INSTRUCTION MANUAL

DIGI-COMP® 1

first real operating digital computer in plastic

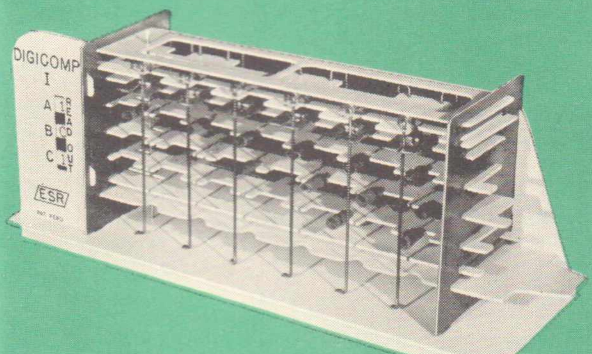


TABLE OF CONTENTS

INTRODUCTION

You now own a DIGITAL COMPUTER! What can DIGI-COMP I do? What does it mean to us that today our civilization is advanced enough to design and build complex machines such as digital computers?

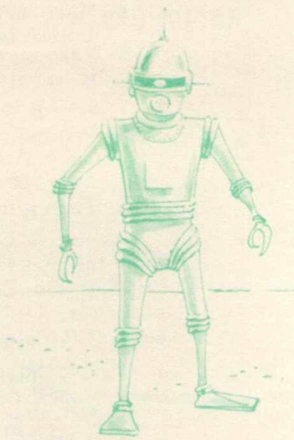
For many years man has tried to substitute a machine for man's brain or to put it another way to **"AUTOMATE"** the machine. The idea of a machine that thinks has always fascinated man. ROBOTS have long been a favorite character of science fiction writers. The science fiction robot is generally thought of as a machine built in the shape of a man whose purpose is to serve man. It cannot develop new and original thoughts – **IT CAN ONLY DO WHAT ITS MASTER INSTRUCTS IT TO DO.**

Today we have real robots. These robots do not look like the tin man in the Wizard of Oz – in fact they do not look like men at all, but they do the work of men and usually do it much better. They can be very simple – such as the thermostat in your home, an automatic elevator, traffic signals, automatic gearshift, automatic toll collectors and many other automatic devices which you see every day. There are also the complex robots that guide our satellites, translate languages, figure payroll checks, solve complex mathematical problems – in fact, they will do almost anything we want them to do. These complex automatic machines are not called robots – THEY ARE CALLED COMPUTERS.

WHAT IS A COMPUTER?

The word COMPUTER can mean several things. If you have ever counted on your fingers YOU could be called a DIGITAL COMPUTER. The word digit means finger (or toe) and a computer is someone or something that does number work.

Large digital computers are sometimes called "THINKING MACHINES". This is wrong. NO MACHINE CAN THINK – it can only do what it is told to do. But it can add, subtract, multiply and divide very, very fast.



Computers range in size from your DIGI-COMP I to small desk size adding machines to the "ELECTRONIC BRAINS" which can occupy an entire room and even in some cases an entire building. To give you an idea of the usefulness of these amazing machines, some computers can do number or mathematical problems in minutes or hours that would take more than your lifetime to solve with a paper and pencil. So you can easily see that without these complex machines we would still be a long way off from launching astronauts into space, designing space craft for moon flights, being able to accurately predict the path of hurricanes, storms and tornadoes or controlling the automatic telephone dialing system, and many, many other things that affect our daily living.

Your DIGI-COMP I is obviously not the size of a giant "electronic brain". It is a mechanical equivalent of an electronic digital computer. With DIGI-COMP I you can PLAY GAMES, SOLVE RIDDLES, and DO ARITHMETIC in the same way as you would on a large digital computer.

DIGI-COMP I is the first real BINARY COMPUTER which works MECHANICALLY the same way as giant electronic digital computers which work electrically.

YOUR DIGI-COMP can be considered as a small version of an actual computer – in fact, with the addition of many more parts, DIGI-COMP could solve very large problems just as an electronic digital computer does. The main difference is that since DIGI-COMP is mechanical it would be much slower and larger than an electronic computer. But the important thing is that EVERYTHING YOU LEARN ON DIGI-COMP CAN BE USED ON LARGE ELECTRONIC DIGITAL COMPUTERS.

OPERATING DIGI-COMP

You now have a completely assembled digital computer. Before putting your first experiment on DIGI-COMP let us make sure all the parts in the computer are working properly. This is called the CHECK OUT.

THE CHECK OUT

STEP 1 – See Figure 1. Move the CLOCK all the way "OUT" (to the right).

STEP 2 – See Figure 1. Make sure all LOGIC RODS are in the "ACTIVE" position (turn to the right).

STEP 3 – See Figure 1. Slowly move CLOCK all the way "IN" (to the left) and then all the way "OUT" (to the right).

THIS IS CALLED **"CYCLING"** THE CLOCK.

CYCLE the clock about 5 times.

Each time you move the clock "in" then "out" a "CYCLE" is completed.

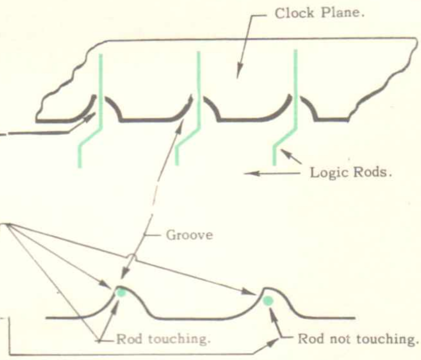
STEP 4 – Slowly move the clock so that the logic rods are in this position.

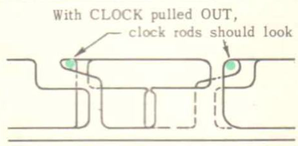
STEP 5 – Carefully look down at the logic rods to see if they are ALL touching the groove of the CLOCK PLANE.

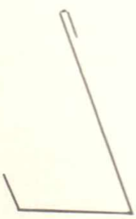
If all logic rods touch, go to Step 6.

If any logic rod is NOT touching the groove of the clock plane:

a. Make sure the logic rod is in the top and bottom holes.
b. Make sure it moves EASILY.
c. If it still doesn't touch, make sure the spring is on properly. If so, it doesn't work because the spring has been bent out. Take it out of the machine and *very carefully* bend it in, until it is this shape. Then reinstall it *carefully*.







STEP 6 – Slowly move the clock all the way out (to the right). Take the computer by the base and turn it around. Look at the CLOCK RODS in the BACK of the computer. See if they are ALL touching the inner edge of the clock plane.

If they all touch, go to Step 7. If one does not touch it is not through the clock planes correctly. Take it out and put it in right. (Step 12 of the Assembly Instructions).

STEP 7 – With the FRONT of the computer facing you, slowly move the clock all the way in (to the left). Look at the CLOCK RODS in the BACK of the computer. They should ALL be as in Figure 2. If they are all right go to the Check Out Problem.

If one is not right:

a. Check to see if the logic rods are all in the slots of the sliders (see Step 9 of Assembly Instructions, and Figure 1).

b. Check to see if the clock rod is in the slot of the slider (see Step 12 of Assembly Instructions, and Figure 2).

c. Check to see if BOTH planes 6 and 7 are moving all the way (if not see Step 6 of Assembly Instructions, and Figure 2).

You are now ready to put a "CHECK OUT" problem into DIGI-COMP. A check out problem is one to which you know the answer. By checking this answer against DIGI-COMP'S answer you can determine if the computer is operating properly.

