TEAM - A TRANSISTORIZED ELECTRONIC ADDING MACHINE

By

WILLIAM JOSEPH WATSON

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

1959

Submitted to the faculty of the Graduate School of
the Oklahoma State University
in partial fulfillment of the requirements
for the degree of
MASTER OF SCIENCE
May, 1960

OKLAHOMA
STATE UNIVERSITY
LIBRARY

SEP 2 1960

TEAM - A TRANSISTORIZED ELECTRONIC ADDING MACHINE

Thesis Approved:

Paul A. McCollum

Thesis Adviser

Harold Fristoe

Robert MacVicar

Dean of the Graduate School

452878

ACKNOWLEDGMENTS

The author wishes to express his indebtedness to Professor Paul A. McCollum, particularly for the instruction supplied in digital computer theory. Appreciation is also extended to Texas Instruments, Incorporated, for furnishing many of the diodes used in constructing the model.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

LIST OF PLATES

CHAPTER I

INTRODUCTION

The advent of the electronic digital computer has produced an unparalleled impact upon our society. Each year finds hundreds of entirely new computer applications. Many are referring to this new era as the "Age of Automation." As highly complex electronic computers replace semi-skilled workers, economists and philosophers busy themselves with long-range predictions and new social theories. (1).

Without competing with professional prophets, it seems safe to assume that electronic digital computers are to become commonplace within a relatively short time. Bookkeeping applications in the fields of banking and insurance are among the newer uses of the digital computer. However, it appears that modern computer advancement has been directed almost exclusively toward large-scale applications. There are myriads of potential small-scale computer applications which invite exploration. One of the most noteworthy of these is the general area of desk calculators.

Electro-mechanical desk calculators are available in a wide variety of types having several degrees of capability. These range from the simplest adding machine to versatile machines which perform the basic arithmetic operations of addition, subtraction, multiplication, and division Among the advantages to be derived from the electronic counterparts of these calculators would be reduction in size, silent operation, and provision for easy maintenance through the use of plug-in component cards. Also, the electronic version would perhaps better lend itself to mass production by automatic means than does the mechanical calculator.

The purpose of this thesis is to demonstrate the application of electronic digital computer techniques to the desk calculator. It was desirable to formulate the design criteria for a useful machine, to evolve the logical design of this machine, and to build a working model that would attest to the validity of the logic employed.

The circuits employed in the model did not result from long, laborious calculations. These circuits have been designed by rule-of-thumb techniques. This procedure was adequate for verifying the logical design. They are not elegant circuits, neither do they pretend to be. Of course, were this machine to be considered on a commercial basis, it would be desirable to carefully redesign the circuits. It was felt that this approach would be best in keeping with the stated purpose of the thesis and would aid in sufficiently limiting the scope of the thesis.

CHAPTER II

DESIGN CRITERIA

With the stated purpose of the thesis in mind, it follows that the design of such a machine should begin by a decision as to what capabilities it should incorporate. In an attempt to limit the scope of the thesis, it was decided that the singular capability of this machine should be that of addition. It was further decided that, in its operation, the machine should be made as simple as possible. Numbers to be added were to be entered from a ten-key keyboard. Any additional control keys were to be held to a minimum. The readout was to be visual. Operation was to be accumulative; that is, each new sum was to be the answer given by adding the new entry to the previous sum. All input and output data were to be decimal in form. For this design, it was decided that the machine would have a capacity of four decimal digits.

The choice of some variation of the binary number system is virtually an automatic one, for the design of an electronic digital computer. This is because logic performed on radix two numbers more easily lends itself to mechanization than for other radices. The choice becomes particularly attractive when it is desirable to use bistable multivibrators, or "flip-flops", as storage devices. The two stable states of a flip-flop may be made to correspond to the 0 and the 1 of the binary number system. Another important aspect of the radix two choice is that the logic may be analyzed through the use of Boolean algebra. (2).

If the arithmetic operations are to be performed on binary numbers, the requirement that input and output data be decimal in form predicates the need for two converters. One converter will be required to convert from decimal input to binary, and another for converting from binary to decimal output.

Let it be supposed that two numbers to be added are merely converted to their binary equivalents and added. As an example, consider the addition of 3275 and 5423. In writing the binary equivalents of these numbers, a short table of powers of two is helpful. (2).

2^0 =	1	2^7 =	128
2^1 =	2	2^8 =	256
2^2 =	4	2^9 =	512
2^3 =	8	2^{10} =	1024
2^4 =	16	2^{11} =	2048
2^5 =	32	2^{12} =	4096
2^6 =	64	2^{13} =	8192

Synthesizing the number 3275 from this table by adding powers of two

2^{11}	----	2048
2^{10}	----	1024
2^7	----	128
2^6	----	64
2^3	----	8
2^1	----	2
2^0	----	1
	sum	3275

Then, in binary form, the number 3275 is

(11)	(10)	(9)	(8)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)
1	1	0	0	1	1	0	0	1	0	1	1

Similarly, the number 5423 is converted to binary

2^{12}	----	4096
2^{10}	----	1024
2^8	----	256
2^5	----	32
2^3	----	8
2^2	----	4
2^1	----	2
2^0	----	1
	sum	5423

(12)	(11)	(10)	(9)	(8)	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)
1	0	1	0	1	0	0	1	0	1	1	1	1

Then, the binary addition to be performed is

```
          3275  ----   110011001011
          5423  ----  1010100101111
sum       8698  ---- 10000111111010
```

This example indicates that 14 binary digits are required to represent the largest numbers in four decimal digits; specifically, those numbers greater than 2^{13} or 8192. The input decimal-to-binary converter would, therefore, require 40 input lines, 10 for each digit, and 14 output lines. The output converter might require 28 input lines, two for each order, and 40 output lines. If parallel addition were used, there would be required a bank of 14 binary adders. For serial addition, a series of 14 additions would be required. (2).

The preceding example is indicative of the cumbersome nature of adding large numbers in the conventional binary system. In order to simplify this procedure, binary coding of the decimal digits is frequently employed (3). If a four-bit binary code is employed, the digits of the example may be represented by the following binary groups:

```
    0011        0010        0111        0101
     3           2           7           5

    0101        0100        0010        0011
     5           4           2           3
  ----------------------------------------------

    1000        0110        1001        1000
     8           6           9           8
```

The preceding technique suggests a method of addition that combines the serial and parallel methods. By this technique, the binary groups for a given decimal column are added in parallel. When this addition is completed, the operation is shifted serially to the other decimal columns. (2). Serial-parallel addition for a four-bit code requires, then, only four parallel binary adders. Conversion is also greatly simplified through the use of a four-bit code. Because the operations are performed on a decimal digit basis, the input converter need have only ten input lines and four output lines. The output conversion may be accomplished with eight input lines and ten output lines. This is explained in detail in the decoder discussion.

In this computer, as in others, there must be some provision for storage of data. As mentioned previously, the use of the binary number system permits the use of flip-flops as storage devices. Because of their straightforward design and of their applicability to this machine, flip-flops shall be used.

A summarization of the general ideas presented in this chapter indicates that the machine to be designed shall utilize serial-parallel addition, that the decimal digits shall be encoded by four-bit binary groups, and that flip-flops shall be used as storage devices. When these ideas are considered together with the requirements presented in the opening paragraph, it appears that the basic design criteria, prerequisite to the logical design, have been established.

CHAPTER III

LOGICAL DESIGN

The Basic System

For convenience in considering the machine from a system standpoint, a block diagram is presented in Figure 1. Assume that the example numbers are to be added, namely, 3275 and 5423. Furthermore, assume that 3275 was the last previous sum, and that 5423 is to be added to it.

