without losing track of the various inputs. In other words, if the output that the truth table was developed for was an input to some follow-on stage, then you can clearly see the relationship of the original inputs to the output of the next stage.

(4) Let's look at the truth table for the circuit in figure 4-10.

A	B	C	$\bar{A}B$	AC	$\bar{A}B$ + AC
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	1	1

b. Summary. Again, a truth table is just a tool to help you understand the circuit in question. The format of the table is not as important as the function it serves. As the circuits you are working on become more complex, listing the output in terms of the inputs will certainly be more helpful to you. For this course feel free to use whatever format you are comfortable with.

As we've discovered, truth tables:

* List all (in most cases) possible input/output conditions of the gate.

* Aid in conceptualizing the functioning of a circuit (relates known to unknown).

4104. Timing Diagrams

The last tool of circuit evaluation that we'll look at is a timing diagram. The timing diagram also relates output to input, shows the active condition, and accounts for inputs. The importance of this troubleshooting aid is in how it relates output to input.

a. Overview. A timing diagram is unique for two reasons. The first reason is its ability to visually relate output to input for specific time periods. This aspect is very important as you attempt to troubleshoot more complex circuits. Secondly, a timing diagram can be constructed for any combination of inputs you choose. Naturally the output waveform will be input dependent.

b. Examples. Let's look at a few examples.

(1) The two timing diagrams in figure 4-11 relate to the logic diagrams of figures 4-4 and 4-7.

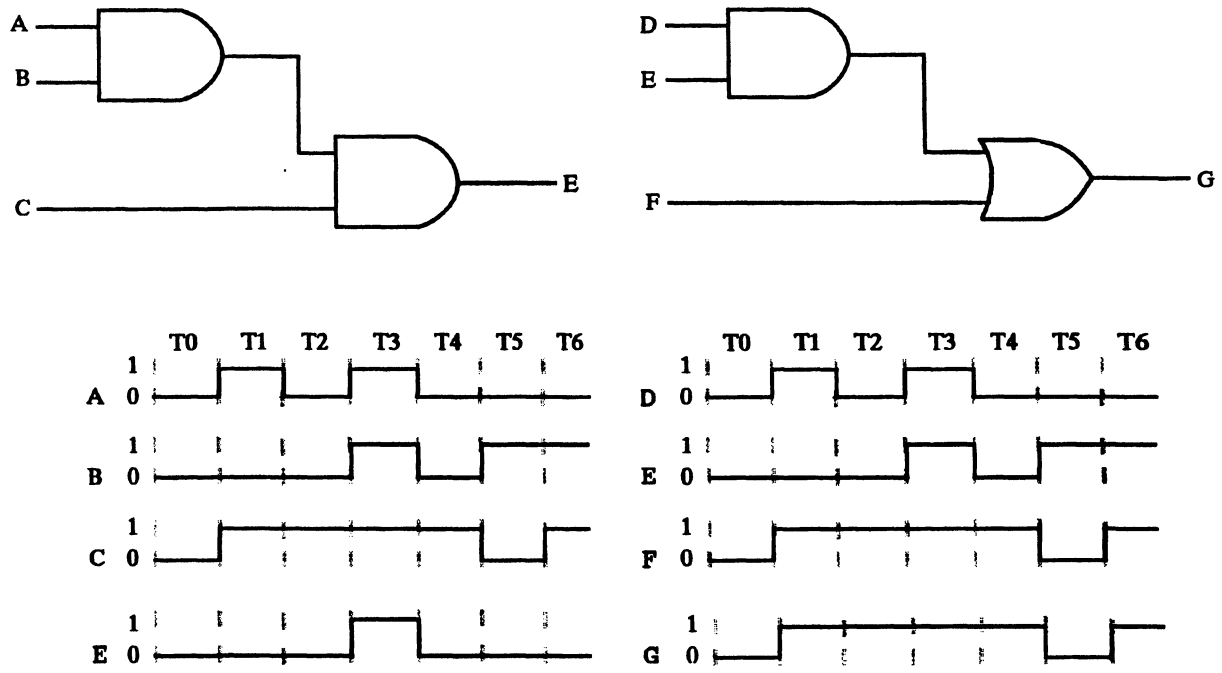


Fig 4-11. Timing diagrams.

Both of these are easy, but they illustrate the function of a timing diagram.

Notice that the timing diagrams:

* Gave a visual representation of the relationship of the input to output waveforms.

(2) The next timing diagram illustrates not only the function of a timing diagram but also its importance in assessing a time sequenced dependent circuit.

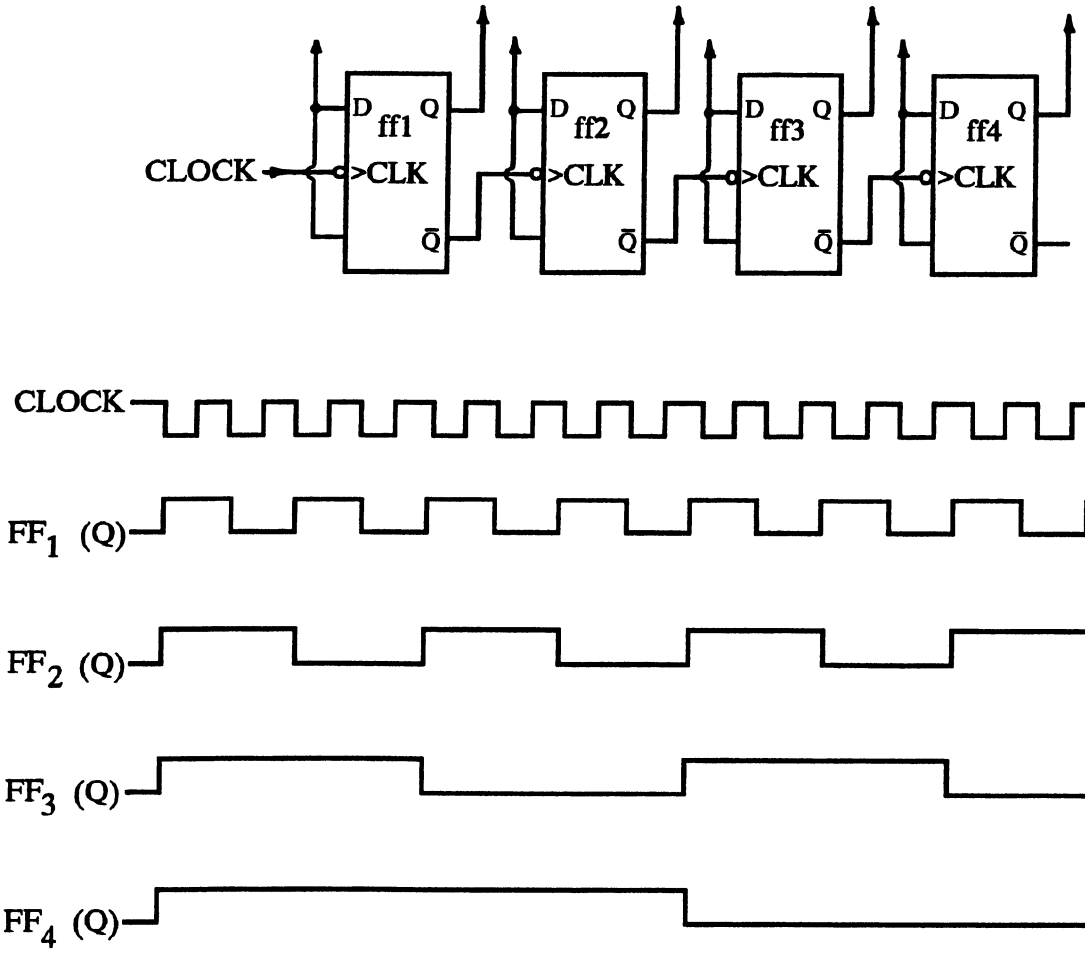


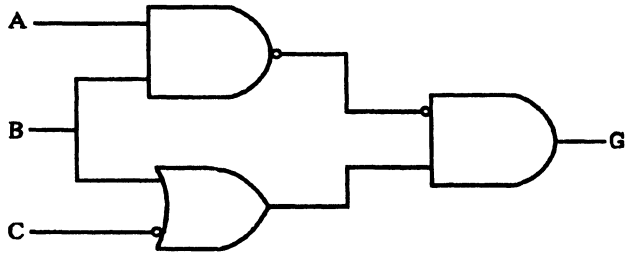
Fig 4-12. Timing diagram.

As you can see from this timing diagram, the clock's importance is clearly highlighted. Only when the clock is changing from high to low (negative edge clocking) will the flip-flop output change. It is this aspect of the timing diagram, the ability to relate output to input for a specific time period, that makes it an important part of the troubleshooting process.

Lesson Summary. You are now equipped to begin the first stage of digital electronic troubleshooting; circuit analysis. This lesson presented the knowledge and tools to assess a digital logic circuit. In the next lesson we will do just that, assess common circuits.

Exercise: Complete items 1 through 4 by performing the action required. Check your responses against those listed at the end of this study unit.

Items 1 and 2 refer to the following diagram. Review the diagram, then answer the items.

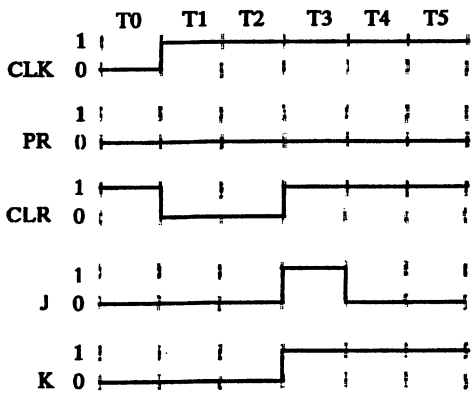


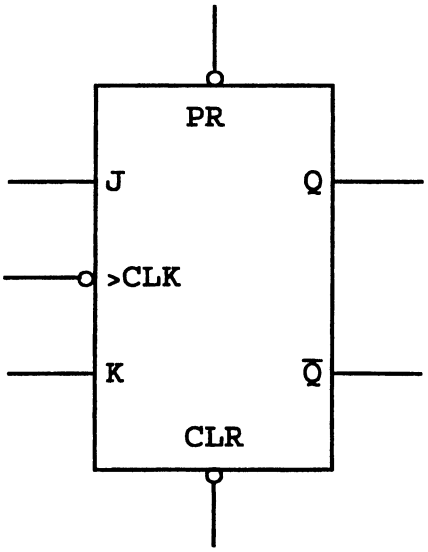
1. In the space provided, construct a truth table describing this logic diagram.

2. In the space provided, write the equation describing this logic diagram.

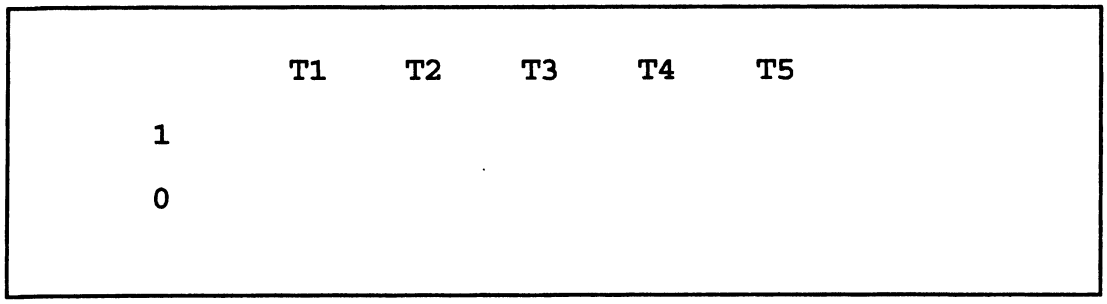
_____.