FIGURE 1

FRONT OF COMPUTER

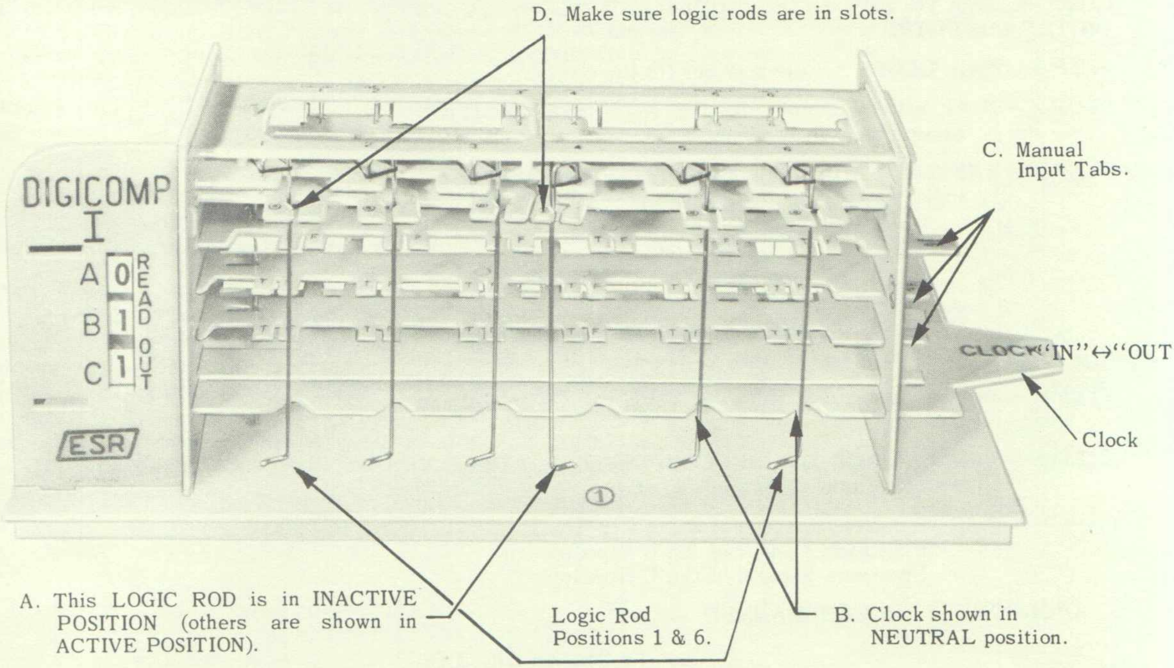
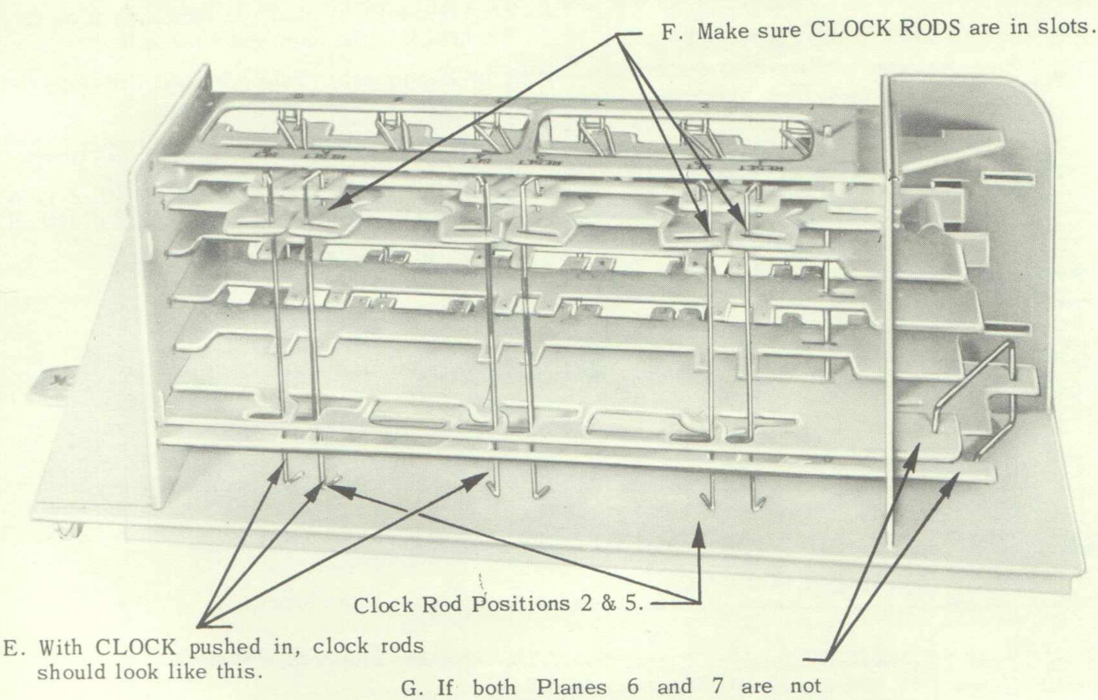


FIGURE 2

BACK OF COMPUTER



CHECK OUT PROBLEM

First you must "PROGRAM" DIGI-COMP. To PROGRAM DIGI-COMP MEANS TO PUT LOGIC TUBES AND CLOCK TUBES ON THE COMPUTER.

STEP 1 – Move CLOCK all the way out (to the right).

STEP 2 – Move all logic rods to the INACTIVE POSITION (Turn to the left).

STEP 3 – With the front of DIGI-COMP facing you FIRMLY put logic tubes (short tubes) onto:

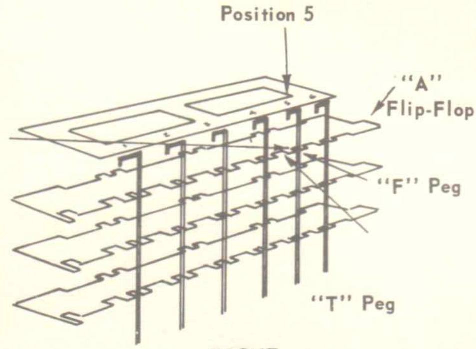
"T" peg of the A flip-flop at Position 5.
"F" peg of the A flip-flop at Position 6.
"F" peg of the B flip-flop at Position 1,
"T" peg of the B flip-flop at Position 2.
"F" peg of the C flip-flop at Position 3.
"T" peg of the C flip-flop at Position 4.

STEP 4 – Move all logic rods to the ACTIVE POSITION (turn to the right).

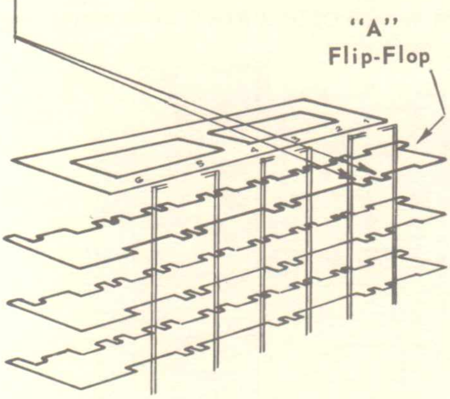
STEP 5 – With the BACK OF DIGI-COMP facing you FIRMLY put clock tubes (long tubes) onto:

Positions 1 and 2 of the A flip-flop.
Positions 3 and 4 of the B flip-flop.
Positions 5 and 6 of the C flip-flop.

DIGI-COMP is now PROGRAMMED.



FRONT



BACK

To Operate DIGI-COMP:

STEP 1 – With the front of the computer facing you make sure the clock is all the way out (to the right). With the Manual Input Tabs SLOWLY move the flip-flops "in" (left) and "out" (right), one at a time, to see if any of the logic tubes are hitting the logic rods.

If a logic tube hits a logic rod, push it on firmly. If it still hits the logic rod replace it.

STEP 2 – With the Manual Input Tabs move all flip-flops to the right (see 0 0 0 in the Read-Out).

STEP 3 – Slowly cycle the clock a number of times.

TO CYCLE THE CLOCK SLOWLY MOVE IT ALL THE WAY IN (LEFT) AND OUT (RIGHT).

You should see in the Read Out:

Before the first clock cycle	0 0 0
After the first clock cycle	0 0 1
After the second clock cycle	0 1 1
After the third clock cycle	1 1 1
After the fourth clock cycle	1 1 0
After the fifth clock cycle	1 0 0
After the sixth clock cycle	0 0 0

As you continue to cycle the clock the sequence will repeat.

If you see this sequence, DIGI-COMP IS WORKING AS A DIGITAL COMPUTER.

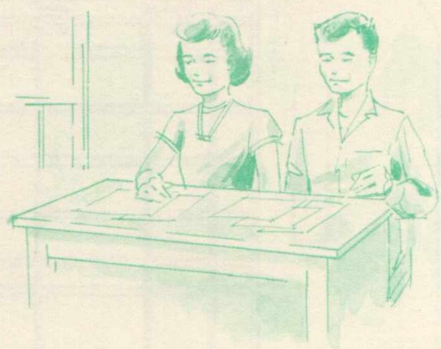
If you do not see this sequence go to Step 1 of Check Out Problem.

What is DIGI-COMP doing?

CHECK OUT PROBLEM CONTINUED

The experiment you have just completed with DIGI-COMP is a typical computer operation known as "COUNTING BY SHIFTING". "Counting by shifting" means that whatever is in the Read Out of a flip-flop is transferred (shifted) to the flip-flop above AND the opposite of what is in the Read Out of the top flip-flop is transferred (shifted) into the bottom flip-flop. You should see the following sequence:

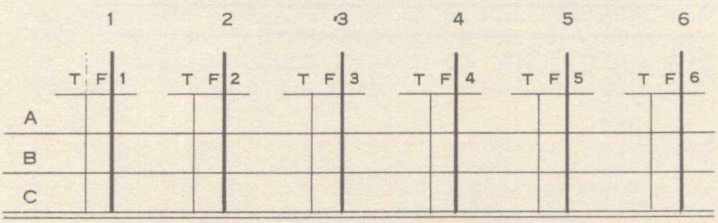
A	0	0	0	1	1	1	0
B	0	0	1	1	1	0	0
C	0	1	1	1	0	0	0



CODING SHEET

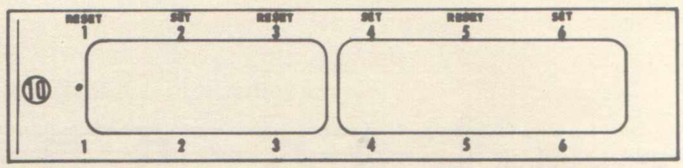
Instead of saying in words where to put the logic tubes and clock tubes, from now on we will use a "CODING SHEET". A coding sheet is used for all electronic digital computers to make it easier for the man to prepare problems for the computer. DIGI-COMP'S coding sheet will be used for the same purpose.

The coding sheet looks like DIGI-COMP to make it easy to use.



The numbers at the top of the coding sheet can be seen on the Top Plate. These are the POSITIONS.