3275	----	augend
5423	----	addend
8698	----	sum

If the digits of the addend were entered from the keyboard in descending order, that is, from left to right, beginning with 5 and ending with 3, it would be necessary to store the first three digits, the 5, 4, and 2. This is true because the addition must begin in the lowest order; in other words, 5 plus 3 is the first step in adding these two numbers. Obviously, the addition could not begin until the 3 had been entered.

If the digits of the addend were entered in reverse order, the need for this temporary storage of some digits would be eliminated. This is true only if the machine can perform a given digit addition prior to entry of the next digit. In the example, the first entry would be 3. This would be added immediately to the 5. The machine would then be ready to accept the 2 for addition to the 7, in the tens' order. In the

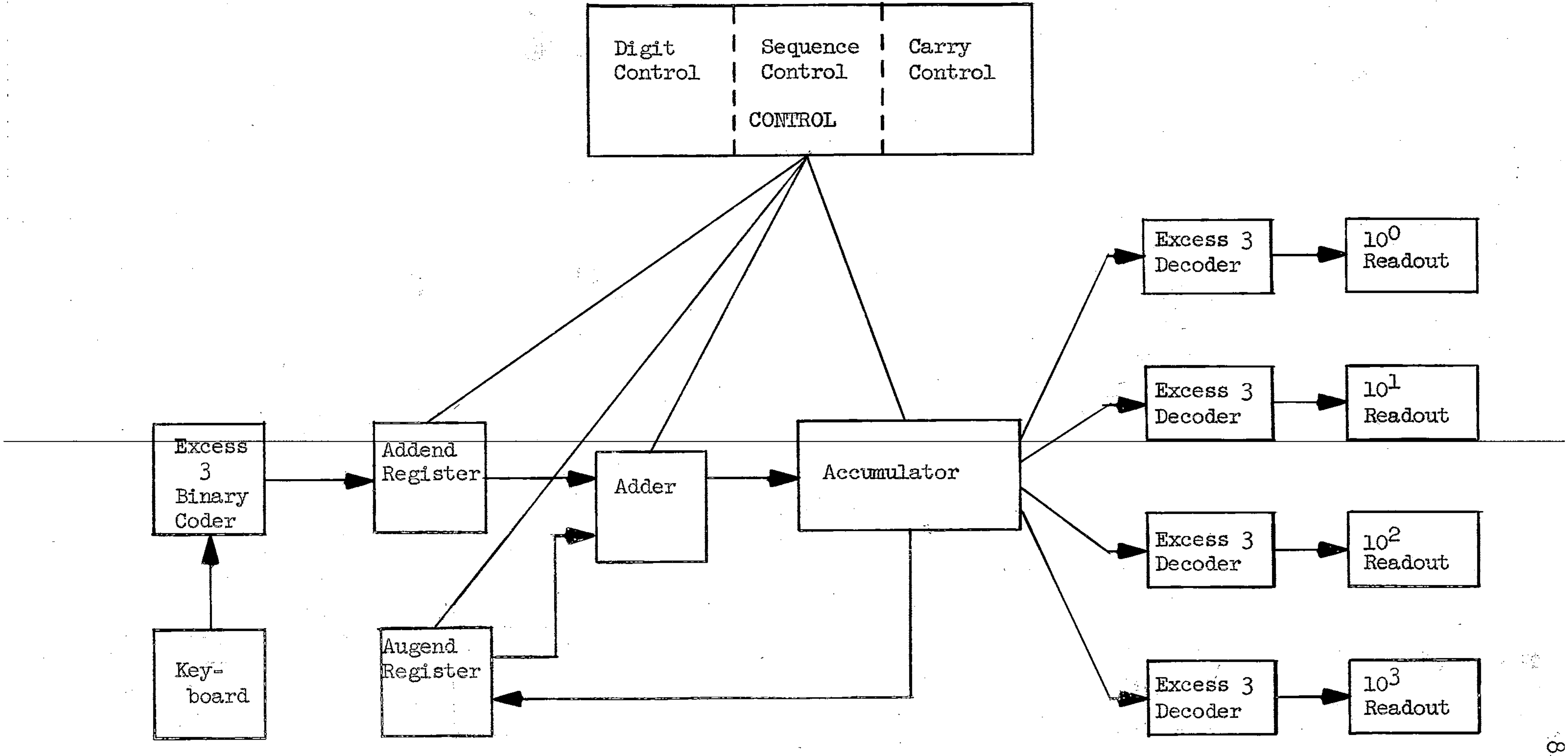


Figure 1. General Block Diagram

interest of minimizing the storage capacity required of the model, the method of entering digits in reverse order shall be used.

Choice of a Decimal Code

There are many possible ways of representing the ten decimal digits by means of four radix two digits. In the example of Chapter II, the coding was simply the binary equivalent of each decimal digit. This is a "weighted" code in that a particular value is assigned to a "1" depending on the order in which it appears. (3). For this code, the weights of the orders, in descending sequence, are 8-4-2-1. For this reason, it is referred to as the "8-4-2-1" code. (3). In this code, the decimal digit equivalents are

0	----	0000
1	----	0001
2	----	0010
3	----	0011
4	----	0100
5	----	0101
6	----	0110
7	----	0111
8	----	1000
9	----	1001

There are many other codes, both weighted and nonweighted. For more extensive information regarding decimal codes, the reader is referred to Richards. (3).

The "excess 3" code has been chosen for use in this machine. Justification of this choice will follow the description of the code. In this code, the binary group is the equivalent of three more than the digit being represented. These groups are

0	----	0011
1	----	0100
2	----	0101
3	----	0110
4	----	0111
5	----	1000
6	----	1001
7	----	1010
8	----	1011
9	----	1100

All of the possible sums of two digits represented by excess 3 code groups, including a carry, are presented in Table I.

TABLE I

ADDITION TABLE FOR EXCESS 3 CODE

Decimal Sum	Uncorrected		Corrected		Correction Required
	Carry	Sum	Carry	Sum	
0	0	0110	0	0011	Subtract 3
1	0	0111	0	0100	
2	0	1000	0	0101	
3	0	1001	0	0110	
4	0	1010	0	0111	
5	0	1011	0	1000	
6	0	1100	0	1001	
7	0	1101	0	1010	
8	0	1110	0	1011	
9	0	1111	0	1100	
10	1	0000	1	0011	Add 3
11	1	0001	1	0100	
12	1	0010	1	0101	
13	1	0011	1	0110	
14	1	0100	1	0111	
15	1	0101	1	1000	
16	1	0110	1	1001	
17	1	0111	1	1010	
18	1	1000	1	1011	
19	1	1001	1	1100	

Consider, for example, the addition of 7 and 6.

```
 7  ----  1010
 6  ----  1001
13  ----  0011 plus 1 carry
```

The required tens' carry is correctly propagated at the proper time, from the highest order, for all cases. This is clear from Table I. However, it becomes necessary to correct the 0011 to 0110 in order that it will be recognized as a 3 in the excess 3 code. The required corrections for all sums are indicated in the table. The automatic propagation of the tens' carry is one important reason for choosing the excess 3 code. Before the problem of correcting the sums of Table I is considered, the matter of coding and decoding shall be studied.

The methods of Boolean algebra are to be used for this analysis. (2). The orders of the excess 3 code are to be represented by the letters A, B, C, and D, in descending sequence.