Item 3 refers to the following logic diagram.

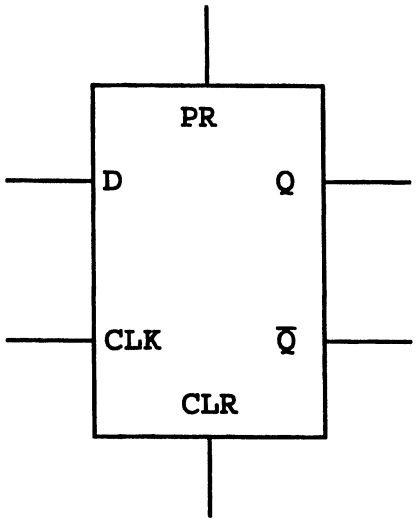




3. In the space provided draw the output waveform (at Q) for time periods T1 through T5.

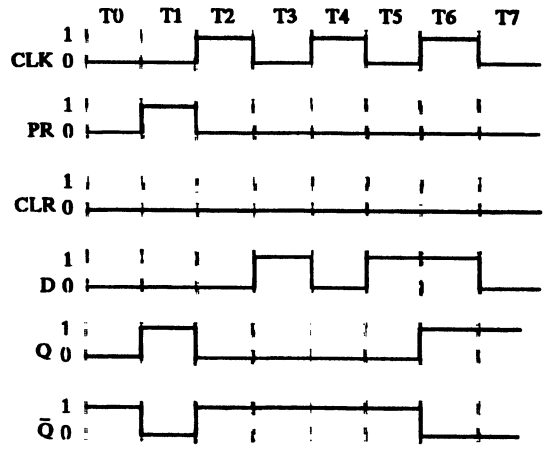


Item 4 refers to the logic diagram below. Review the diagram, then answer the item.

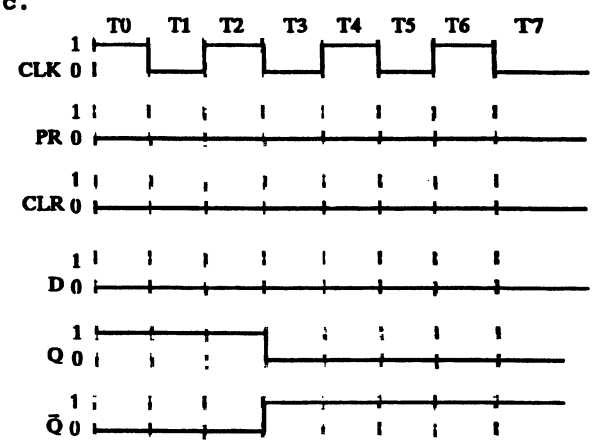


4. Which of the following timing diagrams is consistent with this logic diagram?

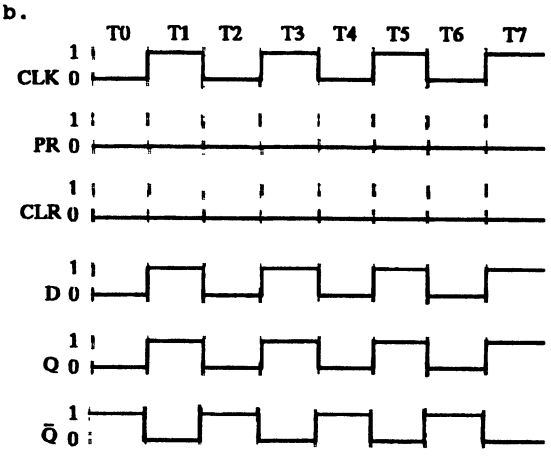
a.



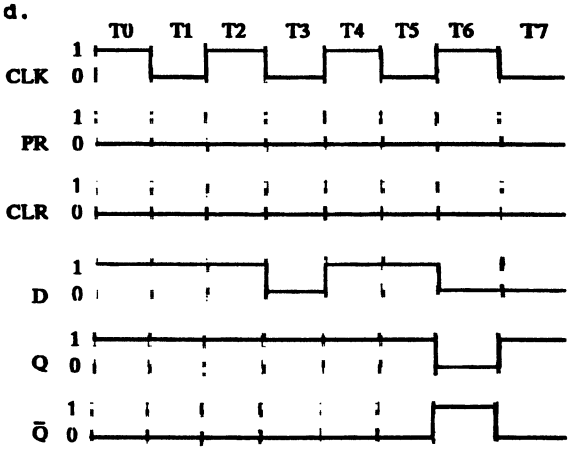
c.



b.



d.



Lesson 2. CIRCUIT EVALUATION

LEARNING OBJECTIVES

1. Determine logic levels throughout a logic circuit when given the input(s).

2. Determine the output of a logic circuit when given the input(s).

4201. Circuit Analysis

Up to this point, in the course, we've learned a number of things:

- The binary numbering system
- Basic logic operations
- How to combine basic logic operations into simple gates and circuits
- The operation of some common circuits
- Truth tables, timing diagrams, and equations with an eye towards using them as tools for circuit analysis

With this information under your belt, you can now assess circuits culled from various T.M.'s. In other words, it's time to look at some circuits you're likely to encounter on the job.

a. Example one. Let's look at the circuit in figure 4-13. The package of schematics this circuit was taken from identifies the circuit as a flip-flop.

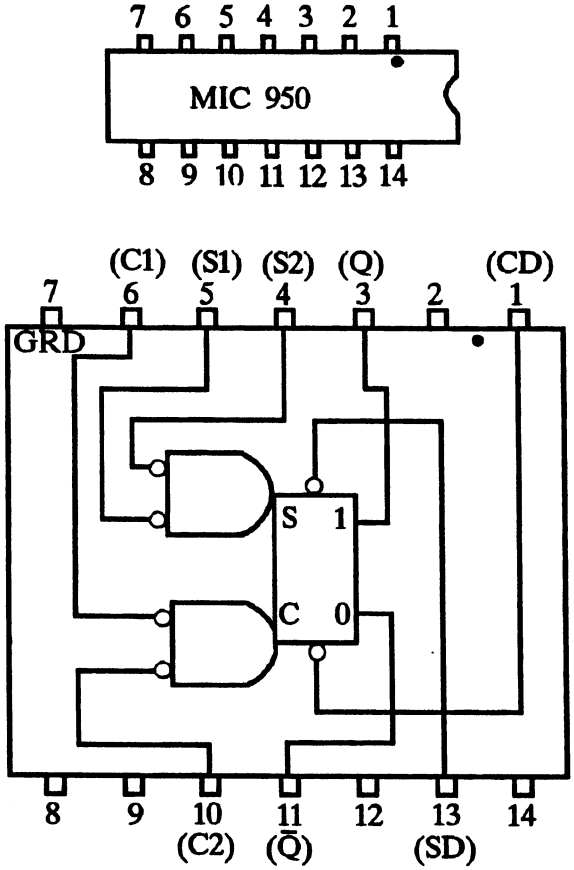


Fig 4-13. Flip-flop.

Is this circuit a flip-flop?

No! This circuit is actually a latch. As you learned before, there is a difference between a latch and a flip-flop. A flip-flop is clocked, a latch is not. As you also learned in study unit 3, latches are often called flip-flops, though technically they are not.

This simple circuit is typical (in form) of circuits encountered on the job. However, there are some minor differences between the way this circuit is configured/labeled and what you have seen up to this point in this course:

* The inhibited **AND** gates don't appear to have outputs.
* The inhibited **AND** gates' outputs are actually the circuit's data inputs.
* The circuit's data inputs are labeled S (set) and C (clear) instead of S (set) and R (reset).
* Preset and clear are not labeled on the latch; however, the input lines to these functions are labeled SD (preset) and CD (clear).
* On the latch Q and $\overline{Q}$ are labeled 1 (Q) and 0 ($\overline{Q}$).

(1) Truth table. Now let's look at the truth table. If you glanced quickly at the truth table, you might think that it is missing the disallowed condition. But is it? Let's examine this truth table in more depth keeping in mind that the inhibited **AND** gates' outputs are the data inputs for this flip-flop. We'll begin by reviewing the truth table from the schematic.

S1	S2	C1	C2	Q
0	0	1	0■1	1
0	0	0■1	1	1
1	0■1	0	0	0
0■1	1	0	0	0

The original truth table resembles this one on the left, which can be thought of as the two truth tables below.

S1	S2	C1	C2	Q
0	0	1	0	1
0	0	0	1	1
1	0	0	0	0
0	1	0	0	0

S1	S2	C1	C2	Q
0	0	1	1	1
0	0	0	1	1
1	1	0	0	0
1	1	0	0	0

First we know that the equations for each gate are $\overline{S1} \times \overline{S2} = S$ and $\overline{C1} \times \overline{C2} = C$. This means that for each gate to be active, the inputs to that gate must both be zero. Knowing this we can simplify the original truth table. Let's construct a truth table in terms of S and C, the inputs to the flip-flop, using the values from our original truth table.

S	C	Q
1	0	1
1	0	1
0	1	0
0	1	0

The values in the truth table to the left were derived by solving the equations for the inhibited **AND** gates. We've simply **AND**ed $\overline{S1}$ and $\overline{S2}$, using the values from the truth table above to arrive at the values for S. The values of C are derived from **AND**ing $\overline{C1}$ and $\overline{C2}$ together. Now what we have is a simpler truth table; in fact, it can be simplified even further to yield the truth table to the right.

S	C	Q
1	0	1
0	1	0

Now back to the question of the disallowed condition in the original truth table. The input values that would lead to the disallowed condition are not shown.

> What are the values that would lead to the disallowed condition?

> Lows on S1, S2, C1, and C2, which would make S and C both high.

(2) Timing diagram. Before you can troubleshoot, you must understand the circuit. In other words, to determine what should be on the output you must know how each input affects a circuit. In the analysis of this circuit so far we've done pretty much just that. There is an addition that would make the analysis clearer still -- a timing diagram. Figure 4-14 depicts a timing diagram showing possible input and output conditions for the circuit shown in figure 4-13. Note that there is no attempt to portray the disallowed condition.

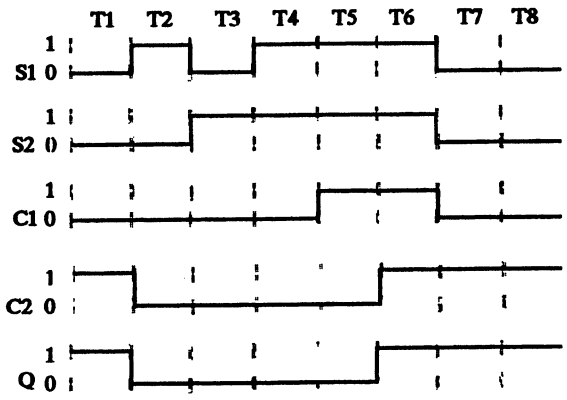


Fig 4-14. Timing diagram.

The blending of equation, truth table, and timing diagram allowed us to see this circuit in a clear logical fashion. It isn't necessary to apply the tools in this sequence; nor is it necessary to use all three tools to make heads or tails of a circuit. The secret is to use whatever tool(s) necessary and in any sequence that allows YOU to understand the system you're working on. For a simple circuit a truth table may suffice, for a more complex circuit you may have to develop a timing diagram as well to a fully understand the circuit's function. If you can determine the state of a circuit at any given time then you've successfully completed the first step of the troubleshooting process. In the next example we'll use a timing diagram and our basic electronic knowledge to evaluate a circuit.

b. Example two. The value of a timing diagram really comes forward when you try to determine the output of this eight bit shift register.