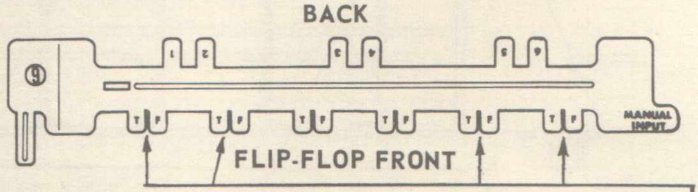
TOP PLATE



FRONT

The numbers in FRONT of the TOP PLATE are the POSITIONS of the Logic Rods. The numbers in BACK on the TOP PLATE are the POSITIONS of the Clock Rods. For example, find LOGIC ROD POSITIONS 1 and 6 on your computer. Now – find the CLOCK ROD POSITIONS 2 and 5 on DIGI-COMP (look at Figures 1 and 2 for the answer).

BACK



Each POSITION (number) on the CODING SHEET has under it a **T** and **F**. These correspond to the six sets of **T** and **F** PEGS on EACH Flip-Flop. If you look at DIGI-COMP you will see that under each of the 6 POSITIONS marked on the Front of the Top Plate there is a set of PEGS marked T and F on each Flip-Flop.

FIGURE 3A FRONT

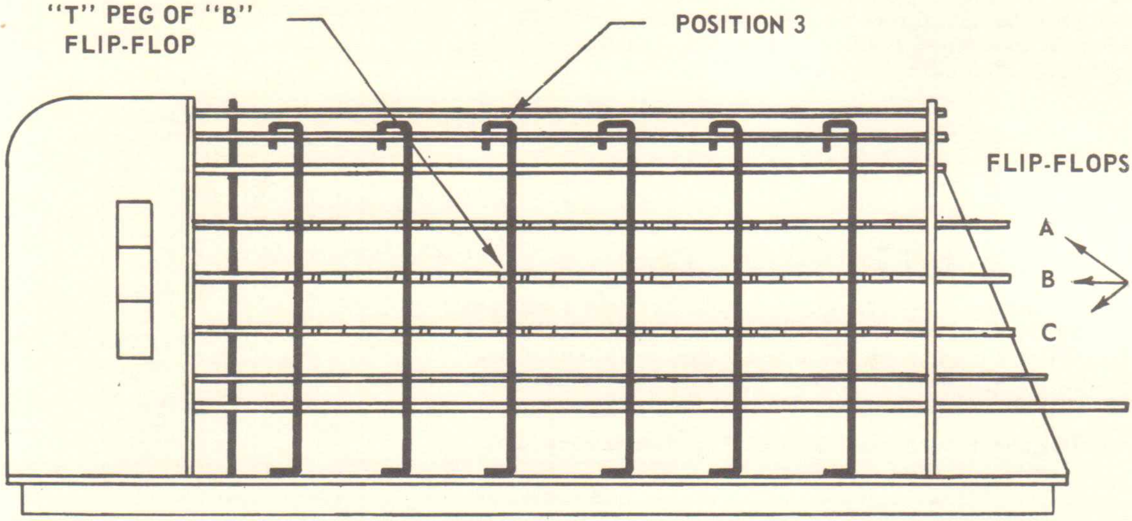
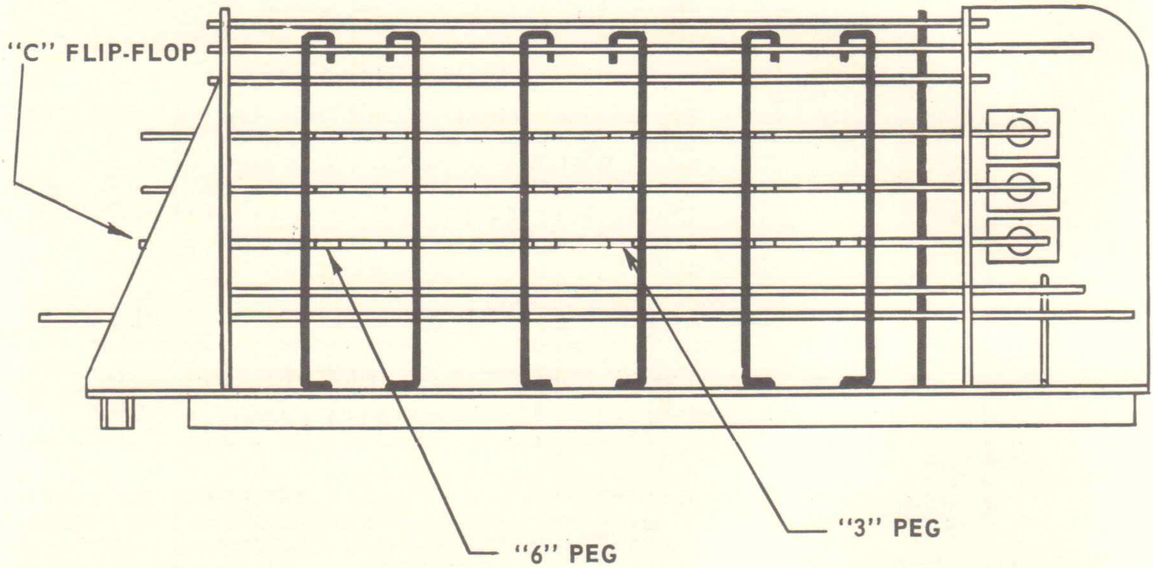


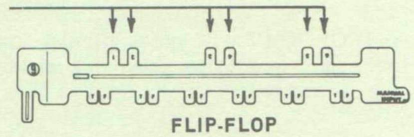
FIGURE 3B BACK



CODING SHEET CONTINUED

For example, find the **T** PEG of the B Flip-Flop at Position 3 on DIGI-COMP.

On the BACK of each Flip-Flop there are six PEGS marked **1, 2, 3, 4, 5, 6**. These pegs also correspond to the six positions marked on the TOP PLATE in BACK. On the Coding Sheet these PEGS are represented by the numbers to the right of the T-F UNDER EACH POSITION at the TOP of the coding sheet. For example, find the **3** and **6** PEGS of the C Flip-Flop on your computer (See Figure 3A and 3B for answers).



FLIP-FLOP

You may have noticed on the Top Plate next to the numbers in back, the words RESET and SET. This is to remind you that Clock Rod Positions 2, 4 and 6 must be used to SET a flip-flop and Clock Rod Positions 1, 3 and 5 to RESET a flip-flop.

> SET means to change a flip-flop from "0" to "1" (in the Read Out). (The Flip-Flop moves to the left for a SET).
>
> RESET means to change a Flip-Flop from "1" to "0" (in the Read Out). (The Flip-Flop moves to the right for a RESET).

If a flip-flop has a CLOCK TUBE on the PEG in CLOCK ROD POSITIONS 1, 3 or 5 it will RESET when the clock is cycled. If a Flip-Flop has a CLOCK TUBE on the PEG in CLOCK ROD POSITIONS 2, 4 or 6 it will SET when the clock is cycled.

Slowly cycle your computer and watch how this works in the back. Notice when a logic rod moves in it pushes the SLIDER. When the slider moves it pushes the clock rod out so that the clock rod is caught by the clock plane.

Now that you understand how the coding sheet corresponds to the various parts of the COMPUTER let's see how to use it.

The coding sheet is FILLED OUT BY MARKING IT with the following symbols:

- L – Firmly put a LOGIC TUBE (Short Tube) onto the indicated PEG of the Flip-Flop at the Position marked.
- C – Firmly put a CLOCK TUBE (long tube) onto the PEG of the Flip-Flop at the Position marked.
- (OUT) – Move the LOGIC ROD in this Position to "INACTIVE" (turn to the left).

Your computer now has the CHECK OUT experiment programmed on it. Try FILLING OUT The Coding Sheet below for the CHECK OUT Experiment on Page 4. For example, on page 4 it says PUT LOGIC TUBES into:

"T" PEG of the A Flip-Flop at Position 5.

You would mark an L as shown on the Coding Sheet.

Also on page 4 it says PUT CLOCK TUBES into:

Positions 1 and 2 of the A flip-flop.

You would mark C's where shown on the Coding Sheet.

	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A			C			C							L					
B																		
C																		

(Go to Page 28 to check your answer.)

Let's program another experiment using the Coding Sheet.

EXPERIMENT 2 – FINAL COUNT DOWN

Your DIGI-COMP will automatically signal the BLAST OFF of a space ship from Cape Canaveral at the right time.

To Operate DIGI-COMP:

STEP 1 – Put Front Panel Card I in place.

STEP 2 – Make sure CLOCK is all the way to the right.

STEP 3 – Remove ALL logic and clock tubes.

STEP 4 – "PROGRAM" (Put in Logic and Clock Tubes) DIGI-COMP according to the coding sheet. (Put logic tubes on first then put clock tubes in place).

	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A	L		C		L	C		L			L			L			L	
B							L		C		L	C		L			L	
C													L		C		L	C

STEP 5 – Manually (with the MANUAL INPUT TABS) move all flip-flops to the LEFT so that you see 1 1 1 in the Read Out.

STEP 6 – Cycle the clock slowly until 0 0 0 appears in the Read Out.

THIS IS THE "BLAST OFF" SIGNAL.

DIGI-COMP has counted 7 clock cycles starting at 1 1 1 and ending at 0 0 0 .

If you want to read about the LANGUAGE OF COMPUTERS go to the next chapter. If you want to try more experiments now go to page 15.