Thus,

0	0011	$\bar{A}\bar{B}CD$
1	0100	$\bar{A}B\bar{C}\bar{D}$
2	0101	$\bar{A}B\bar{C}D$
3	0110	$\bar{A}BC\bar{D}$
4	0111	$\bar{A}BCD$
5	1000	$A\bar{B}\bar{C}\bar{D}$
6	1001	$A\bar{B}\bar{C}D$
7	1010	$A\bar{B}C\bar{D}$
8	1011	$A\bar{B}CD$
9	1100	$AB\bar{C}\bar{D}$

It will be noted that these are in minterm, or canonical, form; that is, each group is expressed as a Boolean product containing all elements of the class. These minterms may be represented as squares on a Veitch diagram. (2). The numbers in the minterm squares correspond to their decimal counterparts. Redundant minterms are denoted by X's.

	A	A	$\bar{A}$	$\bar{A}$	
B	9	X	3	1	$\bar{D}$
B	X	X	4	2	D
$\bar{B}$	6	8	0	X	D
$\bar{B}$	5	7	X	X	$\bar{D}$
	$\bar{C}$	C	C	$\bar{C}$	

The simplest Boolean expressions for each digit may be written directly from the Veitch diagram, taking into account all redundancies.

0	$\bar{A}\bar{B}$
1	$\bar{A}\bar{C}\bar{D}$
2	$\bar{A}\bar{C}D + B\bar{C}D$
3	$\bar{A}C\bar{D} + BC\bar{D}$
4	BCD
5	$\bar{B}\bar{C}\bar{D}$
6	$A\bar{C}D + \bar{B}\bar{C}D$
7	$ACD + \bar{B}C\bar{D}$
8	ACD
9	AB

A similar Veitch diagram for the 8-4-2-1 code may also be made.

0	----	0000	----	$\bar{A}\bar{B}\bar{C}\bar{D}$
1	----	0001	----	$\bar{A}\bar{B}\bar{C}D$
2	----	0010	----	$\bar{A}\bar{B}C\bar{D}$
3	----	0011	----	$\bar{A}\bar{B}CD$
4	----	0100	----	$\bar{A}B\bar{C}\bar{D}$
5	----	0101	----	$\bar{A}B\bar{C}D$
6	----	0110	----	$\bar{A}BC\bar{D}$
7	----	0111	----	$\bar{A}BCD$
8	----	1000	----	$A\bar{B}\bar{C}\bar{D}$
9	----	1001	----	$A\bar{B}\bar{C}D$

	A	A	$\bar{A}$	$\bar{A}$	
B	☒	☒	6	4	$\bar{D}$
B	☒	☒	7	5	D
$\bar{B}$	9	☒	3	1	D
$\bar{B}$	8	☒	2	0	$\bar{D}$
	$\bar{C}$	C	C	$\bar{C}$	

The simplest Boolean expressions, including redundancies, may again be written.

0	----	$\bar{A}\bar{B}\bar{C}\bar{D}$
1	----	$\bar{A}\bar{B}\bar{C}D$
2	----	$\bar{B}C\bar{D}$
3	----	$\bar{B}CD$
4	----	$B\bar{C}\bar{D}$
5	----	$B\bar{C}D$
6	----	$BC\bar{D}$
7	----	BCD
8	----	$A\bar{B}\bar{D}+A\bar{C}\bar{D}$
9	----	AD

A comparison between this list of simplest Boolean terms and that for the excess 3 code reveals that those for the excess 3 code are simpler. Because of this, decoding will be easier to achieve. In fact, the Veitch diagram pattern of the excess 3 code makes the fullest possible use of the six redundant minterms. No other progressive pattern can make possible a simpler list of Boolean expressions. This fact, together with the self-propagation of the tens' carry when adding, has been the principle factor in the choice of the excess 3 code.

Addition

As pointed out in the discussion of the excess 3 code, the sums produced by adding must be corrected. If there is no carry resulting from a given addition, the correction is effected by subtracting 3. If a carry does exist, the correction may be achieved by adding 3. One method for subtracting 3 is to add 13 and subtract 16. The subtraction of 16 may be done by ignoring the carry which results from adding 13, because this has a value of 16. In adding either 3 or 13, inversion of the lowest order is required since both numbers are odd. The corrections may be made by means of the configuration shown in Figure 2. (3). In this arrangement, one is added to the sum by adding a 1 in the D order. If a carry exists, then two is also added by adding 1 in the C order. If a carry does not exist, then 12 is also added by adding 1's in the A and B orders. The method of Figure 2 shall be used in the machine.

The adding elements of Figure 2 are referred to as binary "full adders." (3). A full adder has three inputs which accomodate the two bits to be added plus an input carry bit from a lower order. The full adder has both sum and carry outputs. The operation of a full adder may be demonstrated by letting

X ---- augend
Y ---- addend
C ---- input carry, from lower order

A truth table may be constructed which illustrates all possible combinations of inputs with their associated outputs.

X Y C	Sum	Carry
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

Separate Veitch diagrams may be drawn for the sum and carry.