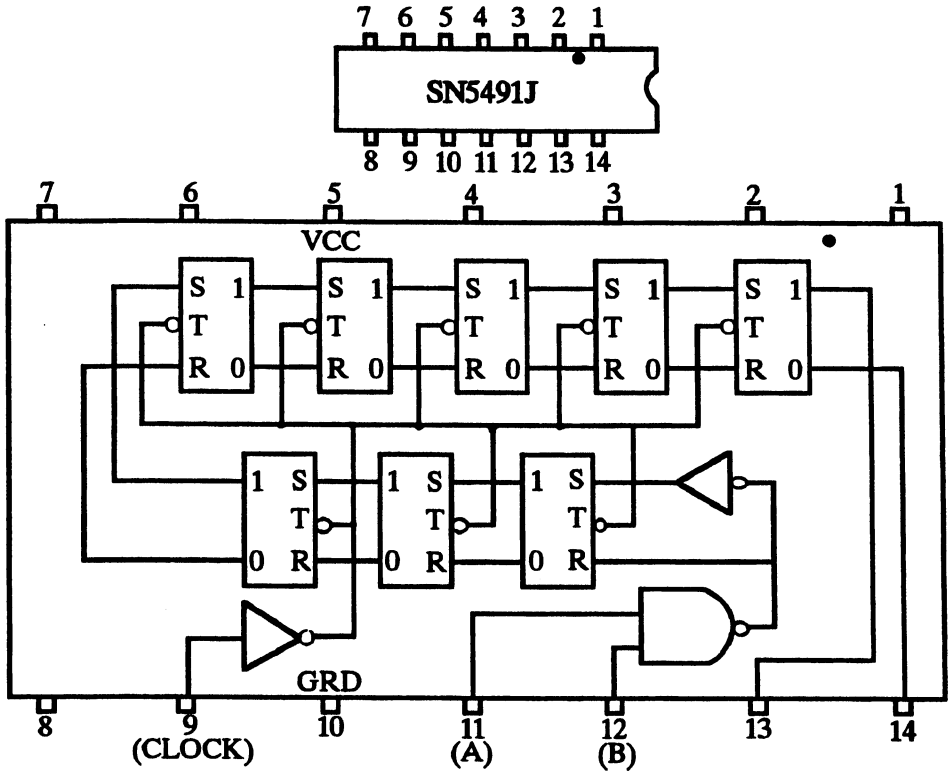


Fig 4-15. Shift register.

(1) Let's step through the operation of this register and build a timing diagram to get a picture of its operation. But first, some questions.

What is the purpose of a register?

What logic level must be on pin 9 to effect a change on the flip-flops?

On what pin(s) can you detect the output of this register?

Can we drive the flip-flops into the disallowed condition?

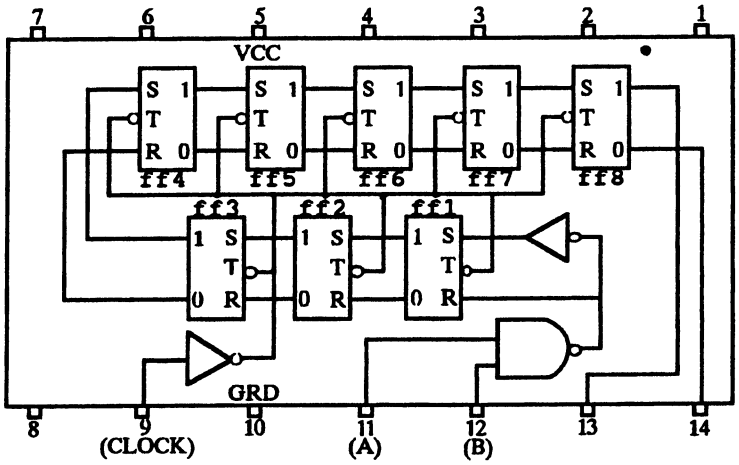
The purpose of a register is to temporarily store data, to act as a buffer, to generate sequences, to divide, and to convert information from one format to another.

A high must be on pin 9 to effect a change on the flip-flops.

The outputs can be detected on pins 13 and 14. 13(Q) and 14($\overline{Q}$)

The invertor on the Set line assures that Set and Reset won't be the same logic level.

(2) With these questions out of the way, let's now analyze our register with the help of a timing diagram. Also we'll number the flip-flops to make them easier to identify.



Note: The outputs of these flip-flops are labeled 1 and 0, instead of Q and $\overline{Q}$.

You've already learned the basic operation of a register, so this step through should be easy for you.

How can we store a one on ff1?

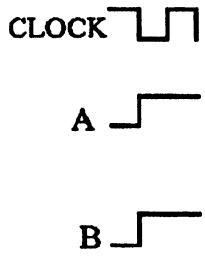
If you answered...

There's only one way to store a one on ff1; a high on both pins 11(A) and 12(B).

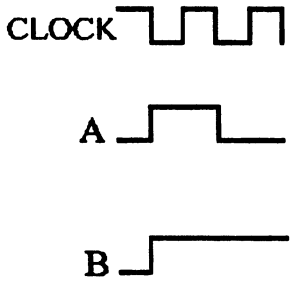
WRONG!!

There is only one way to store a one on ff1;
a high on both pins 11(A) and 12(B), **and**
a high on pin 9(clock).

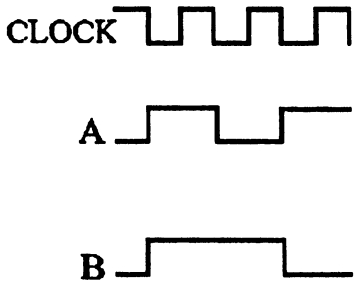
Let's do this!!!



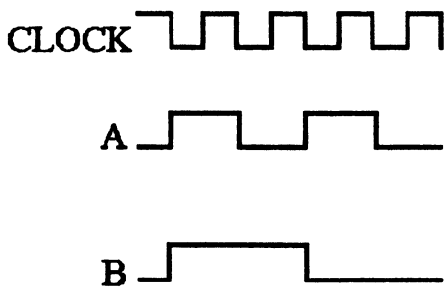
Now, let's input a low on 11 and a high on 12.



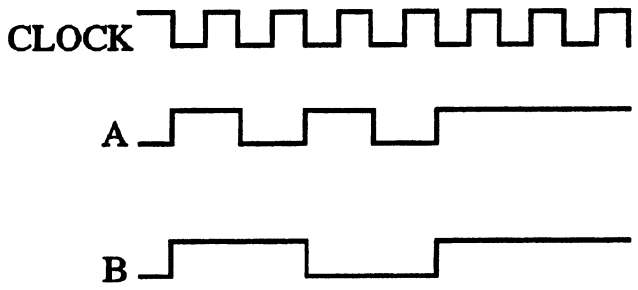
Now, a one on 11 and a zero on 12.



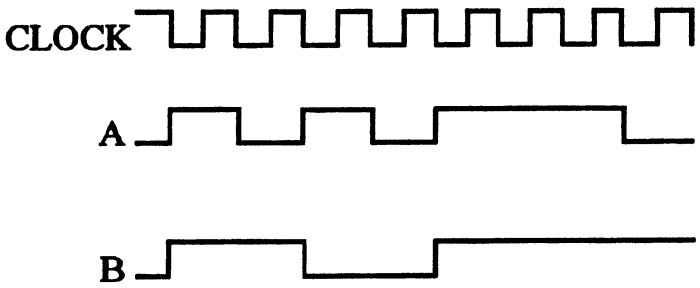
Now, lows on both inputs.



Let's try highs on both pins as our next three inputs.



Now, finally a zero on A and a one on B.



These timing diagrams showed only those logic levels on the inputs, A and B. But what about inside the register itself? What is happening to the various flip-flops?

The first inputs along with a high on the clock set ff1; that is, it is storing a one (Q=1). The second set of inputs along with a high on the clock puts a low on ff1, but since the clock is felt at all flip-flops simultaneously, the high that was stored in ff1 is now passed to ff2. The next set of input pulses continues this process of passing data from one flip-flop to the next. Ff3 is now storing the high; ff1 and ff2 are now both storing a low.

If we assume that all flip-flops were originally at zero, we can create a timing diagram reflecting the logic levels for the clock, the inputs, and each flip-flop as well (fig 4-16).

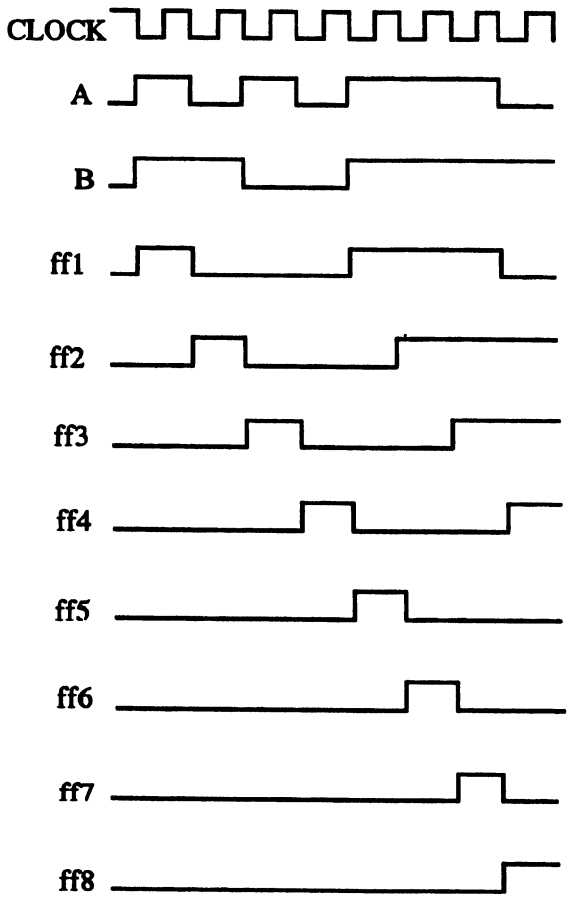


Fig 4-16. Timing diagram.

Notice that in all cases the transfer of data occurred only when the clock was high.

(3) The only thing left to consider is the output. Our next diagram (fig 4-17) adds pin 13(Q) to our other pulses.

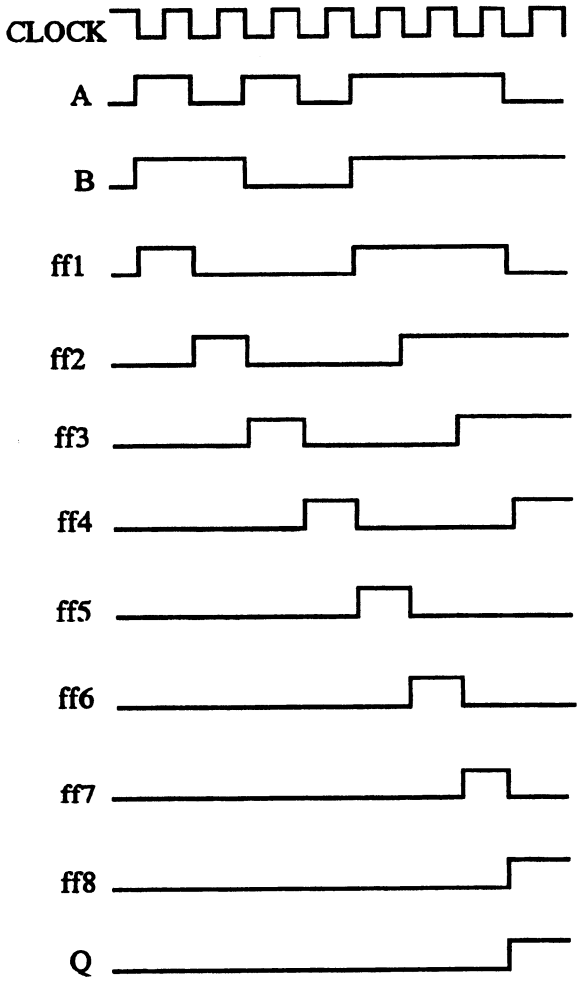


Fig 4-17. Timing diagram.