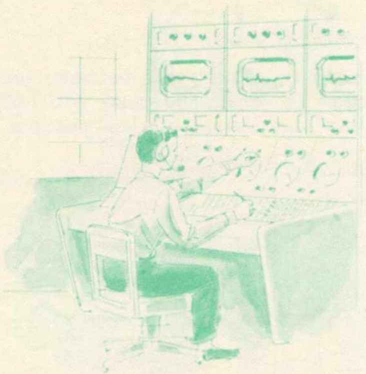
THE LANGUAGE OF COMPUTERS

Binary System

To understand DIGI-COMP you must learn the LANGUAGE OF COMPUTERS. If you are to tell DIGI-COMP what to do and if you are to understand the computer's answers, you must use a language you and DIGI-COMP both know.

Luckily this language is very simple – much easier to learn than it was to learn decimal numbers. THIS LANGUAGE MAKES THE OPERATION OF DIGITAL COMPUTERS POSSIBLE.

The computer's language is known as the **"BINARY SYSTEM"**. BINARY NUMBERS use only two symbols, 0 and 1, rather than the ten decimal numbers (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).



Since the binary system has only two symbols, these symbols can be used to mean many things. They may mean ON–OFF, TRUE–FALSE, YES–NO or any other kind of two way operation. Therefore, the binary system can be used in many devices – lights ON or OFF or, as in DIGI-COMP, two positions of a flip-flop – LEFT and RIGHT.

You have learned that in decimal numbers 683 actually means:

6 one hundreds + 8 tens + 3 ones.

Then 683 can be written as:

$$683 = 600 + 80 + 3$$
$$\text{or } 683 = 6 \times 100) + (8 \times 10) + (3 \times 1)$$
$$\text{or } 683 = (6 \times 10 \times 10) + (8 \times 10) + (3 \times 1)$$
$$\text{or } 683 = (6 \times 10^2) + (8 \times 10^1) + (3 \times 10^0)$$

(Any number raised to the power of zero is equal to one, thus $10^0 = 1$.)

In the number 683 the digit 6 is in the "hundreds" position – the digit 8 is in the "tens" position – and the digit 3 is in the "ones" or "unit" position.

In other words we may say that each digit in any decimal number is a different power of ten depending on the position of the digit within the number.

What will each position in the binary number system mean?

In the binary system we have only two symbols (0 and 1). Therefore, EACH POSITION REPRESENTS A DIFFERENT POWER OF TWO (2). That is, they are 2^0, 2^1, 2^2, 2^3, and so on.

An example of a binary number is: 110
This number is read: "ONE, ONE, ZERO."

The meaning of this binary number can be written as in the following example:

$$110 = (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$$
$$\text{or } 110 = (1 \times 2 \times 2) + (1 \times 2) + (0 \times 1)$$
$$\text{or } 110 = (1 \times 4) + (1 \times 2) + (0 \times 1)$$

Instead of saying a number has "digits" as in the decimal system we say a binary number has **"bits"**.

BINARY SYSTEM CONTINUED

In the binary system the FIRST position AT THE RIGHT is called the "ones bit" (2^0), the SECOND position from the right is the "twos bit" (2^1), the THIRD position from the right the "fours bit" (2^2), the fourth position the "eights bit" (2^3) and so forth.

You can TRANSLATE this binary number to a decimal number.

110 = (1 x 4) + (1 x 2) + (0 x 1)
= 4 + 2 + 0 = 6

(4's bit) (2's bit) (1's bit)

As you see, translating between the binary and decimal number systems is easy. The chart will help you translate between the decimal and binary number systems when you use DIGI-COMP.

TRANSLATION CHART

	BINARY SYSTEM		
	FOURS BIT	TWOS BIT	ONES BIT
	4	2	1
DECIMAL EQUIVALENT	2^2	2^1	2^0
0 =	0	0	0
1 =	0	0	1
2 =	0	1	0
3 =	0	1	1
4 =	1	0	0
5 =	1	0	1
6 =	1	1	0
7 =	1	1	1

Using the Translation Chart try translating these DECIMAL numbers to BINARY:

6 =

4 =

3 =

Also, try translating these BINARY numbers to DECIMAL:

101 =

001 =

111 =

(To check your answers go to page 28.)

BINARY ADDITION

It is very easy to add two binary numbers. THERE ARE ONLY FIVE RULES TO REMEMBER:

RULE 1	0 + 0 = 0
RULE 2	0 + 1 = 1
RULE 3	1 + 0 = 1
RULE 4	1 + 1 = 0 with a "1" carry (read: one, zero; 10)
RULE 5	1 + 1 + 1 = 1 with a "1" carry (read: one, one; 11)

Compare this with the decimal addition table! A few examples are:

1 + 0 = 1   2 + 3 = 5   3 + 6 = 9

8 + 4 = 2 with a "1" carry (12).

9 + 7 = 6 with a "1" carry (16).

In fact, there are **65** addition rules in the decimal system!

As you know when two decimal numbers are added, the "carry" from one column is added to the digits of the next column. For example:

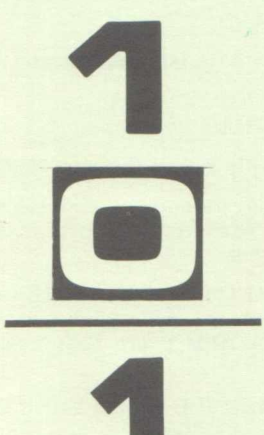
```
carry    111
         976
        +425
        ----
        1401
```

Exactly the same thing is done in the binary system. For example.

```
carry    1111
         11011
       + 10110
       -------
        110001
```

(Can you figure out the decimal equivalent of these numbers?) (See page 28 for the answer.)

In the binary system the "CARRIES" FROM ONE COLUMN ARE ADDED TO THE "BITS" OF THE NEXT COLUMN.



Although the binary system is very simple, it requires many binary bits to represent a number.

For example:

The decimal number **50** (only 2 decimal digits) is 110010 in binary (6 binary bits).

This is one of the reasons that although digital computers can be made of very simple parts, many thousands of these parts are needed.

NOTE that when more than three ones are added, there will be more than one carry. As an example:

```
carry ──→(1) (1)
         (1) (1)
           1  1             3
           0  1      or     1
           1  1             3
         + 1  1           + 3
         ------           ---
         101  0            10
```

BINARY SYSTEM IN COMPUTERS

As we said before, the binary system can be adapted to many uses, including YES–NO or TRUE-FALSE representation. Thus the modern computer can make simple decisions about complicated matters using the binary language.

The Computer need only decide **True** or **False** each time a question is asked. As an example, before a space ship can be launched, a digital computer is used to make the decision whether everything is A-O-K. It can do this by looking at thousands of parts of the space ship and decide whether they are operating properly. The computer does this by determining if the part is giving the correct answer to the "question" being asked of it. If the answer does not compare with what it should be then the answer is **False;** if it does compare, then the answer is **True.** There are so many of these **True–False** decisions to be made in such a short period of time, that the computer must make thousands of such decisions each second.

Your DIGI-COMP uses **True–False** as well as binary numbers (0 and 1 in the Read Out) as you will see in the next section on PROGRAMMING.

PROGRAMMING

Now that you understand the computer's language, let us see how DIGI-COMP uses the language. As we know, a computer has no brain, but it can work very complex problems, if it is **told** how to work them step-by-step. THIS IS CALLED PROGRAMMING.

You have been programming all your life! Whenever you tell someone to do something, or how to do it, you are programming. Your mother tells you it is raining outdoors. If you go outdoors then:

A. If you wear your rubbers

AND

B. If you wear your raincoat

THEN

C. You will not get wet.

This is a program.

Another way of stating this is:

STATEMENT A – If it is **true** that you wear your rubbers

AND

STATEMENT B – If it is **true** that you wear your raincoat

THEN

STATEMENT C – It is **true** that you won't get wet.

Now, let us forget about rain, raincoats, and rubbers for a moment. Just think about the connection between Statements A, B, and C. You can easily see that:

1. If A is **True** AND If B is **True** THEN C is True. It is important to understand that C is **true** **ONLY** when **A AND B** are **both true.** You can also reason:
2. If A is **false** AND If B is **true** THEN C is **false.** Also, you may say that:
3. If A is **true** AND If B is **false** THEN C is **false.** Also,
4. If A is **false** AND If B is **false** THEN C is **false.**

PROGRAMMING CONTINUED

In order to make things simpler, let us introduce some symbols to represent words or statements. We will now substitute:

A for Statement A if it is **true**.

$\overline{A}$ for Statement A if it is **false**.

B for Statement B if it is **true**.

$\overline{B}$ for Statement B if it is **false**.

C for Statement C if it is **true**.

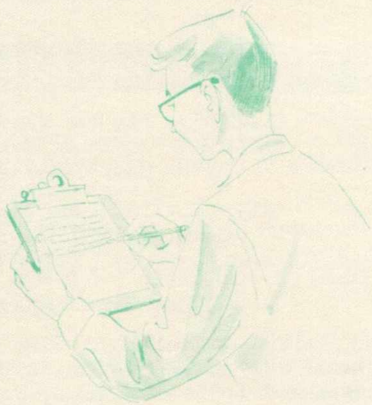
$\overline{C}$ for Statement C if it is **false**.

Also,

= (equals sign) is substituted for the word THEN.