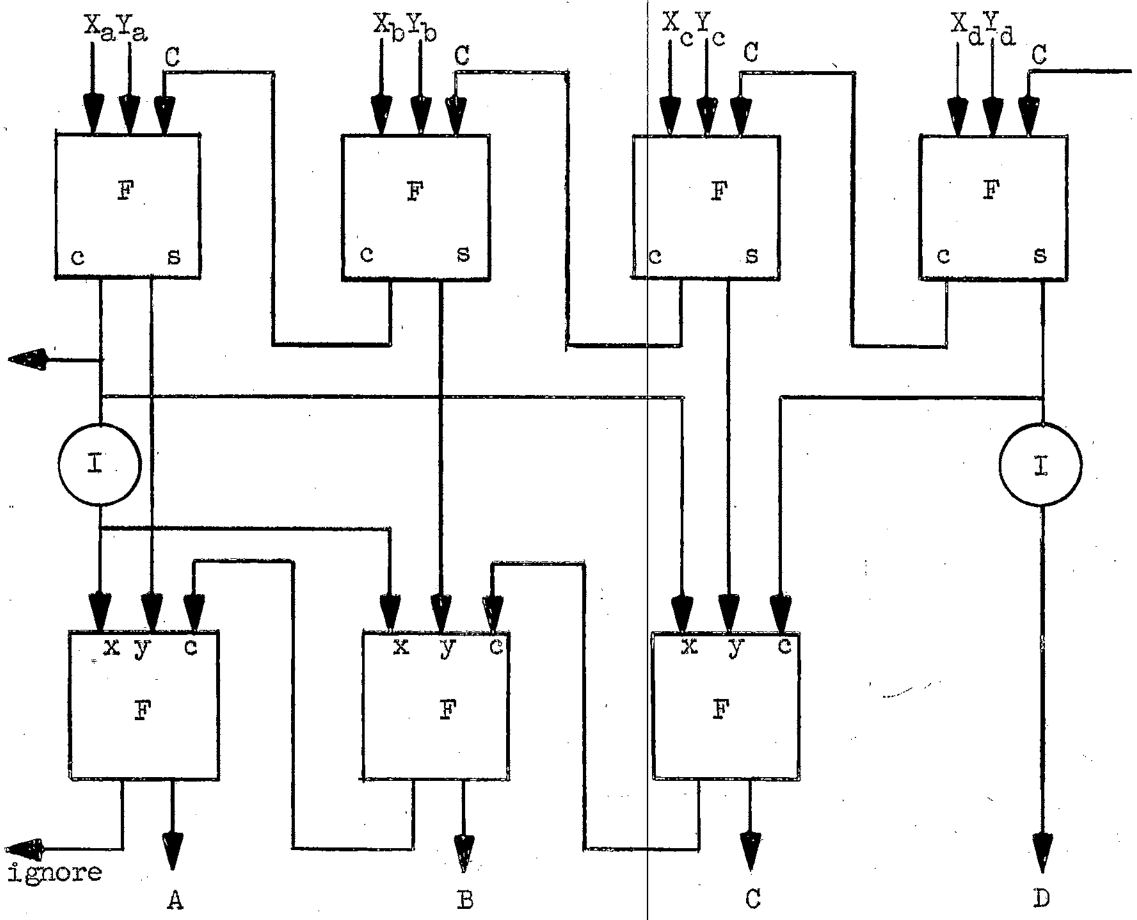


Figure 2

Excess 3 Adder

	X	X	$\bar{X}$	$\bar{X}$	
Y		1		1	Sum
$\bar{Y}$	1		1		
	$\bar{C}$	C	C	$\bar{C}$	

	X	X	$\bar{X}$	$\bar{X}$	
Y	1	1	1		Carry
$\bar{Y}$		1			
	$\bar{C}$	C	C	$\bar{C}$	

From these Veitch diagrams, the sum and carry expressions may be formed as

$$\text{sum} = \bar{X}\bar{Y}C + \bar{X}Y\bar{C} + X\bar{Y}\bar{C} + XYC$$

$$\text{carry} = XY + XC + YC$$

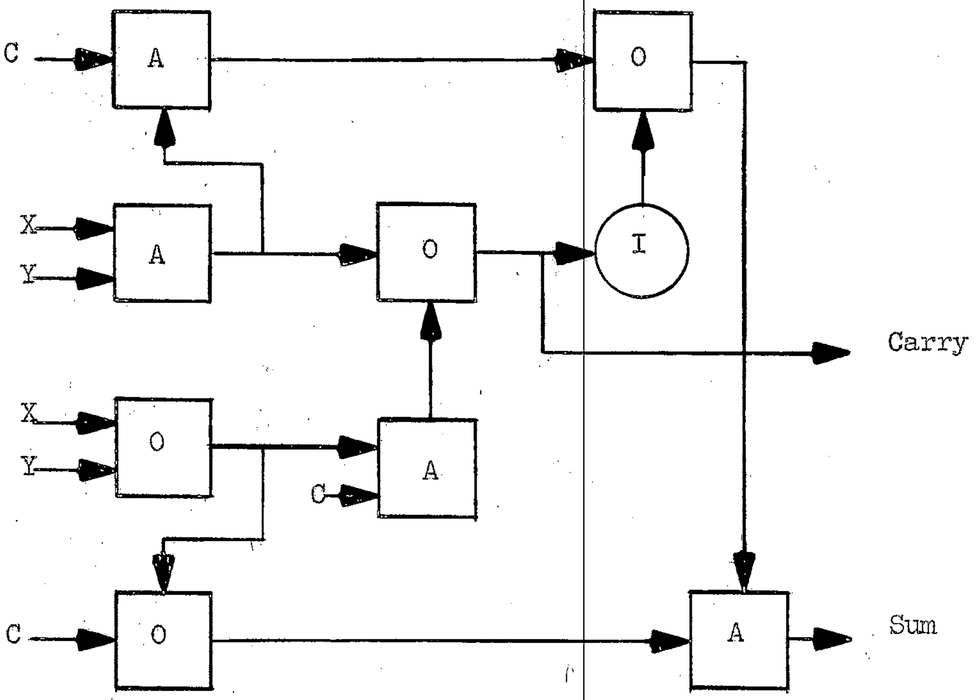


Figure 3

Binary Full Adder

The sum expression may be factored to

$$\text{sum} = (X+Y+C) \quad (\bar{X}\bar{Y}+\bar{X}\bar{C}+\bar{Y}\bar{C})+XYC$$

$$= (X+Y+C) \quad (\overline{XY+XC+YC})+XYC$$

$$= (X+Y+C) \quad (\overline{XY+XC+YC}+XYC)$$

$$= \left[(X+Y)+C\right] \quad \left[\overline{XY+(X+Y)C}+(XY)C\right]$$

The carry may be factored as

$$\text{carry} = XY+(X+Y)C$$

These expressions are presented in Richards (3) and are the bases for the logic diagram of the full adder of Figure 3.

Coding

Because digits are entered from a decimal keyboard, there must be a conversion made from decimal form to the excess 3 code. The coder can

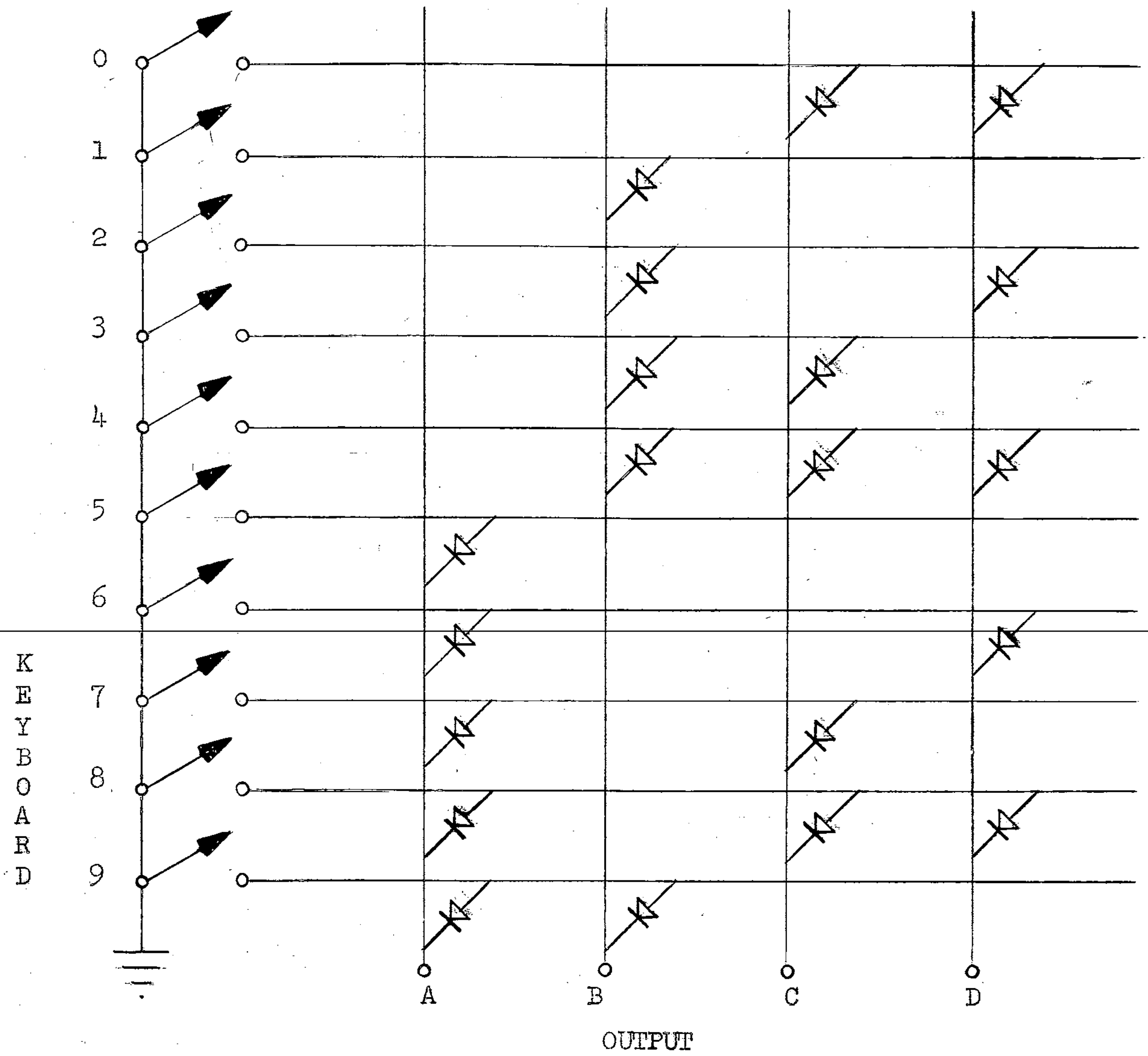


Figure 4. Coder

be formed from a diode matrix having ten input lines, one for each digit key, and four output lines, one for each order of the binary code group. The function of the coder is that of triggering the flip-flops of the addend register to correspond to the digits of the correct code group. The diode matrix of Figure 4 is arranged so that this desirability is fulfilled. (4). The output lines of the coder will be routed to negative voltages appearing at the collectors of the "off" transistors of the addend register. By referring to the list of the excess 3 code groups, it may be seen that the matrix diode positions correspond to the 1's of the code groups. Because of the fact that entries are made one digit at a time, only one coder is required.