Notice that there is no output until the eighth data bit is received.

What does the wave form look like for pin 14 at this point?

Draw your answer in this box. ↓

This is the truth table for our register.

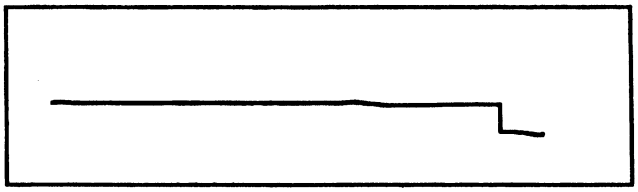
TN		TN+8
A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1

TN - Bit time before clock pulse.

TN+8 - Bit time after eight clock pulses.

As you can see from the table, all of our work has been handily summarized. The truth table tells us two things about our circuit. One, the input has no effect until after eight clock pulses. Two, only a high on both A and B can produce a high on Q, the output.

Oh, by the way, the pulse train you drew should look similar to this one. Naturally, it is just the opposite of the waveform for pin 13.



4202. Review

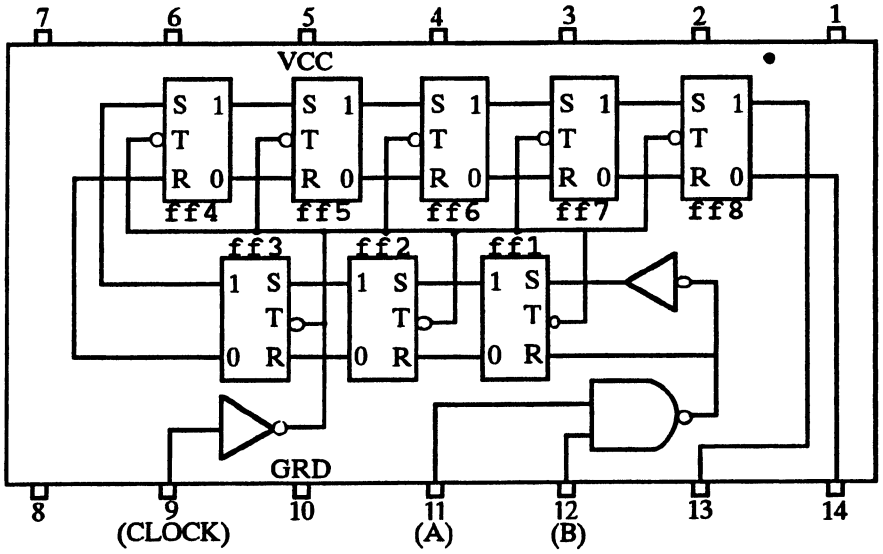
The repair of digital equipment begins in your*, the repairman's, head. When confronted with a malfunctioning piece of equipment, you compare its normal operation with the faulty ones at hand. In trying to logically localize the problem area, you'll call on the tools and knowledge you've learned in this course to help you pinpoint the fault.

This course is only the beginning of the body of knowledge needed to become a successful repairman. But it is designed to serve as a starting point for your future studies.

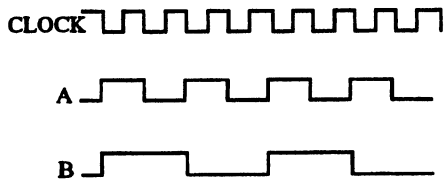
Lesson Summary. This lesson showed you different ways to assess digital circuits. You learned to use the "tools" of digital troubleshooting in your assessment of these circuits. This is the first step on the road of digital troubleshooting.

Exercise: Complete items 1 through 4 by performing the action required. Check your responses against those listed at the end of this study unit.

For items 1 through 4 refer to the following diagram. Review the diagram, then answer the items.

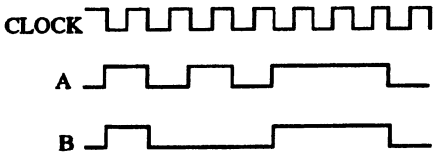


1. In the space provided, draw the waveform generated at ff2 (Q) given these input conditions.



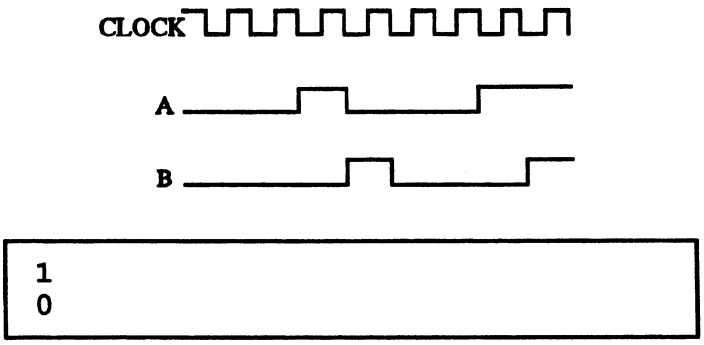
1
0

2. In the space provided, draw the waveform generated at ff5 (Q) given these input conditions.

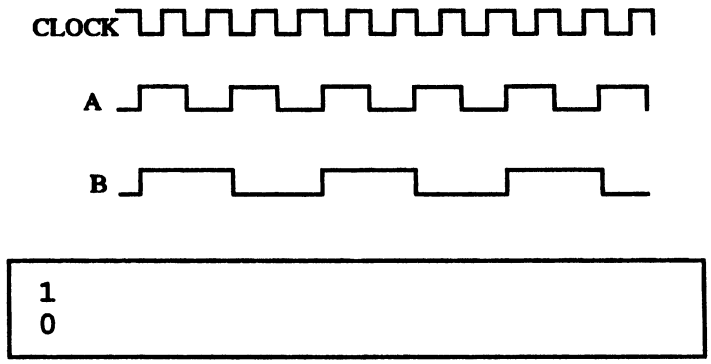


1
0

3. In the space provided, draw the waveform generated at pin 13 given these input conditions.



4. In the space provided, draw the waveform generated at ff7 given these input conditions.



UNIT SUMMARY

This study unit taught you the value of timing diagrams, equations, and truth tables in evaluating digital circuits. These are important first steps on the road to becoming a master repairman.

Lesson 1 Exercise Solutions

Reference

1. 4103

A	B	C	$\overline{AB}$	$B+\overline{C}$	$AB(B+\overline{C})$
0	0	0	1	1	0
0	0	1	1	0	0
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	0	1	1
1	1	1	0	1	1

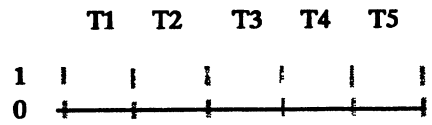
or

1.

A	B	C	$\overline{AB}$	$B+\overline{C}$	G
0	0	0	1	1	0
0	0	1	1	0	0
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	0	1	1
1	1	1	0	1	1

2. $AB(B+\overline{C}) = G$ 4102

3. 4104



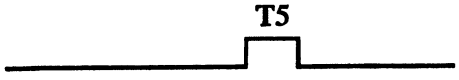
4. b. 4104

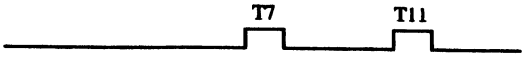
Lesson 2 Exercise Solutions

Reference

1. 4201

2. 4201



	Reference
3.	4201
4. 	4201

FUNDAMENTALS OF DIGITAL LOGIC

REVIEW LESSON EXAMINATION

INSTRUCTIONS: This review lesson is designed to aid you in preparing for your final examination. You should try to complete this lesson without the aid of reference materials, but if you do not know an answer, look it up and remember what it is. The enclosed answer sheet must be filled out according to the instructions on its reverse side and mailed to MCI using the envelope provided. The items you miss will be listed with references on a feedback sheet (MCI R-69) which will be mailed to your commanding officer with your final examination. You should study the reference material for the items you missed before taking the final examination.

Select the ONE answer which BEST completes the statement or answers the item. After the corresponding number on the answer sheet, blacken the appropriate circle.

1. This number, 752_8 may be read as seven five two base

   a. two.
   b. eight.
   c. nine.
   d. ten.

2. How many digits are used in the base two numbering system?

   a. Two
   b. Three
   c. Four
   d. Six

3. What is the value of the highlighted digit in this number, 10101_2?

   a. Zero
   b. Two
   c. Four
   d. Eight

4. What is the place value of the highlighted digit's position in this number, 10101_2?

   a. Zero
   b. Two
   c. Four
   d. Eight

5. What is the decimal equivalent of 100.1_2 ?

   a. 13.5
   b. 10.5
   c. 8.1
   d. 4.5

6. What is the decimal equivalent of 10101.1_2 ?

   a. 21.5
   b. 13.1
   c. 9.5
   d. 6.25

7. What is the binary equivalent of 30.75?

a. 1111.11_2
b. 11001.011_2
c. 11110.11_2
d. 11111.11_2

8. What is the binary equivalent of 19.875?

a. 1111.101_2
b. 10011.011_2
c. 10011.111_2
d. 11111.111_2

9. The answers for the following binary multiplication problems in order are

0 x 0
1 x 0
0 x 1
1 x 1

a. 0, 0, 0, 1.
b. 0, 1, 0, 1.
c. 1, 0, 0, 1.
d. 1, 1, 1, 0.

10. The answers for the following binary subtraction problems in order are

0 - 0
1 - 0
1 - 1
0 - 1

a. 0, 0, 0, 1.
b. 0, 0, 0, 1.
c. 0, 1, 0, 1 borrow 1.
d. 0, 1, 1 borrow 1, 1.

11. The answers for the following binary addition problems in order are

0 + 0
0 + 1
1 + 0
1 + 1

a. 0, 1, 1, 1 carry 1.
b. 0, 1, 1, 0 carry 1.
c. 0, 0, 1, 0 carry 1.
d. 0, 0, 0, 1 carry 1.

12. On, high, active, and one are associated with a

a. false.
b. true.
c. logic gate.
d. zero volt condition.

13. Off, low, inactive, and zero are associated with a

a. false.
b. logic gate.
c. true.
d. clocked gate.

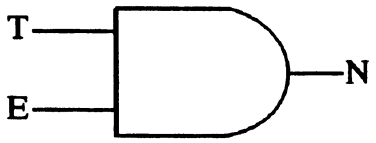
14. The logic state of a gate is determined by its _____ condition.

a. input
b. voltage
c. output
d. static

15. **AND**ing, **OR**ing, and complementation are the

a. three basic logic gates.
b. three basic truth tables.
c. three basic logic symbols.
d. three basic logic operations.

For items 16 and 17 refer to the following gate. Examine the logic gate, then answer the items.



16. Which truth table describes this gate?

a.

T	E	N
0	0	0
0	1	0
1	0	0
1	1	1

b.

T	E	N
0	0	1
0	1	0
1	0	0
1	1	1

c.

T	E	N
0	0	0
0	1	1
1	0	1
1	1	1

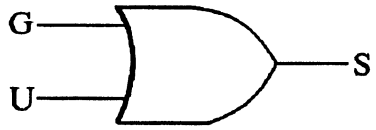
d.

T	E	N
0	0	0
0	1	1
1	0	1
1	1	0

17. Which equation describes this gate?

a. $T \times E = N$
b. $\overline{T} \times E = N$
c. $T + E = N$
d. $T + \overline{E} = N$

For items 18 and 19 refer to the following gate. Review the gate, then answer the items.



18. Which truth table describes this gate?

a.

G	U	S
0	0	0
0	1	0
1	0	0
1	1	1

b.