• (dot) is substituted for the word AND.

We can now rewrite our sets of statements to read:



1. If A is **true** and B is **true**, then C is **true**, or:

   $A \cdot B = C$

2. If A is **false** and B is **true**, then C is **false**, or:

   $\overline{A} \cdot B = \overline{C}$

3. If A is **true** and B is **false**, then C is **false**, or:

   $A \cdot \overline{B} = \overline{C}$

4. If A is **false** and B is **false**, then C is **false**, or:

   $\overline{A} \cdot \overline{B} = \overline{C}$

These four sets of statements demonstrate one of the basic operations of a computer. This is called the **"And"** operation. To relate what we have just learned to the binary number system, we will let:

True be represented by 1

False be represented by 0

In other words, we can make a chart showing the following relationships:

A	=	**True**	=	1
$\overline{A}$	=	**False**	=	0
B	=	**True**	=	1
$\overline{B}$	=	**False**	=	0
C	=	**True**	=	1
$\overline{C}$	=	**False**	=	0

To demonstrate these relationships on your DIGI-COMP I, let the:

A flip-flop be Statement A

B flip-flop be Statement B

C flip-flop be Statement C (the conclusion)

If a flip-flop has a "0" in its read-out, then the flip-flop represents a **false** statement. If a flip-flop has a "1" in its read-out, then it represents a **true** statement.

PROGRAMMING CONTINUED

Let us program DIGI-COMP for the first set of statements:

1. A · B = C.

STEP 1 – Remove all the logic tubes from the front and all the clock tubes from the back of your computer.

STEP 2 – Put logic and clock tubes on according to the coding sheet.

	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A				L														
B				L														
C						C												

STEP 3 – Move the clock all the way out to the right.

STEP 4 – Move the C flip-flop all the way out to the right so that you read a "0" in its read-out.

STEP 5 – With the Manual Input Tabs set the A and B flip-flops to the left so that you read "1" in their read-outs.

STEP 6 – Cycle the clock once. A "1" should appear in the C read-out.

Try any other combination of A and B; for example:

A = 1 and B = 0 (C will NOT become "1").

WE HAVE JUST PROGRAMMED THE "AND" OPERATION FOR THE 1st SET OF STATEMENTS ON DIGI-COMP. This operation is represented on the computer by the logic rod in Position 2. Logic rod 2 will go "in" **only** when flip-flops A AND B are both "1". If either one of them is "0", logic rod 2 will be blocked from moving "in" by the logic tubes. (Set A = 1, B = 0 and watch the logic rod being blocked from moving in as you slowly cycle the clock.)

Now let us program the second set of statements on DIGI-COMP ($\bar{A} \cdot B = \bar{C}$). In this case, if A = 0 (A is **false**) AND B = 1 (B is **true**) then C must be made "0" when the clock is cycled. This means C must be **reset** when A = 0 and B = 1.

At POSITION 1 (reset) put a logic tube into the A flip-flop on the "F" peg (A is **false**) AND a logic tube on the B flip-flop on the "T" peg (B is **true**). Put a clock plug on the C flip-flop (on the back) at Position "1" (reset). This is the program for the second set of statements. You can fill out a coding sheet for this program as shown below.

	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A		L																
B	L																	
C			C															

Try this second set of statements on your computer.

STEP 1 – Move clock all the way to the right.

STEP 2 – With the Manual Input Tabs set "0" in A and "1" in B.

STEP 3 – Cycle the clock – C should have a "0" in its Read-Out.

Try PROGRAMMING Statements 3 and 4 for DIGI-COMP. (See Page 28 if you have trouble.)

YOU HAVE JUST PROGRAMMED YOUR COMPUTER.

CHAPTER III – EXPERIMENTS

Now you are ready to program the different problems, riddles and games that are in this chapter. There are **real** experiments showing how automation is used; for amusement there are logical riddles and games. At the end of the chapter, binary arithmetic and the different operating parts of large digital computing machines are discussed and demonstrated using DIGI-COMP.

In the experiments that follow the program has been worked out for you. If you would like to try programming an experiment yourself, go to Page 28.

You have a set of FRONT PANEL CARDS included with the Instruction Manual. Each experiment uses one of these cards. You must put the correct card for each experiment in the slots on the Front Panel. The card should be bent on dotted line to hold it in place.

IN ALL EXPERIMENTS MAKE SURE LOGIC TUBES AND CLOCK TUBES ARE **ONLY** IN THE POSITIONS INDICATED BY THE CODING SHEET.

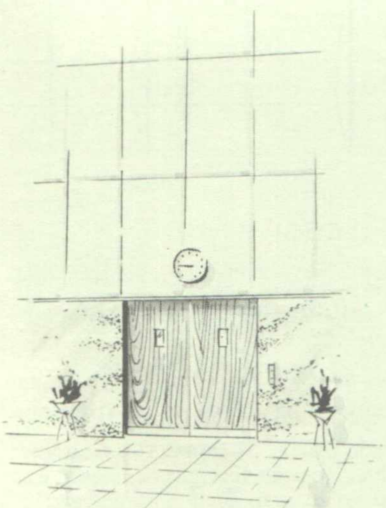
EXPERIMENT 3 – AUTOMATIC ELEVATOR

The automatic elevator is a common example of automation which uses MEMORY. A simple automatic elevator is the two floor elevator. The automatic system is required to REMEMBER at which of the two floors it has stopped.

In the elevator there are two buttons, one to tell the elevator to go to the 1st floor and the second to tell it to go to the 2nd floor. On each floor there is a button (CALL button) which you can push to tell the elevator to come to your floor.

Imagine the CALL button on the 1st floor is the same as the button in the elevator directing it to the 1st floor. Also, the 2nd floor CALL button is the same as the 2nd floor button in the elevator. Put FRONT PANEL CARD II into place. On this card there is marked "UP–DOWN". If the bottom flip-flop is at "1", the elevator is directed to the 2nd floor, and if it is at "0", to the 1st floor. (Look at the Read Out next to UP–DOWN.) The floor at which the elevator is located is indicated by a "1" in the Read Out by the floor.

DIGI-COMP can be used to show how this elevator works. **PROGRAM** DIGI-COMP according to the coding sheet.



	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A			C			C												
B									C			C						
C		L		L			L				L							

To operate DIGI-COMP:

STEP 1 – Make sure logic and clock tubes are ONLY in the positions indicated on the coding sheet.

STEP 2 – Make sure all LOGIC RODS are in the ACTIVE POSITION (turned to the right) and the CLOCK is all the way out (to the right).

STEP 3 – Manually set the **"up–down"** flip-flop to "1" if you want to go up or to "0" if you want to go down.

STEP 4 – Cycle the clock.

EXPERIMENT 4 – BANK LOCK

Each morning the president of the MONEY bank has to open the bank vault. It is a combination lock using binary numbers.

This morning the president was surprised to find he couldn't remember the combination. Can you open the vault for him?

Program DIGI-COMP as in the coding sheet below:

	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A	L		C	L		C				L						L		
B	L				L		(OUT)				L	C	(OUT)				L	
C	L			L						L						L		C

STEP 1 – Use Front Panel Card I.

STEP 2 – You guess the binary combination and manually set the combination into the three flip-flops.

STEP 3 – Cycle the clock ONCE.

If you see $\begin{smallmatrix}1\\1\\1\end{smallmatrix}$ in the Read-Out, you have successfully opened the vault! If you do not see $\begin{smallmatrix}1\\1\\1\end{smallmatrix}$ try again! What is your chance of opening the vault on the first try?

EXPERIMENT 5 – SEQUENTIAL BANK LOCK

To make it more difficult for anyone but the president of the bank to open the vault, a new type lock was installed. This lock requires **two** binary numbers in a **particular order** to open it. Program DIGI-COMP according to the coding sheet.

	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A					L	C					L					L		
B	(OUT)			L			(OUT)			L		C	(OUT)				L	
C				L						L							L	C

To operate DIGI-COMP:

STEP 1 – Use Front Panel Card I.

STEP 2 – Make sure clock is all the way to the right.

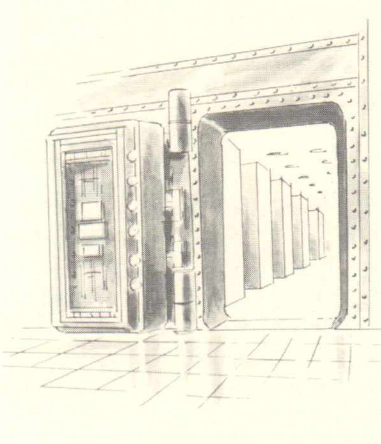
STEP 3 – Manually move all flip-flops to the **right.** (See $\begin{smallmatrix}0\\0\\0\end{smallmatrix}$ in the Read-Out.)

STEP 4 – Guess the first binary number $\begin{smallmatrix}0\\0\end{smallmatrix}$, $\begin{smallmatrix}0\\1\end{smallmatrix}$, $\begin{smallmatrix}1\\0\end{smallmatrix}$ or $\begin{smallmatrix}1\\1\end{smallmatrix}$ and manually enter it in the A and B flip-flops.

STEP 5 – Cycle the clock ONCE. If the C flip-flop goes to "1" then the number you entered was correct and go to step 6. If it stays at "0" try Steps 4 and 5 again.