Decoding

Decoding is also accomplished through the use of a diode matrix. The list of simplest expressions shows that for the 2, 3, 6, and 7 there exists an "or" choice of simplest terms. For purposes of decoding, the first listed of the "or" Boolean sums, for each case, shall be used. In order to uniquely identify each digit, it is seen from the list of simplest expressions that both states of each of the four orders are required. Hence, eight input lines will be needed. There are connected from the 0 and 1 outputs of the accumulator flip-flops. The diode matrix equivalent of the list of simplest terms is shown in Figure 5. Every possible combination of input levels, excluding the never-occurring redundant combinations, will uniquely identify the corresponding decimal digits. For example, if the accumulator digit were decimal 8

8 ---- 1011 ---- $A\bar{\bar{B}}CD$

simplest expression, due to redundancy of ABCD, ---- ACD

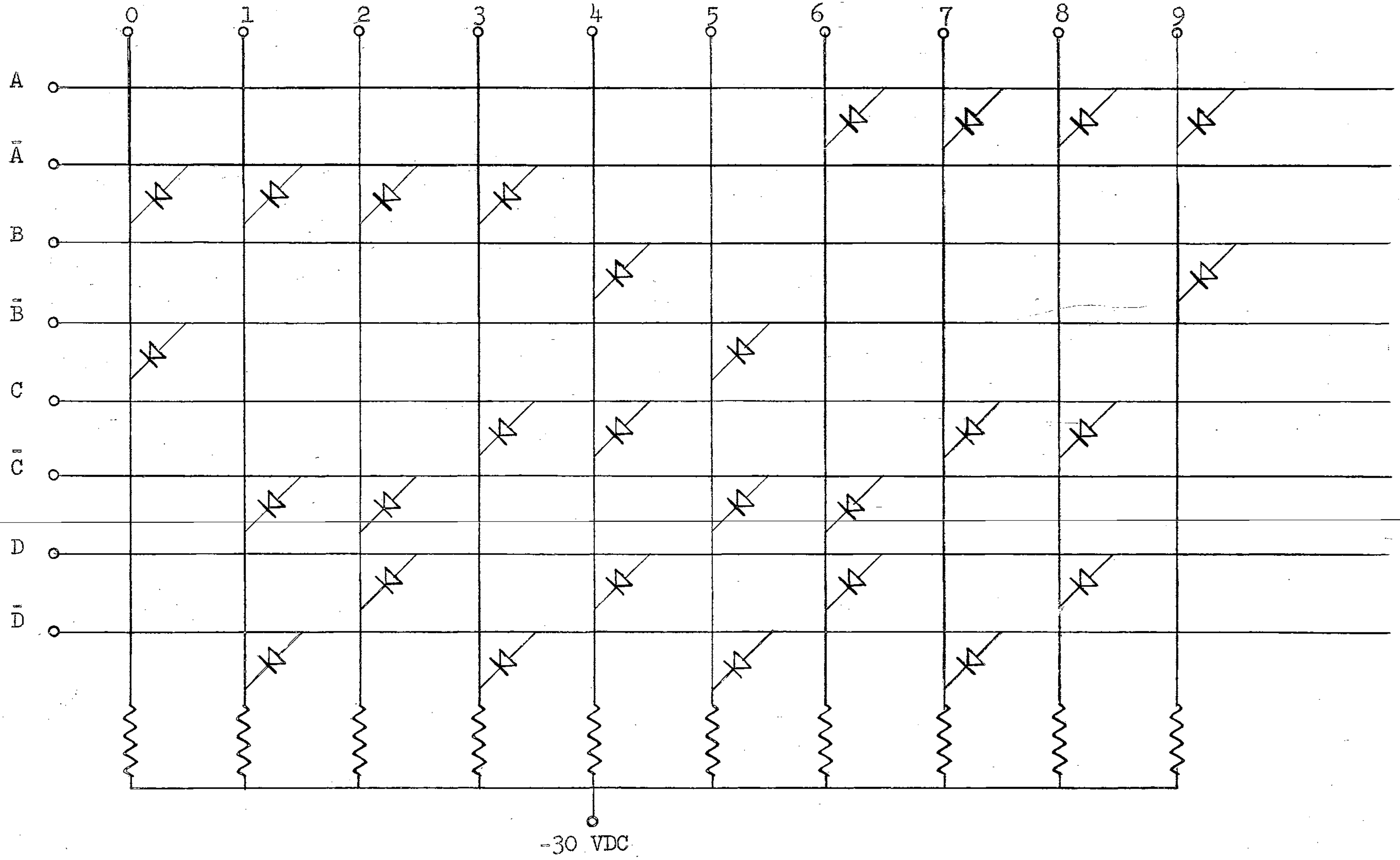


Figure 5. Decoder

For this case, 1's would be present on lines A, $\bar{B}$, C, and D. However, only 1's on lines A, C, and D would be used to derive the 8 output. This results in the saving of one diode. Because the readout is presented as four simultaneously appearing decimal digits, four decoders are required.

Registers

The two registers are temporary storage units for the addend and augend of a given addition operation. Since the digits are being added in four-bit parallel form, each register must be capable of storing a four-bit code group. As previously proposed, flip-flops are to be used as storage devices; therefore, four flip-flops are required in each register. These flip-flops must have "set" and "reset" provisions. (3). The registers must be reset to zero prior to a given operation. The basic register is shown in Figure 6. The transitory code group from the keyboard decoder, or from the accumulator, is fed to the set lines A, B, C, and D. The flip-flops are thereby "latched" in the correct states for representing the trigger code group.

Accumulator

The accumulator is the main memory section of the machine. Its function is to store, and make available for readout, each sum that is produced by the adder circuits. This function could be carried out by storage of sum code groups, or by storage of the decoded decimal sums. Again, because of the use of flip-flops as storage devices, the storage will be done on the binary level by storing the code groups.