G	U	S
0	0	1
0	1	0
1	0	0
1	1	1

c.

G	U	S
0	0	0
0	1	1
1	0	1
1	1	1

d.

G	U	S
0	0	0
0	1	1
1	0	1
1	1	0

19. Which equation describes this gate?

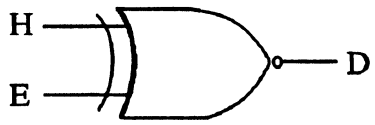
a. $G \times U = S$

b. $\overline{G} \times U = S$

c. $G + U = S$

d. $G + \overline{U} = S$

For items 20 and 21 refer to the following gate. Review the logic gate, then answer the items.



20. Which truth table describes this gate?

a.

H	E	D
0	0	0
0	1	0
1	0	0
1	1	1

b.

H	E	D
0	0	1
0	1	0
1	0	0
1	1	1

c.

H	E	D
0	0	0
0	1	1
1	0	1
1	1	1

d.

H	E	D
0	0	0
0	1	1
1	0	1
1	1	0

21. Which equation describes this gate?

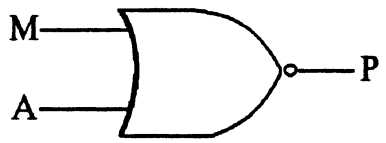
a. $(\overline{H} \times E) + (H \times \overline{E}) = D$

b. $\overline{(H \times \overline{E})} + (H \times E) = D$

c. $(\overline{A} + B) \times (A + \overline{B}) = D$

d. $\overline{(H + \overline{E})} \times (H + E) = D$

For items 22 and 23 refer to the following gate. Review the gate, then answer the items.



22. Which truth table describes this gate?

a.

M	A	P
0	0	1
0	1	0
1	0	0
1	1	0

b.

M	A	P
0	0	1
0	1	0
1	0	0
1	1	1

c.

M	A	P
0	0	0
0	1	1
1	0	1
1	1	1

d.

M	A	P
0	0	0
0	1	1
1	0	1
1	1	0

23. Which equation describes this gate?

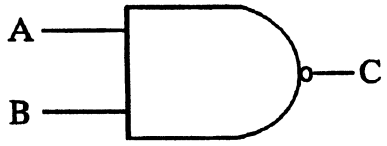
a. $\overline{M} \times \overline{A} = P$

b. $\overline{M \times A} = P$

c. $\overline{M} + \overline{A} = P$

d. $\overline{M + A} = P$

For items 24 and 25 refer to the following gate. Review the logic gate, then answer the items.



24. Which truth table describes this gate?

a.

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

b.

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

c.

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

d.

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

25. Which equation describes this gate?

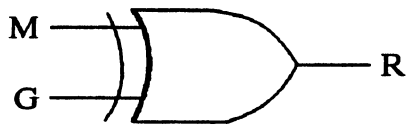
a. $\overline{A} \times \overline{B} = C$

b. $\overline{A \times B} = C$

c. $\overline{A} + \overline{B} = C$

d. $\overline{A + B} = C$

For items 26 and 27 refer to the following gate. Review the gate, then answer the items.



26. Which truth table describes this gate?

a.

M	G	R
0	0	0
0	1	0
1	0	0
1	1	1

b.

M	G	R
0	0	1
0	1	0
1	0	0
1	1	1

c.

M	G	R
0	0	0
0	1	1
1	0	1
1	1	1

d.

M	G	R
0	0	0
0	1	1
1	0	1
1	1	0

27. Which equation describes this gate?

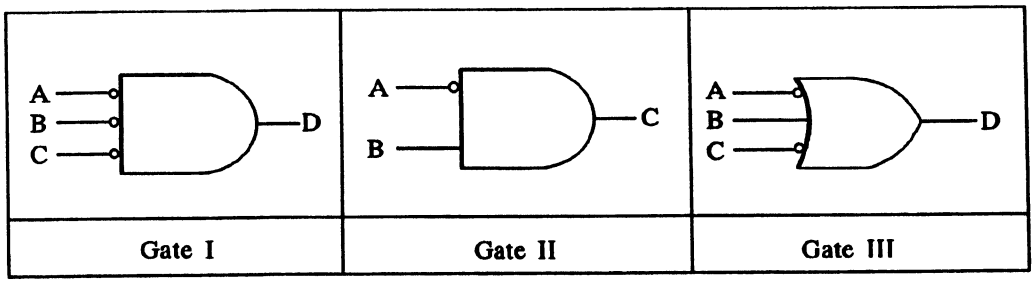
a. $(\overline{M} \times G) + (M \times \overline{G}) = R$

b. $\overline{(M \times G)} + (M \times G) = R$

c. $(\overline{M} + G) \times (M + \overline{G}) = R$

d. $\overline{(M + G)} \times (M + G) = R$

For items 28 through 31 refer to the chart of logic gates below. Review the gates, then answer the items.



28. Which truth table describes gate I?

a.

A	B	C	D
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	1

b.

A	B	C	D
0	0	0	0
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	1

c.

A	B	C	D
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	0

d.

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

29. What is the equation for gate I?

a. $\overline{A} \times \overline{B} \times \overline{C} = D$

b. $\overline{ABC} = D$

c. $\overline{A} + \overline{B} + \overline{C} = D$

d. $\overline{A + B + C} = D$

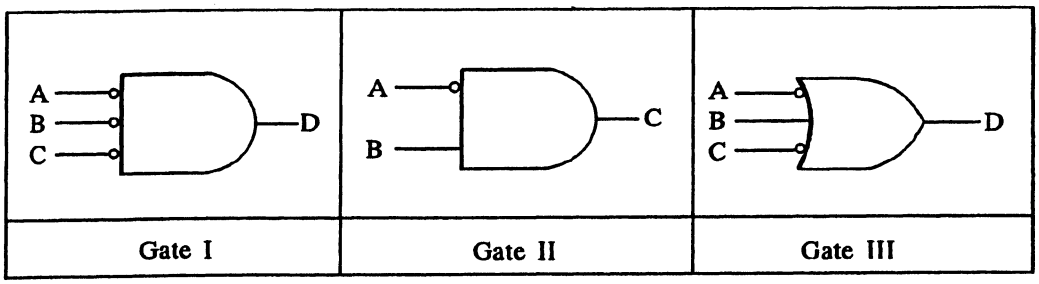
30. What is the equation for gate II?

a. $\overline{A} + B = C$

b. $\overline{A} + B = \overline{C}$

c. $\overline{A} \times B = C$

d. $\overline{A} \times B = \overline{C}$



31. Which is the truth table for gate III?

a.

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

b.

A	B	C	D
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	0

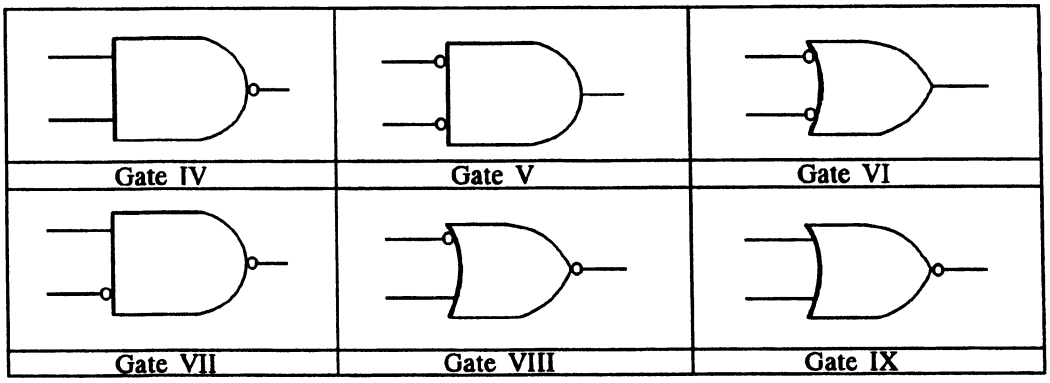
c.

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

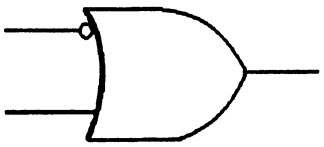
d.

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

For items 32 and 33 refer to the chart of logic gates below. Review the gates, then answer the items.

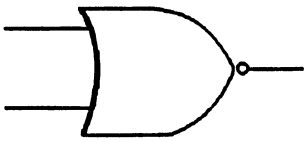


32. Which gate(s) in the chart has(have) the same truth table as the following gate?



a. IV
b. VI
c. V
d. VII

33. Which gate(s) in the chart has(have) the same truth table as the following gate?



a. V
b. VII
c. VIII
d. IX

34. Which statement best describes "triggering"?

a. The act of changing a digital electronic device from one state to another
b. A signal that determines when a device will change from one condition to another
c. A description of waveforms inputs and outputs to a digital electronic device
d. That part of the output signal that corresponds to the input signal

35. Clocking can best be described as

a. a signal that determines when a device will change from one condition to another.
b. the act of changing a digital device from one condition to another.
c. a description of the cycle of operation for a digital device.
d. a signal that monitors a digital device or circuit.

36. A bi-stable device can best be described as a digital electronic device

a. with two or more stable states.
b. with two stable states; true or high.
c. that has two stable states, with the output remaining stable until the device receives an input to change states.
d. that has two or more stable states, with the output remaining constant during each state.

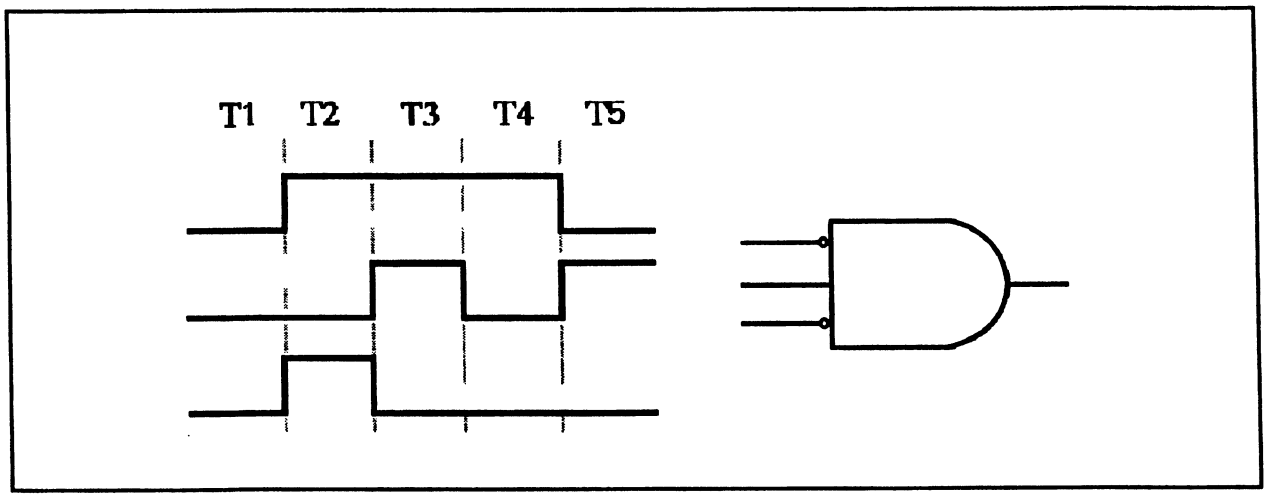
37. Race can best be described as a condition

a. where the output of a bi-stable device can not be predicted because the device has been driven into the disallowed state.
b. where a flip-flop changes states too rapidly.
c. where a bi-stable device fails to maintain its memory.
d. where the output of a bi-stable device locks at zero because the device has been driven into the disallowed state.