STEP 6 – Guess the second binary number $\begin{smallmatrix}0\\0\end{smallmatrix}$, $\begin{smallmatrix}0\\1\end{smallmatrix}$ or $\begin{smallmatrix}1\\0\end{smallmatrix}$ not $\begin{smallmatrix}1\\1\end{smallmatrix}$ and manually enter it in flip-flops A and B.

STEP 7 – Cycle the clock ONCE. If you see $\begin{smallmatrix}1\\1\\1\end{smallmatrix}$ in the Read-Out, you have opened the vault. If not, try Steps 5 and 6 again.



This time what is your chance of opening the vault the first try?

EXPERIMENT 6 – HO-HUM (Logical Riddle)

In the Pacific Ocean there are two neighboring islands which few white men have seen. If you were to take a trip and visit these islands you would find the natives of island HO always tell the truth while the natives of island HUM always tell falsehoods. As an outsider you cannot tell to which island any of the natives belong. When you visit the islands you see a group of three natives standing around, and out of curiosity ask the first to which island he belongs.

FIRST NATIVE'S ANSWER: "BZYTPL."

Since he doesn't speak English, you cannot understand his answer. You ask his two friends: "What did he say?"

SECOND NATIVE ANSWERS: He say he belong "HUM".

THIRD NATIVE ANSWERS: "No, first man say he belong HO."

Riddle: To which island does the **third** native belong?

To operate DIGI-COMP:

STEP 1 – Put Front Panel Card III into the slots on the Front Panel.

STEP 2 – Make sure clock is all the way out (to the right).

STEP 3 – Program DIGI-COMP according to the coding sheet.

(You may find it helpful to move some of the logic rods to the INACTIVE POSITION (turn to the left) when putting on logic tubes.)

	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A	L									L			L			L		
B		L		OUT			OUT			L				L		L		
C			C									C			C			C

STEP 4 – Make sure logic rods 1, 4, 5 and 6 are in the ACTIVE POSITION (turn to the right).

STEP 5 – With the Manual Input Tabs enter the answers of the second and third natives into the flip-flops indicated by the Front Panel Card. If the answer was HO enter a "1", if HUM enter "0".

STEP 6 – Cycle the clock ONCE.

The answer in the Read Out will be "1" if the third native is from HO and "0" if he is from HUM.

Try entering other answers for the 2nd and 3rd Natives. DIGI-COMP will always correctly tell you from where the third native comes.

EXPERIMENT 7 – SPACE SHIP CHECK OUT
(A Job For An Astronaut)

This is the last moment before "blast off". It is time for you to make sure all the important parts ("systems") of your space ship are operating properly. You must now go through the final "check-out procedure".

If anything is wrong the launching will be canceled.

There are three systems to be checked:

1. Oxygen system
2. Space-Ship controls
3. Radio

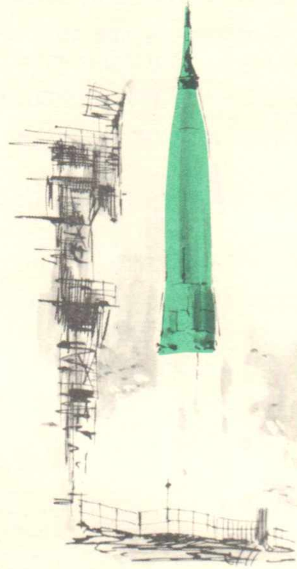
To check these systems the following questions are asked. They must be answered correctly to avoid a cancellation of the launch.

1. How much oxygen is flowing?
2. Is the "control stick" in the correct position?
3. To what frequency is your radio turned?

All 3 answers have a numerical value. These values will be checked one at a time with the values stored in DIGI-COMP. If they check correctly the computer will indicate that the system is "A-O.K." If not, the computer will indicate a malfunction (not working properly).

Program DIGI-COMP according to the coding sheet:

	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A					L	C	L			L						L		
B	OUT			L			L		C		L	C	OUT			L		
C				L			L			L							L	C



To operate DIGI-COMP:

STEP 1 – Use Front Panel Card I.

STEP 2 – Move all logic rods to INACTIVE (turn to the left) except Nos. 2 and 3.

STEP 3 – Guess how much oxygen is flowing and manually move the flip-flops to the numerical value (anything from 000 to 111).

STEP 4 – Cycle the clock ONCE. If the result is 1/1/1 in the Read-Out, then the system is operating "A-O.K.". If the result has a "0" ANYPLACE, then the system is NOT operating correctly. Move the flip-flops and cycle the clock until you get the "A-O.K." signal, 1/1/1.

STEP 5 – Repeat Steps 3 and 4 for the next two questions. For the second question move logic rod number 4 to the ACTIVE POSITION (turn to the right). For the third question move logic rod number 6 to the right.

When all three systems are "A-O.K." you are ready for your interplanetary voyage.

EXPERIMENT 8 – "GUESS THE NUMBER"
(A Logical Riddle)

Here is a **logical riddle** to test your reasoning powers. Have a friend choose any number between 0 and 7, (0, 1, 2, 3, 4, 5, 6, 7). You are to find the number he chose from his answers to three questions. He will answer Yes or No to each question.

QUESTION 1: Is the number even?

QUESTION 2: Is the number 0, 1, 2 or 3?

QUESTION 3: Add 10 (ten) to the number and divide the result by 6; is the **remainder** 0, 1, 2 or 3? (For example, if the number is 5, then 5 + 10 = 15 and 15 divided by 6 is 2 plus a remainder of 3. The answer to the statement would be **Yes** . If the number is 7, then 7 + 10 = 17 and 17 divided by 6 is 2 plus a remainder of 5. The answer would be **No.**)

Now try it:

STEP 1 – Put Front Panel Card IV in place.

STEP 2 – Program DIGI-COMP according to the coding sheet.

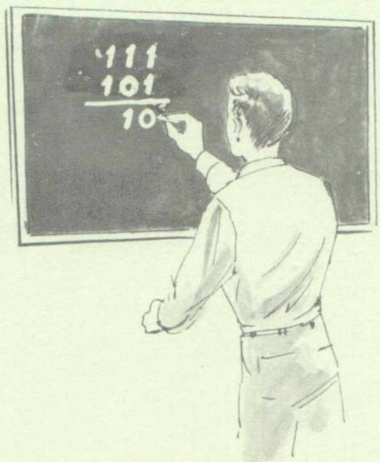
	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A	L		C		L	C												
B							L		C		L	C	L				L	
C								L			L		L		C		L	C

STEP 3 – Make sure clock is all the way out (to the right).

STEP 4 – With the Manual Input Tab enter the answers to the three questions into the flip-flop indicated by the Front Panel. Enter a "1" if the answer was YES. Enter a "0" if the answer was NO.

STEP 5 – Cycle the clock ONCE.

The Read-Out will have the number your friend chose in the Binary System.



EXPERIMENT 9 – SPACE CAPSULE RE-ENTRY

Augie, the Astronaut, can re-enter the earth's atmosphere by pushing his Re-entry buttons in the correct sequence (order).). Luckily, his space ship was designed so that if he made a mistake and pushed the WRONG sequence, nothing would happen. But then he must try ALL over again.

Can you help Augie re-enter without making a mistake? Program DIGI-COMP as in the coding sheet.

	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A	L		C	L		C	L			L			(	)		(	)	
B		L					L		C	L		C	OU	T		OU	T	
C				L			L			L			(	)		(	)	

To Operate DIGI-COMP:

STEP 1 – Use Front Panel Card V.

STEP 2 – Make sure CLOCK is all the way out (to the right). Manually move all flip-flops to the RIGHT (see $\begin{smallmatrix}0\\0\\0\end{smallmatrix}$ in the Read-Out).

STEP 3 – Manually move the **"re-entry"** button to either "0" or "1". This is the choice of a bit in the sequence.

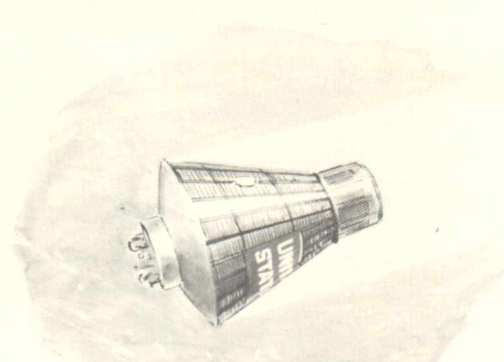
STEP 4 – Cycle the clock ONCE.

If the Re-Entry Button is in the CORRECT position the "Counter" will advance to $\begin{smallmatrix}1\\0\end{smallmatrix}$. If you did NOT choose correctly the "Counter" will be reset to $\begin{smallmatrix}0\\0\end{smallmatrix}$.

STEP 5 – Repeat steps 3 and 4 so that the counter sequence is $\begin{smallmatrix}0\\0\end{smallmatrix}$, $\begin{smallmatrix}1\\0\end{smallmatrix}$, $\begin{smallmatrix}0\\1\end{smallmatrix}$, $\begin{smallmatrix}1\\1\end{smallmatrix}$.

You have completed the re-entry for Augie's space ship successfully!

Nothing can go wrong. (Notice that continued operation of the "Re-Entry Button" and cycling the clock will not change the "counter"). DIGI-COMP has automatically STOPPED.



EXPERIMENT 10 – BINARY COUNTER

Binary counters are used in nearly all digital computers and automatic devices. There are many uses for these counters.