It was originally specified that the machine should have a four

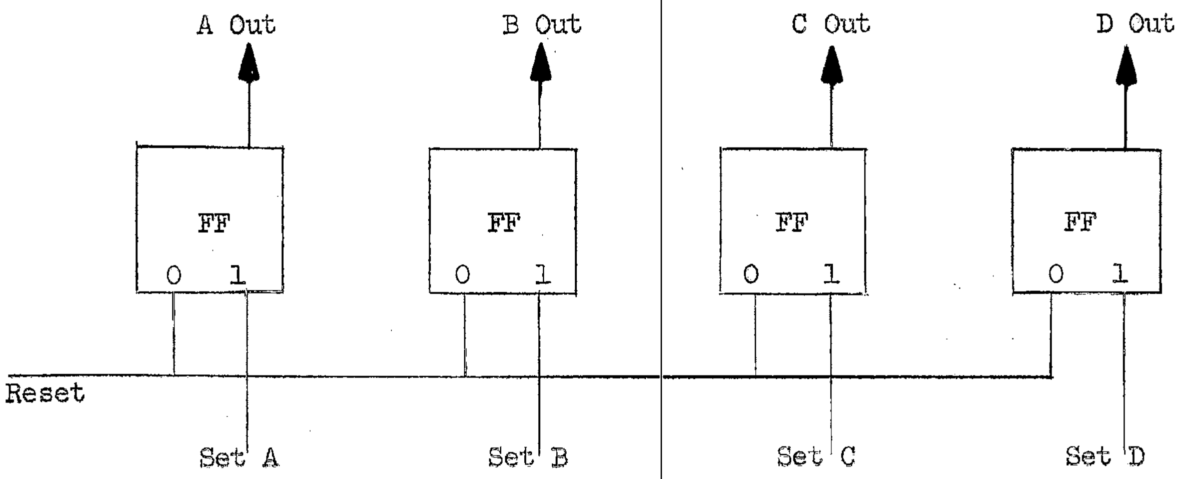


Figure 6. Basic Register

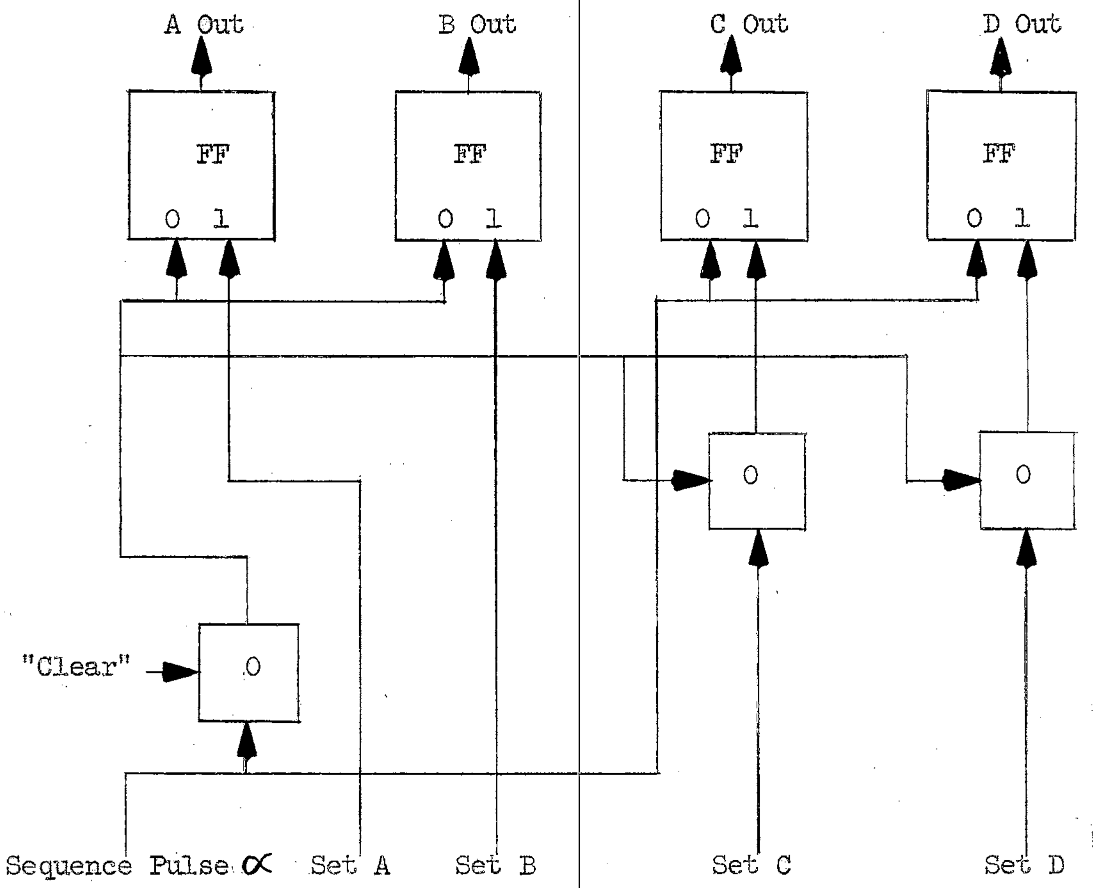


Figure 7. Accumulator Section

decimal digit capacity. This necessitates having a four-section accumulator, with each section capable of storing a four-bit code group. Each section of the accumulator will then be similar to the registers and will consist of a similar arrangement of flip-flops. One noteworthy difference is that, because the first augend of any series of additions is to be zero, each accumulator section must be capable of being set to zero, which is 0011. However, in order to accept sums, it is necessary that the accumulator be reset to 0000. This is done after the augend register has been set to 0011, but before the sum has been presented to the accumulator. An explanation of the control operations appears as a subsequent portion of the thesis. The provisions for meeting the set and reset requirements are shown in Figure 7.

Control

A digit control scheme is required to produce gating pulses for feeding the sum from a given addition to the correct section of the accumulator. Consider, for example, the addition of 7132 and 2061.

$$\begin{array}{lll} 7132 & ---- & \text{augend} \\ \underline{2061} & \underline{----} & \underline{\text{addend}} \\ 9193 & ---- & \text{sum} \end{array}$$

$$\begin{array}{cccccl} 9 & 1 & 9 & 3 & ---- & \text{accumulator} \\ A & B & C & D & & \end{array}$$

The addition would begin in the D order and proceed, one digit at a time. The accumulator would receive the sum in the sequence D, C, B, and A. Assume, for the time being, that negative gating pulses are to be used. Then the digit control pulses required would be those of Figure 8.

A method for producing these gating pulses is presented in Figure 9. The time chart of Figure 10 describes this method in detail.