38. Preset and clear, in relation to bi-stable devices, can best be described as

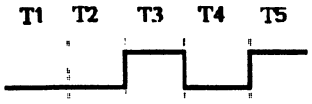
a. signals that allow a flip-flop to be set or reset without a signal on either the clock or data input lines.
b. the two states of a flip-flop or latch.
c. a condition where $Q = 0$.
d. a condition where $\overline{Q} = 0$.

For item 39, refer to the following diagram.

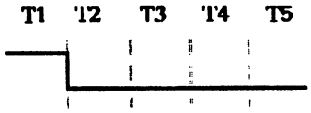


39. Given this gate and these input waveforms, select the proper output waveform.

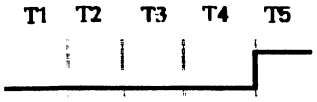
a.



c.

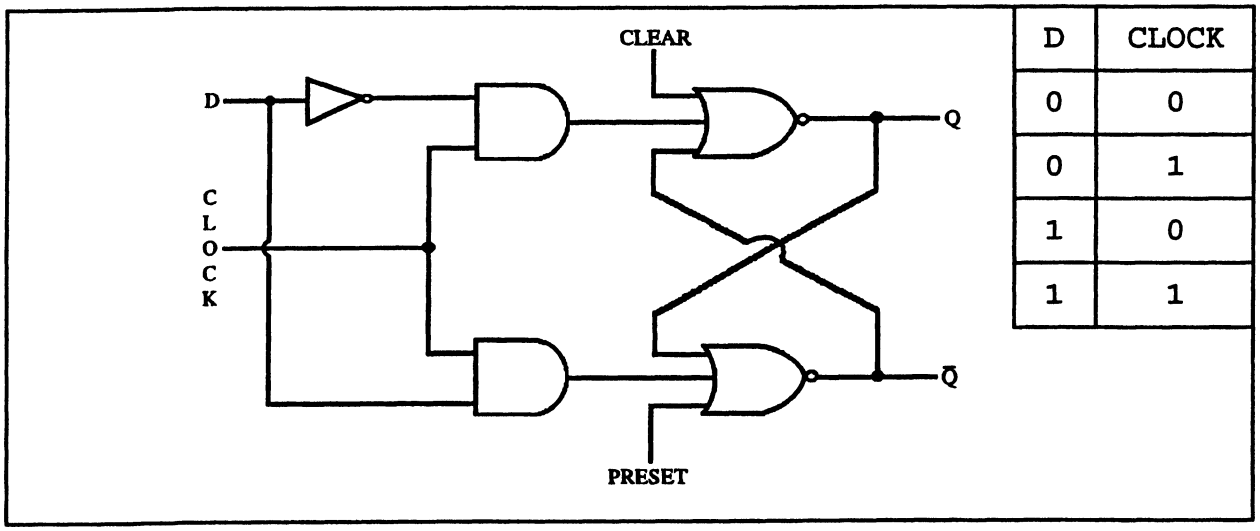


b.



d.

Refer to the following figure for item 40.



D	CLOCK
0	0
0	1
1	0
1	1

40. Select the proper values for Q and not $\overline{Q}$.

a.

Q	$\overline{Q}$
no change	
0	1
no change	
1	0

c.

Q	$\overline{Q}$
1	0
0	1
1	0
0	1

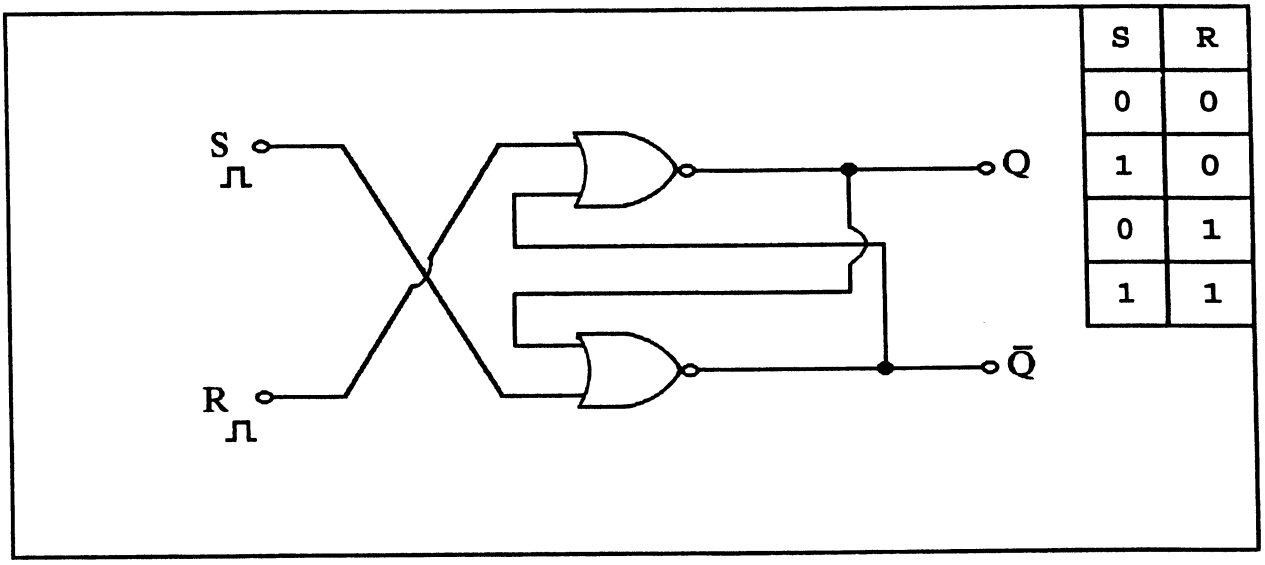
b.

Q	$\overline{Q}$
no change	
0	1
1	0
disallowed state	

d.

Q	$\overline{Q}$
no change	
1	0
no change	
0	1

Refer to this figure for item 41.



S	R
0	0
1	0
0	1
1	1

41. Select the proper values for Q and not $\overline{Q}$.

a.

Q	$\overline{Q}$
no change	
1	0
0	1
disallowed state (both 0's)	

c.

Q	$\overline{Q}$
no change	
0	1
1	0
disallowed state (both 1's)	

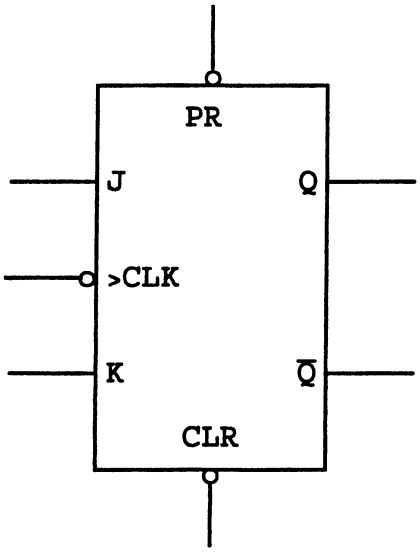
b.

Q	$\overline{Q}$
disallowed state	
1	0
0	1
no change	

d.

Q	$\overline{Q}$
no change	
0	1
1	0
no change	

42. What statement is true about this logic symbol?



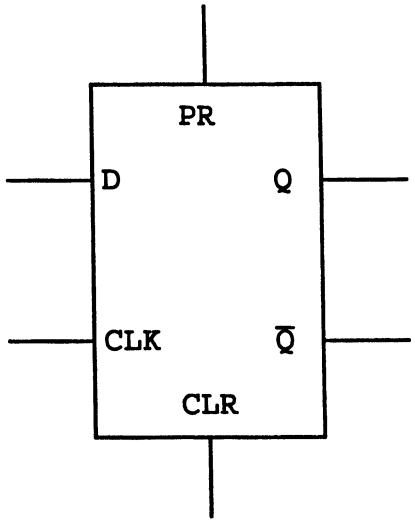
a. The J and K inputs are high level triggered, the clock is negative edge triggered, and the preset and clear are negative edge triggered.

b. The J and K inputs are high level triggered, the clock is negative edge triggered, and the preset and clear are low level triggered.

c. In order to preset or clear, there must be a high on the clock line.

d. In order to preset or clear, there must be a low on the clock line.

43. What statement is true about this logic symbol?



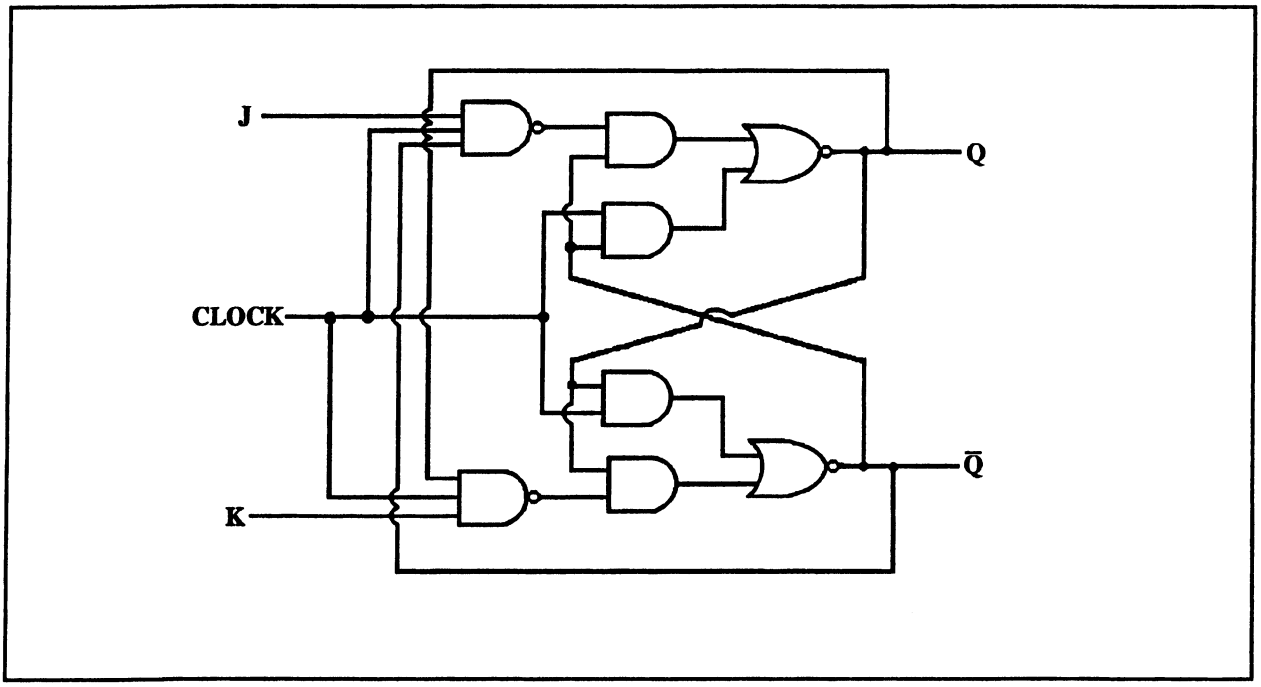
a. The disallowed state is initiated by lows on the input lines.

b. The disallowed state is initiated by highs on the input lines.

c. The only time there is a change in the output is when the clock is 0.

d. There is no disallowed state because one input is inverted.

Refer to this figure for item 44.



44. Which truth table is derived from this flip-flop?

a.

J	K	OUTPUT Q
0	0	no change
1	0	0
0	1	1
1	1	changes to opposite

c.