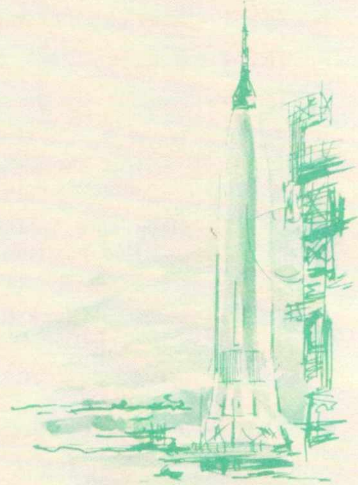
They may count **events** or **things**. For example, a binary counter is used to count time so a computer will know when it has to do different operations. Binary counters can be used to automatically count candy bars being sent to the grocery store.

You have already counted **down** to "blast off" a space ship. Now see if you can count **up** from 0 to 7 in binary. Try writing the count from 000 to 111.

To check yourself, use DIGI-COMP.

Program DIGI-COMP as in the coding sheet.

	1			2			3			4			5			6		
	T	F	1	T	F	2	T	F	3	T	F	4	T	F	5	T	F	6
A	L		C		L	C	L			L			L			L		
B							L		C		L	C	L			L		
C													L		C		L	C



To operate DIGI-COMP:

STEP 1 – Use Front Panel Card I.

STEP 2 – Make sure clock is out. Enter "0"s into all 3 flip-flops.

STEP 3 – Cycle the clock slowly a number of times. Look at the Read-Out and watch DIGI-COMP count!

THE "OR" OPERATION

In the section on Programming on page 12 you learned about the "AND" operation and how DIGI-COMP used the AND operation. If you look at the three statements again you can see there is **another** connection between them besides "AND".

STATEMENT A – If you **do not** wear your rubbers

OR

STATEMENT B – If you **do not** wear your raincoat

THEN

STATEMENT C – You **will** get wet.

In the shorthand we have used, this can be written as:

A **OR** B = C

"OR" OPERATION CONTINUED

To program this relationship on DIGI-COMP you will have to use the **Double Slider**. To put one of the double sliders on DIGI-COMP:

STEP 1 – Remove Stop Pin.

STEP 2 – Remove Top Plate.

STEP 3 – Move clock to Neutral (see Figure 1).

STEP 4 – Remove **Clock Rods** at positions 1 and 2.

STEP 5 – Remove Single Sliders at positions 1 and 2.

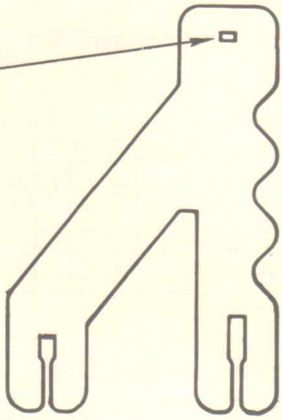
STEP 6 – Insert Double Slider from the rear of the machine with the slots on Front of the Slider going around Logic Rods 1 and 2.

STEP 7 – **Make sure hole in Back of Slider is by Clock Rod** position 2. If it is **Not** turn the slider over and put back in place as in Step 6.

STEP 8 – Insert a **Clock Rod** into position 2 in the back of DIGI-COMP. **Make sure** the Clock Rod goes through the hole in **back** of the **Double Slider.**

STEP 9 – Put on Top Plate.

STEP 10 – Put Stop Pin into place.



DOUBLE SLIDER

Now Program DIGI-COMP according to the Coding Sheet below. The symbol for the Double Slider is:

Means the Double Slider goes into the two positions marked by this symbol. The hole in the back of the slider goes in the Clock Rod position marked by the circle in the symbol.

	T	F	1	T	F	2	T	F	3	T	F	4	T	F	5	T	F	6
A	L																	
B				L				OUT			OUT			OUT			OUT	
C						C												

If you move A to "1" **OR** B to "1" then C will be **set** when the clock is cycled.

DIGI-COMP is **programmed** for the "OR" operation.

Now you may perform the rest of the experiments in the manual.

EXPERIMENT 11 – GAME OF NIM

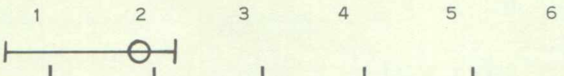
NIM, an ancient game over 2,000 years old, was played by the wise men of the east. Two wise men sat at a table, each one having a pile of stones by his side. The first wise man places either one or two stones in the center of the table. The second wise man **adds** 1 or 2 of his stones to the stones in the center of the table. Each player takes a turn until there are 7 stones in the pile in the center of the table. The wise man who added the last stones to make the pile **equal** to 7 is the smartest of all wise men.

DIGI-COMP claims to be very wise. Can **you** beat DIGI-COMP? Either you, or the computer, may go first.

To play NIM:

STEP 1 – Place Front Panel Card I in position.

STEP 2 – Program DIGI-COMP according to the coding sheet.



	T	F	1	T	F	2	T	F	3	T	F	4	T	F	5	T	F	6
A		L			L	C	L		C	L								
B		L					L				L	C	L		C	L		
C		L		L				L						L			L	C

STEP 3 – Make sure clock is all the way out.

STEP 4 – Make sure all logic rods are turned to the right.

STEP 5 – Make sure all flip-flops are all the way out (to the right).

YOU OR DIGI-COMP MAY GO FIRST.

If you go first:

STEP 6 – Add either 1 or 2 to the number in the Read Out 0 0 0

If you want to add 1, move the flip-flops with the Manual Input Tabs until you see 1 0 0 which is 1, in the Read Out.

If you want to add 2, move the flip-flops until you see 0 1 0, which is 2, in the Read Out.

STEP 7 –

It is now DIGI-COMP'S turn. Cycle the clock ONCE.

If the computer goes first:

STEP 6 –

Cycle the clock ONCE.

STEP 7 –

It is your turn. Use the translation chart to help you translate the numbers in the Read Out to decimal. Add 1 or 2 to this number and translate back to binary. Enter the binary number by moving the flip-flops with the Manual Input Tabs until you **see** the number in the Read Out.

YOU MUST ADD EITHER **1 OR 2** TO THE NUMBER IN THE READ OUT ON YOUR TURN.

STEP 8 – Repeat Steps 6 and 7 (taking turns with the computer) until you see 1 1 1 in the Read Out. This is 7.

THE FIRST TO MAKE THE PILE **EQUAL** TO 7 WINS!

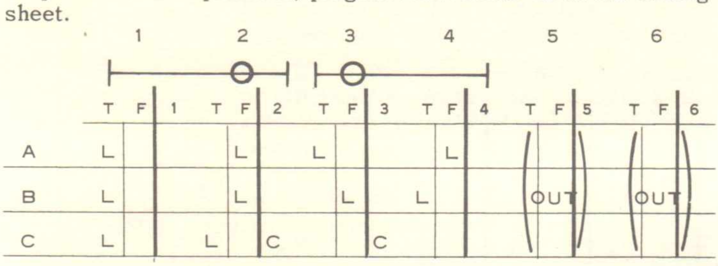
EXPERIMENT 12 – BINARY NUMBER COMPARATOR

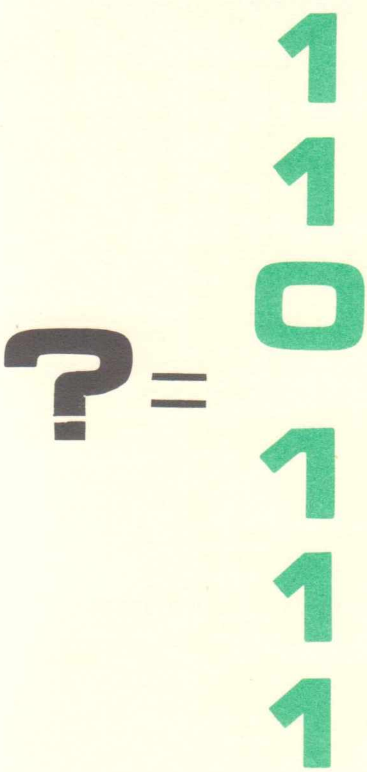
All electronic digital computers have the capability to determine whether two binary numbers are equal or not equal. On the basis of the answer (whether they are equal or not equal) the computer can make **decisions**.

It is this ability of the digital computer to decide what it will do next, based on the comparison of two binary numbers which makes people say the computer "thinks". The computer does not "think" since someone programmed it for each choice it could make.

DIGI-COMP can be programmed to compare two binary numbers and determine if they are equal or not equal.

To perform this experiment, program DIGI-COMP as in the coding sheet.





To operate DIGI-COMP:

STEP 1 – Use Front Panel Card VI.

STEP 2 – Make sure clock is out. Move the C flip-flop to "1".

STEP 3 – Enter the first bits of the two numbers into flip-flops A and B (see example below).

STEP 4 – Cycle the clock.

If the two binary bits are equal, C will have a "1" in its Read Out. If they are not equal C will have a "0" in its Read Out.
Stop when you see the first "0".
Repeat Steps 3 and 4 for the other bits of the two binary numbers. If at **any time** C is "0", then the two numbers are **not** equal.

FOR EXAMPLE: compare the two binary numbers

A – 1 0 1 1 0 ← enter these bits in A

and B – 1 1 1 1 0 ← enter these bits in B

(last bits) ← enter these two bits 1st
(fourth bits) ← enter these two bits 2nd
(third bits) ← enter these two bits 3rd and so on.