J	K	OUTPUT Q
0	0	changes to opposite
1	0	0
0	1	1
1	1	no change

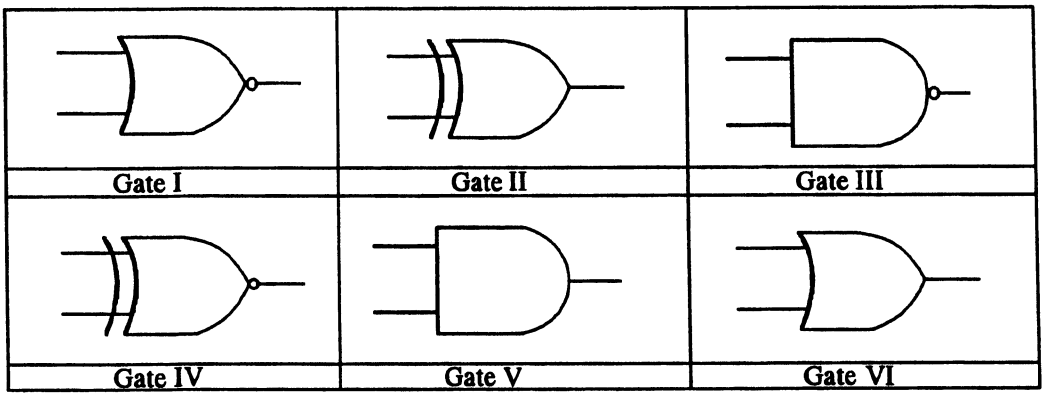
b.

J	K	OUTPUT Q
0	0	no change
1	0	1
0	1	0
1	1	changes to opposite

d.

J	K	OUTPUT Q
0	0	changes to opposite
1	0	1
0	1	0
1	1	no change

For items 45 through 50 refer to the chart that follows. Review the chart, then answer the items.



45. Which schematic symbol is associated with the **NAND** function?

a. II
b. III
c. IV
d. VI

46. Which schematic symbol is associated with the **NOR** function?

a. I
b. IV
c. V
d. VI

47. Which schematic symbol is associated with the **XNOR** function?

a. II
b. III
c. IV
d. V

48. Which schematic symbol is associated with the **AND** function?

a. I
b. II
c. IV
d. V

49. Which schematic symbol is associated with the **OR** function?

a. III
b. IV
c. V
d. VI

50. Which schematic symbol is associated with the **XOR** function?

a. I
b. II
c. IV
d. VI

51. Which algebraic symbol is associated with the **XOR** function?

a. +
b. x
c. ⊕
d. +

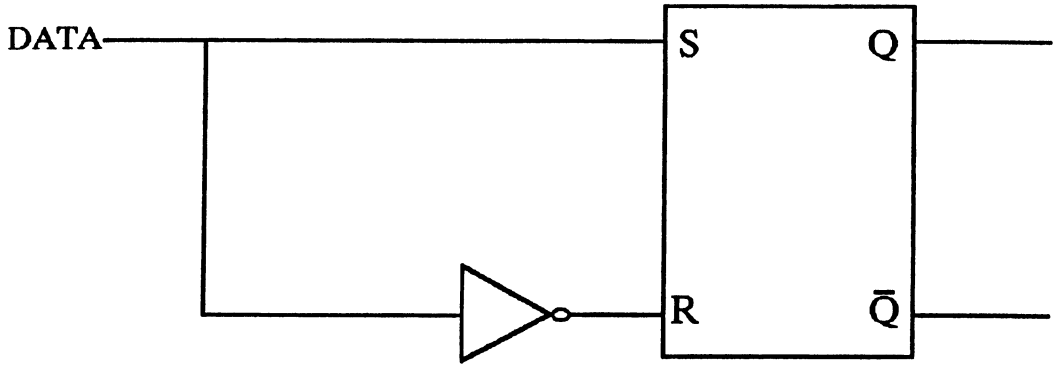
52. Which algebraic symbol is associated with the **OR** function?

a. +
b. x
c. ⊕
d. +

53. Which algebraic symbol is associated with the **AND** function?

a. +
b. x
c. ⊕
d. +

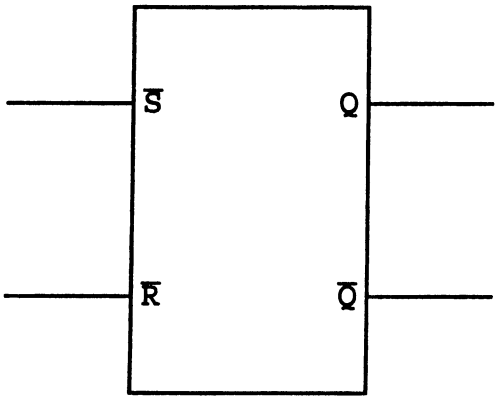
Refer to the following figure for item 54.



54. This is the logic symbol for a(n)

a. RS latch.
b. RS flip-flop.
c. D latch.
d. D flip-flop.

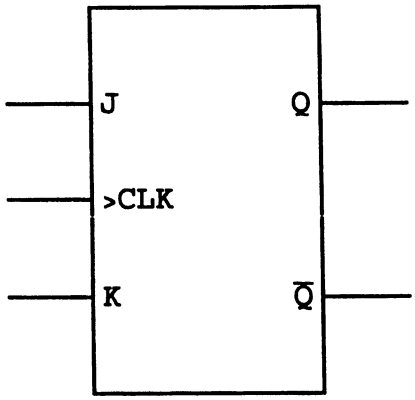
Refer to the following figure for item 55.



55. This is the logic symbol for a(n)

a. RS latch.
b. RS flip-flop.
c. D latch.
d. D flip-flop.

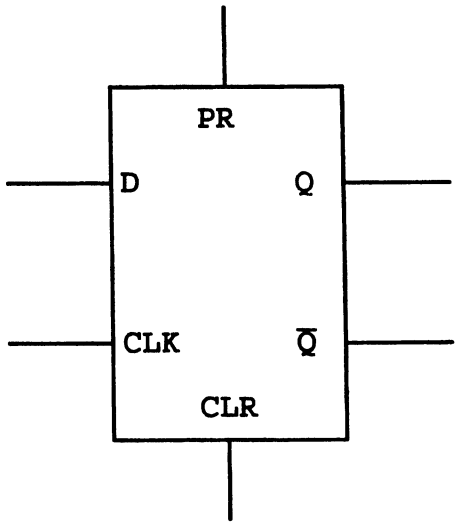
Refer to the following figure for item 56.



56. This is the logic symbol for a(n)

a. JK latch.
b. edge triggered JK flip-flop.
c. level triggered JK latch.
d. level triggered JK flip-flop.

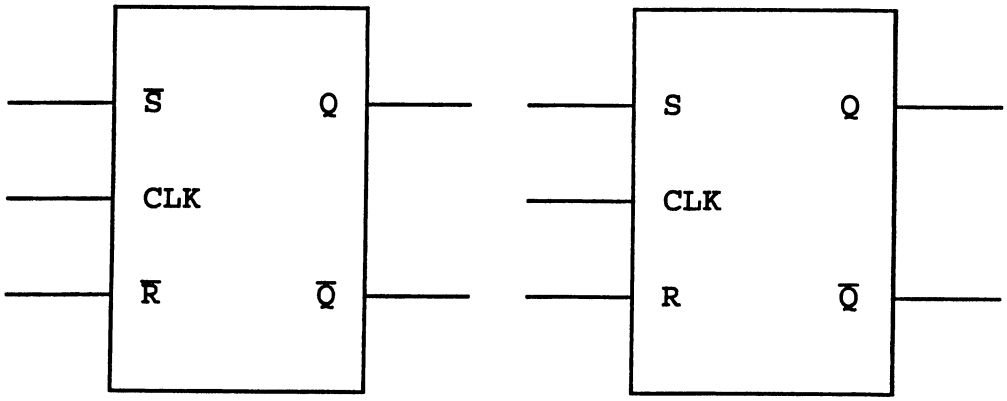
Refer to the following figure for item 57.



57. This is the logic symbol for a(n)

a. level triggered D latch.
b. level triggered D flip-flop.
c. edge triggered D latch.
d. edge triggered D flip-flop.

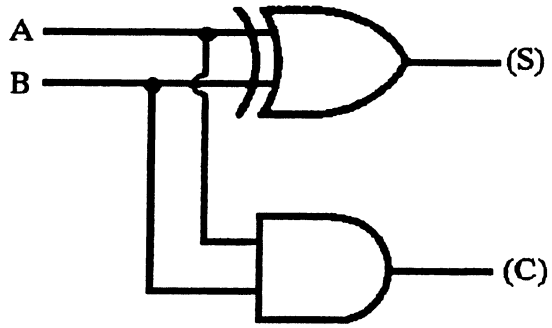
Refer to the following figure for item 58.



58. These are the logic symbols for

a. RS flip-flops.
b. RS latches.
c. RS counters.
d. RS registers.

Refer to the following figure for item 59.



59. Select the truth table that describes this circuit.

a.

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

b.

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

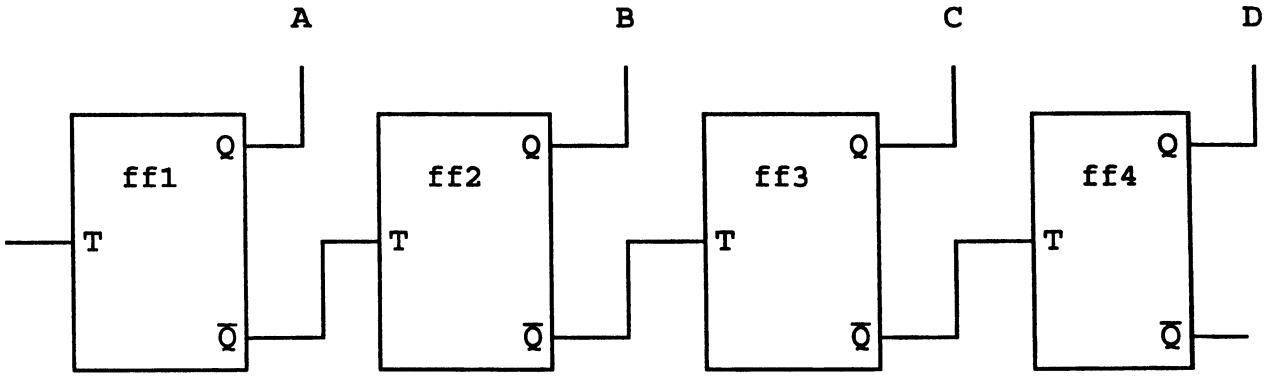
c.

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

d.

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

For items 60 and 61 refer to the following figure.



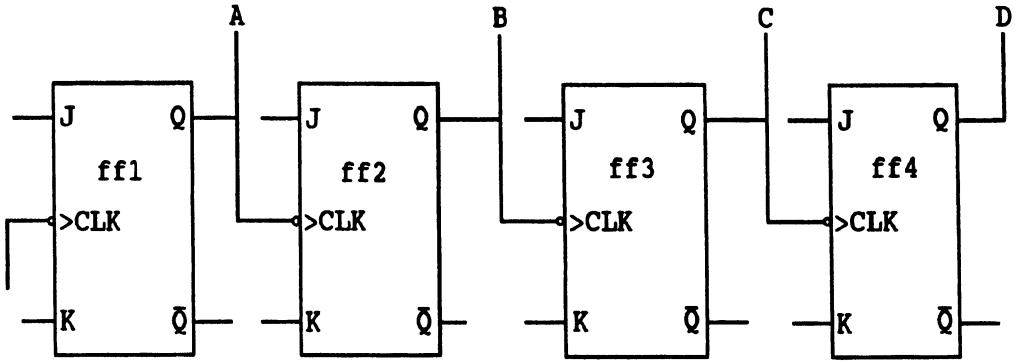
60. This is a logic diagram for a

a. T-latch counter.
b. decade counter.
c. T-latch register.
d. 2 bit counter.