EXPERIMENT 13 – BINARY ADDITION

You have learned how to add binary numbers in the chapter on "The Language of Computers". Now you will **see** how a digital computer adds two binary numbers.

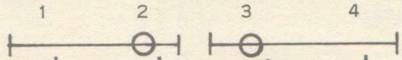
Remember, in addition, there can be a "carry" from the previous column.

For example: carry 11.

$$\begin{array}{r} 110 \\ +\ 011 \\ \hline 1001 \end{array}$$

The **adder** is a basic unit in all digital computers. Your DIGI-COMP can be programmed to operate as an adder which will work exactly as an adder in large electronic digital computers works.

Program DIGI-COMP as in the coding sheet.



	1			2			3			4			5			6		
	T	F	1	T	F	2	T	F	3	T	F	4	T	F	5	T	F	6
A		L			L	C	L		C	L				L		L		
B	L				L			L		L				L		L		
C		L		L			L				L				C			C

To operate DIGI-COMP:

STEP 1 – Use Front Panel Card VII.

STEP 2 – Make sure clock is out. Manually enter the first two binary bits to be added into the A and B flip-flops (see example). Make sure the C flip-flop is "0".

STEP 3 – Cycle the clock ONCE.

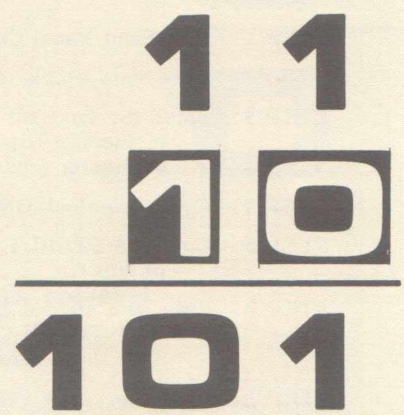
A will remember the **sum** and C will remember the **carry**.

STEP 4 – Copy the Read Out of A onto a sheet of paper. This is the first bit of your answer.

STEP 5 – Enter the second bit of the two numbers onto A and B. DO NOT TOUCH C (C remembers the carry from the previous step).

STEP 6 – Cycle the clock ONCE.

Repeat Steps 4, 5, and 6 until you have added the two numbers.. Remember to copy the carry after the last operation.



Try this example:

		Enter 3rd	Enter 2nd	Enter 1st
No. A		1	0	1
No. B		1	1	1
Carries C	1	1	1	
SUM (Answer in A)	1	1	0	0

Can you check this by translating to the decimal system?

EXPERIMENT 14 – BINARY SUBTRACT

Binary subtraction is as simple as binary addition.

The rules for binary subtraction are:

RULE 1: 1 - 1 = 0

RULE 2: 1 - 0 = 1

RULE 3: 0 - 1 = 1 with a "1" borrow.

RULE 4: 0 - 0 = 0

RULE 5: 1 - 1 and a "1" borrow = 1 with a "1" borrow.

RULE 6: 1 - 0 and a "1" borrow = 0

RULE 7: 0 - 1 and a "1" borrow = 0 with a "1" borrow.

RULE 8: 0 - 0 and a "1" borrow = 1 with a "1" borrow.

The program for a binary subtractor is given in the coding sheet. Compare this coding sheet with the one for the binary adder. Notice the only difference is at Position 5 and 6.

1 2 3 4 5 6

	T	F	1	T	F	2	T	F	3	T	F	4	T	F	5	T	F	6
A		L			L	C	L		C	L			L				L	
B	L				L			L		L				L		L		
C		L		L			L				L				C			C

To operate DIGI-COMP:

STEP 1 – Use Front Panel Card VIII.

STEP 2 – Make sure clock is out. Make sure C is "0".

STEP 3 – Enter the first bit of the **subtrahend** (1st No.) in the top flip-flop and the first bit of the **subtractor** (2nd No.) in the middle flip-flop.

STEP 4 – Cycle the clock ONCE.

STEP 5 – Copy the RESULT (in top flip-flop) on a piece of paper.

STEP 6 – Enter the second bits of the two numbers. DO NOT TOUCH C as it contains the "borrow" from the previous step.

STEP 7 – Cycle the clock ONCE. Repeat Steps 5, 6 and 7 until the two numbers are subtracted.

REMEMBER, the number subtracted must be less than, or equal to the number from which it is being subtracted.

AS AN EXAMPLE

	4th Bit	3rd Bit	2nd Bit	1st Bit
1st No.	1	0	1	0
2nd No.	0	1	1	1
(Borrow)	1	1	1	
(Difference)	0	0	1	1

Try to subtract any two binary numbers. Use DIGI-COMP to check your answer.

EXPERIMENT 15 – BINARY MULTIPLICATION

Binary multiplication is carried out in a way similar to decimal multiplication. The binary multiplication table is:

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

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

This is even simpler than binary addition!

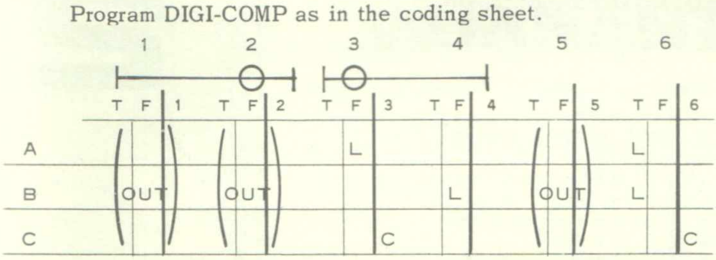
To multiply in the binary system you multiply by each bit of the multiplier and add. This is the same as in the decimal system.

For example:

In binary:	**In decimal:**
1101	13
x 1101	x 13
1101	39
0000	
1101	13
1101	
10101001	169

DIGI-COMP can be used to multiply two binary numbers according to the binary multiplication table.

Program DIGI-COMP as in the coding sheet.



TO OPERATE DIGI-COMP

STEP 1 – Use Front Panel Card IX.

STEP 2 – Enter the first multiplicand bit into the top and the first multiplier bit into the middle flip-flop.

STEP 3 – Cycle the clock ONCE. C will hold the result – copy it on paper.

STEP 4 – Enter additional bits of the multiplicand and cycle the clock, copying the answer each time (as in the example).

STEP 5 – Enter the second bit of the multiplier and repeat Step 4 for all bits of the multiplicand. Copy the answer as in Step 4 but shifted to the left one bit (see example).

STEP 6 – Repeat Step 5 for all bits of the multiplier.

You may use DIGI-COMP to add the partial products to get the final result (use the Binary Adder Experiment).

PROBLEMS FOR YOU TO PROGRAM

You have been programming different experiments on your DIGI-COMP for which the program has been given. Now you have the opportunity to do what persons who program large electronic digital computers must do. YOU can now work out programs and put these programs onto your DIGI-COMP.

Two experiments which you can program and put on DIGI-COMP are given below.

EXPERIMENT – AUTOMATIC COUNTER

The automatic counter is a modification of the Binary Counter (Experiment 10). Instead of starting to count over again as in the Binary Counter, the Automatic Counter must count 4 clock cycles, then stop. This means that if you cycle the clock when there is $\begin{smallmatrix}0\\0\\1\end{smallmatrix}$ in the Read Out, nothing will happen. Now YOU work out the program for this experiment and try it on DIGI-COMP!

EXPERIMENT – A LOGICAL RIDDLE

Three boys went on a fishing trip. The boys were Tom, Dick and Jim. When they returned they were asked who caught the fish. Being bright boys, they replied as follows:

If Tom didn't, then Dick did. "Tom did" and "Jim didn't" are not both true statements. If Dick did, then Tom did and Jim did.

Is it true that Jim did?

NOW MAKE UP YOUR OWN EXPERIMENTS TO PROGRAM ON YOUR DIGI-COMP.

ANSWERS:

Coding Sheet – See Page 7.

Correct coding sheet for CHECK OUT EXPERIMENT.

	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A			C			C							L				L	
B		L		L					C			C						
C								L		L					C			C

Binary System: Page 10

6 = 110; 4 = 100; 3 = 011

101 = 5; 001 = 1; 111 = 7

Binary Addition: Page 11

11011 = 27; 10110 = 22; 110001 = 49

Programming: Page 14

Statement 3: $A \cdot \overline{B} = \overline{C}$ means if A is **true** AND B is **false** then C is **false**.

and for Statement 4: $\overline{A} \cdot \overline{B} = \overline{C}$

	1 T	1 F	1	2 T	2 F	2	3 T	3 F	3	4 T	4 F	4	5 T	5 F	5	6 T	6 F	6
A							L							L				
B								L						L				
C									C						C			

GUARANTEE

YOUR DIGI-COMP I IS GUARANTEED TO BE OF THE HIGHEST QUALITY. IF ANY PART BREAKS E.S.R., INC. WILL REPLACE IT FREE OF CHARGE WITHIN 2 YEARS FROM DATE OF PURCHASE.

JUST SEND A NOTE INDICATING THE PART NUMBER AND NAME OF THE PART TO BE REPLACED ALONG WITH 25¢ IN COIN (TO COVER MAILING AND HANDLING COSTS) TO:

E. S. R., INC. – 32-34 LABEL ST., MONTCLAIR, N. J.