61. This is a(n) _____ counter.

a. synchronous
b. parallel
c. asynchronous
d. 5 bit

For items 62 and 63 refer to the following figure. Assume that J and K are held logic high and the device is cleared immediately prior to the conditions stated in each item.



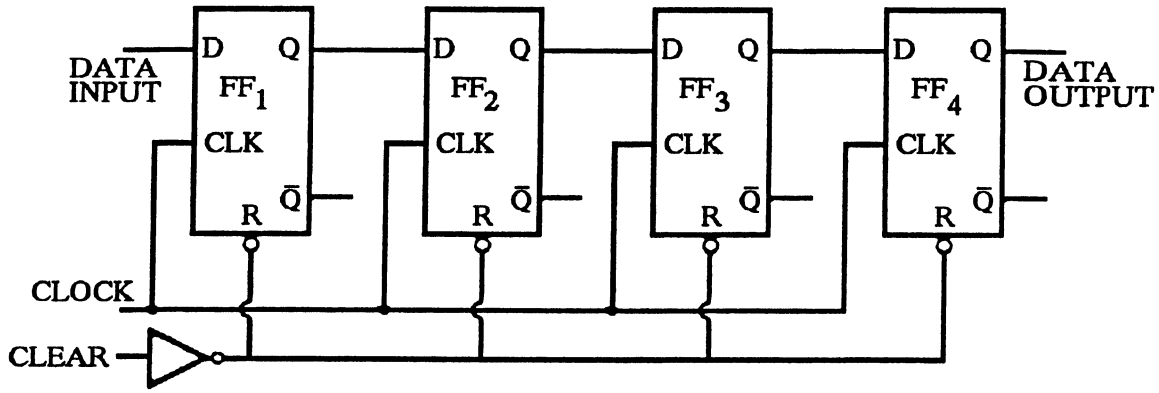
62. After five input pulses, the logic levels felt at points A, B, C, and D, respectively, are _____, _____, _____, and _____.

a. high, low, high, low
b. low, high, low, high
c. high, low, low, high
d. low, high, high, low

63. After six input pulses, the logic levels felt at points A, B, C, and D, respectively, are _____, _____, _____, and _____.

a. high, low, high, low
b. low, high, low, high
c. high, low, low, high
d. low, high, high, low

For items 64 through 66 refer to the following figure.



64. This a logic diagram for a(n)

a. shift register.
b. shift counter.
c. sequential counter.
d. non-sequential counter.

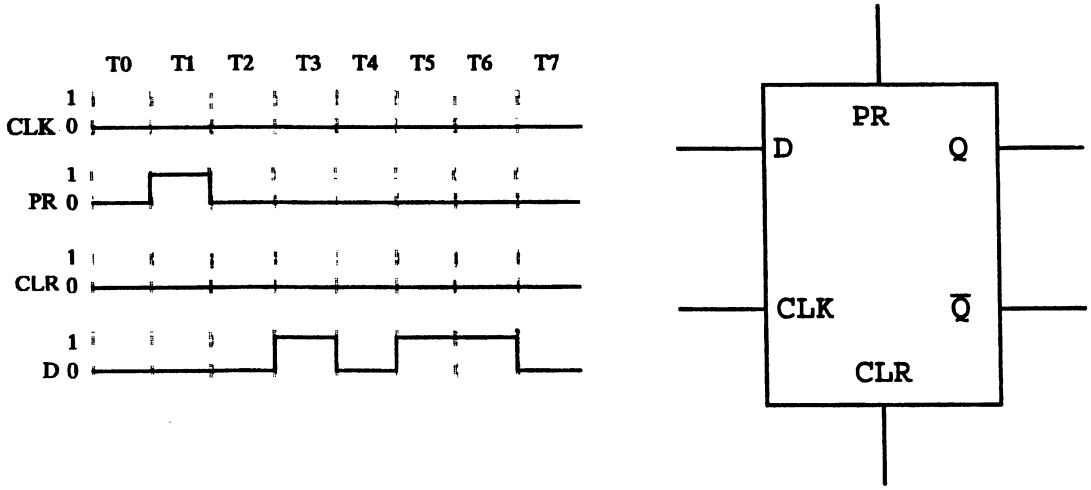
65. The format of this circuit is

a. parallel.
b. down counting.
c. up counting.
d. serial.

66. It takes _____ input pulses before this device produces an output.

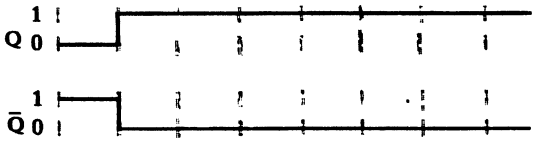
a. two
b. four
c. eight
d. ten

For Item 67 refer to the logic diagram below. Review the diagram, then answer the item.

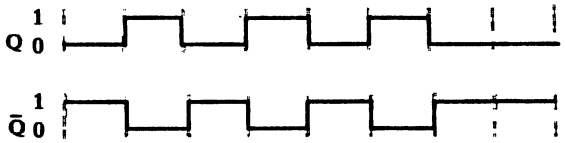


67. Which of the following timing diagrams is consistent with this logic diagram?

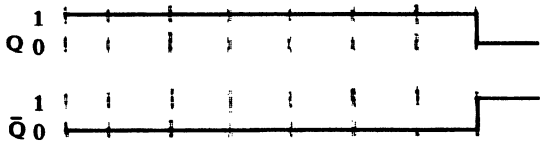
a.



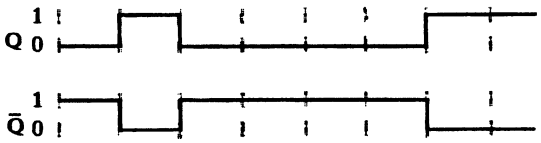
b.



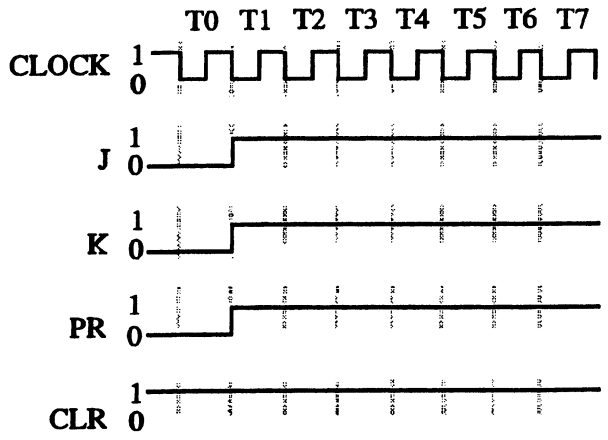
c.

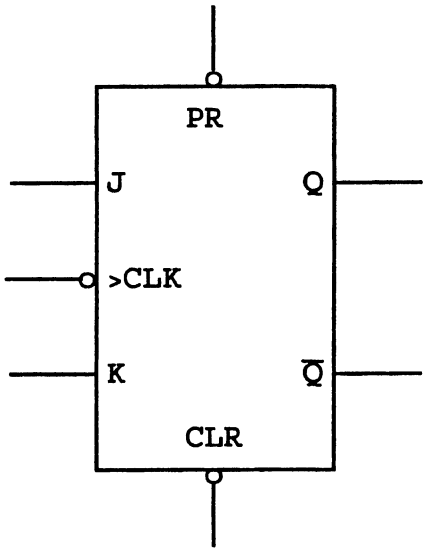


d.



For item 68 refer to the following logic and timing diagrams. Review the diagrams, then answer the items.

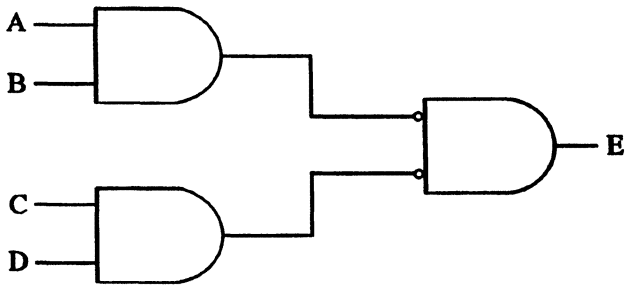




68. When is Q logic high?

a. T0
b. T0, T2, T4, T6
c. T1, T3, T5, T7
d. T1, T2, T3, T4, T5, T6, T7

For items 69 and 70 refer to the following logic diagram. Review the diagram and answer the items.



69. Which truth table describes this circuit?

a.

A	B	C	D	AB	CD	E
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	1	0

c.

A	B	C	D	AB	CD	E
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

b.

A	B	C	D	AB	CD	E
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	0	0	1
0	1	1	1	0	1	1
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	0	1
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	0

d.

A	B	C	D	AB	CD	E
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	1	0
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	1	1

70. Which equation describes this circuit?

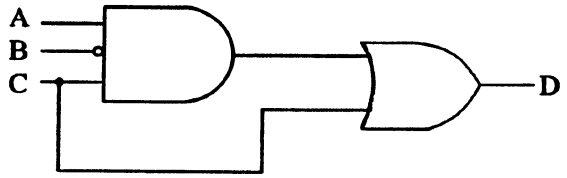
a. $\overline{ABCD} = E$

b. $\overline{\overline{AB} + \overline{CD}} = E$

c. $\overline{A} \times \overline{B} \times \overline{C} \times \overline{D} = E$

d. $\overline{AB} \times \overline{CD} = E$

For items 71 and 72 refer to the following logic diagram. Review the diagram and answer the items.



71. Which equation describes this circuit?

a. $ABC + C = D$

b. $A\overline{B}C + C = D$

c. $A + B + C = D$

d. $A + \overline{B} + C = D$

72. Which truth table describes this circuit?

a.

A	B	C	$A\overline{B}C$	D
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

b.

A	B	C	ABC	D
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

c.

A	B	C	$A\overline{B}C$	D
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	1

d.

A	B	C	ABC	D
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0	0
1	1	1	0	1

LAST PAGE OF THE REVIEW LESSON

REVIEW LESSON ANSWER KEY

Course Title: Fundamentals of Digital Logic Course Number: 28.6g

Review Lesson Solutions		Reference Work Units
1.	b	1101
2.	a	1101
3.	a	1102
4.	d	1102
5.	a	1201
6.	a	1201
7.	c	1202
8.	c	1202
9.	a	1203a
10.	c	1203b
11.	b	1203c
12.	b	2102
13.	a	2102
14.	c	2102
15.	d	2101
16.	a	2105b
17.	a	2105
18.	c	2104b
19.	c	2104
20.	b	2205
21.	b	2205
22.	a	2202
23.	c	2202
24.	d	2203
25.	a	2203
26.	d	2204
27.	a	2204
28.	d	2302b
29.	b	2302a
30.	c	2302a
31.	d	2303
32.	d	2303
33.	a	2304
34.	a	3101b
35.	a	3101c
36.	c	3101d
37.	a	3201c
38.	a	3202d

		Reference Work Units
39.	b	3101a
40.	a	3203
41.	a	3201
42.	b	3206
43.	d	3203b
44.	b	3203b
45.	b	2203
46.	a	2202
48.	d	2205
49.	d	2103
50.	b	2204
51.	c	2204
52.	d	2103
53.	b	2104
54.	c	3203a
55.	a	3201d
56.	d	3204c
57	b	3203b
58.	a	3202
59.	d	3301
60.	a	3302
61.	c	3302
62.	d	3302c
63.	d	3302c
64.	a	3303
65.	d	3303
66.	b	3303
67.	a	4104
68.	b	4104
69.	a	4103
70.	d	4102
71.	b	4102
72.	c	4103

GPO U.S. GOVERNMENT PRINTING OFFICE:2005-320-084/81766