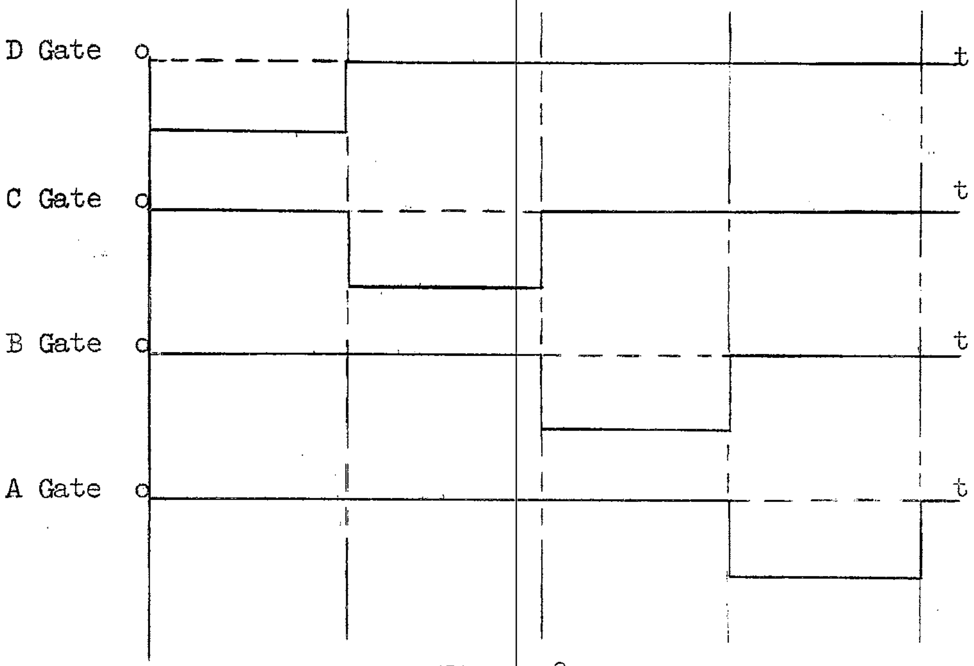


Figure 8

Digit Control Pulses

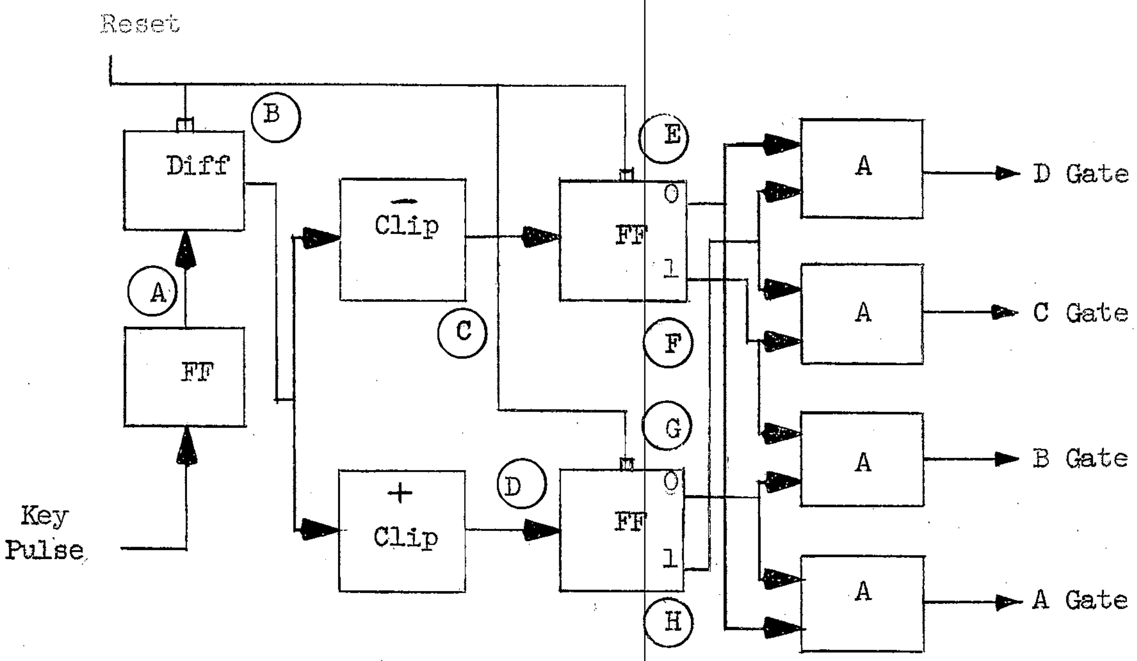


Figure 9

Generation of Digit Control Pulses

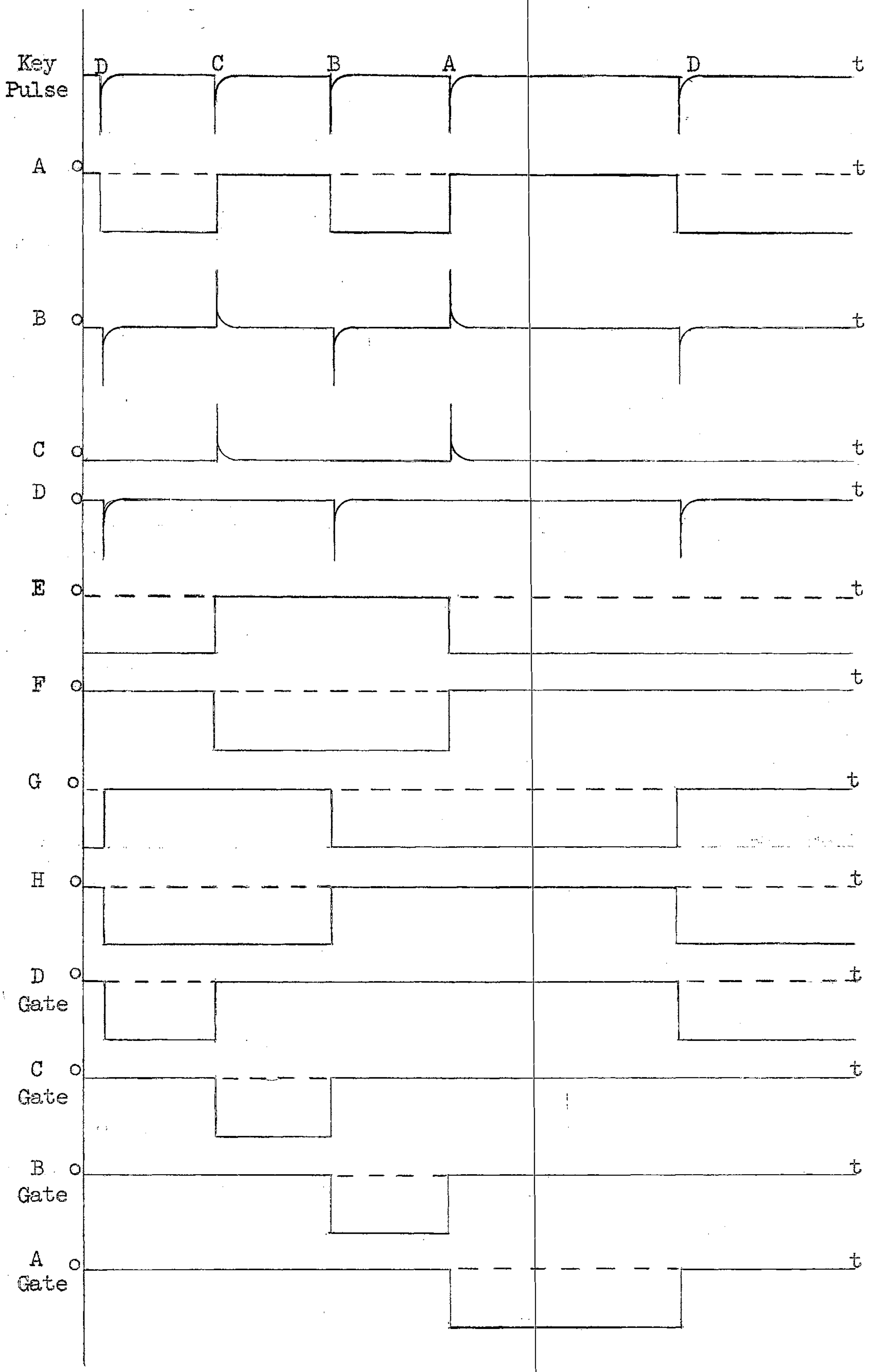


Figure 10. Digit Control Time Chart

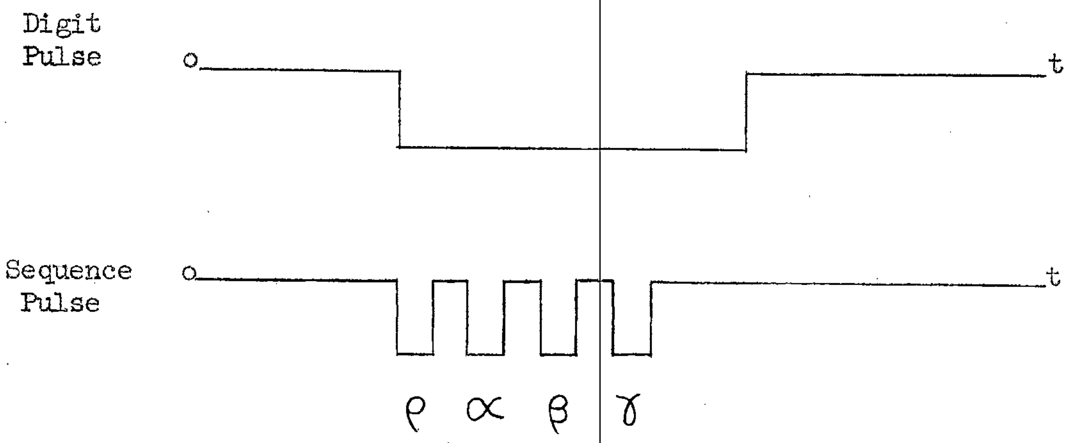


Figure 11

Relationship Between Digit and Sequence Pulses

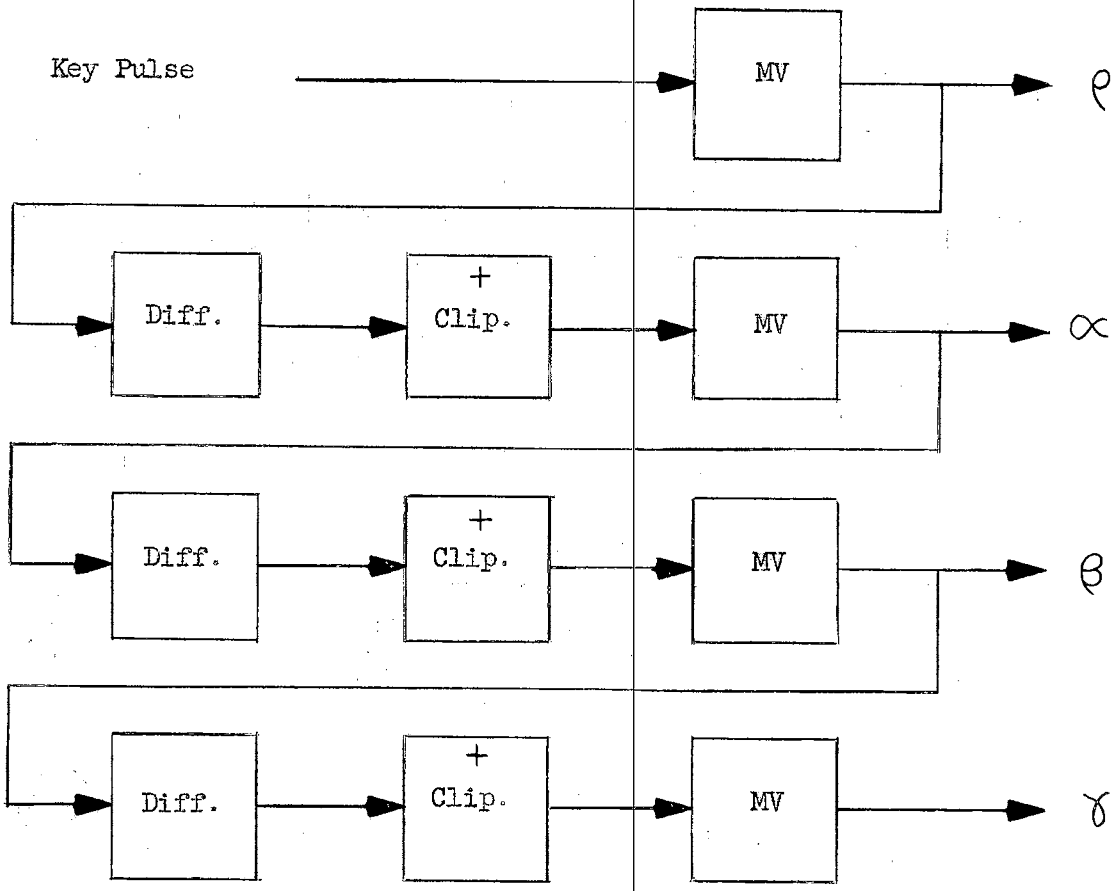


Figure 12. Method for Deriving Sequence Pulses

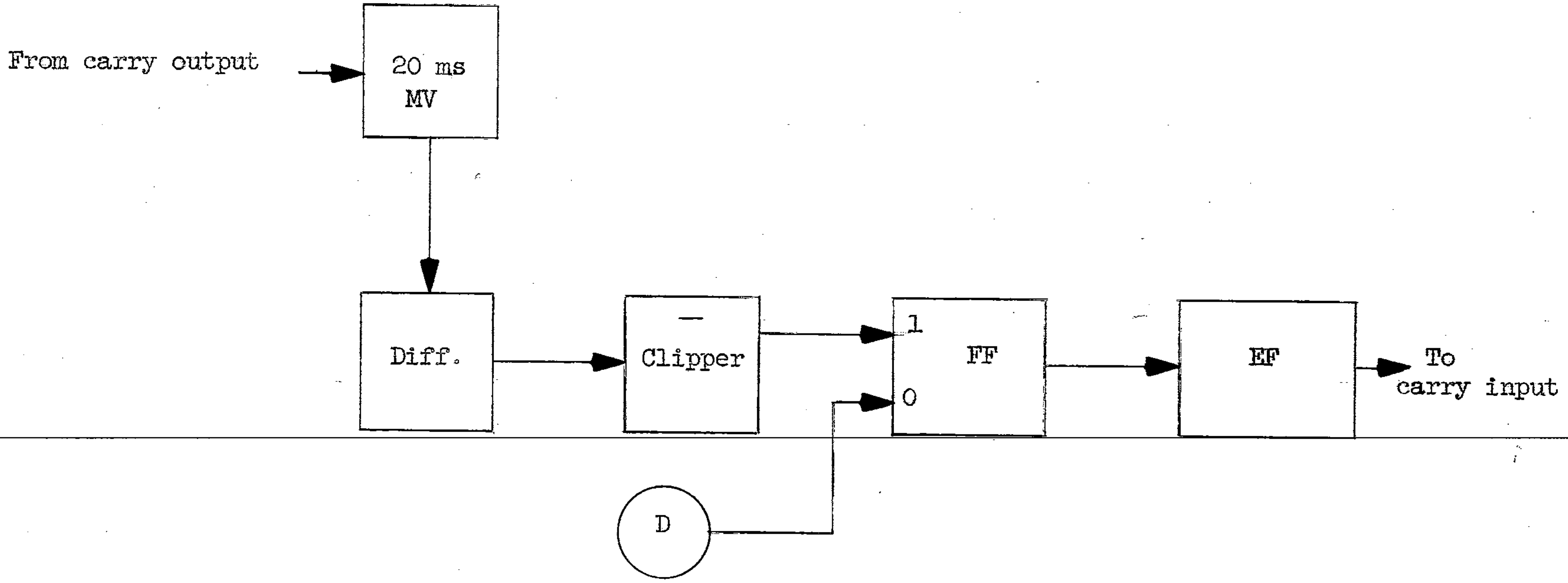


Figure 13. Carry Control

The encircled letters of Figure 9 correspond to those of Figure 10 and are used to associate each waveform with its location in the block diagram of Figure 9.

A sequence control is required to provide for the execution of the "housekeeping" operations in their proper order. This sequence is repeated for each digit:

1. Set augend register to accumulator value ---- ρ
2. Reset accumulator to 0000 ---- α
3. Add contents of registers ---- β
4. Reset registers to 0000 ---- γ

The relationship between these gating pulses and a digit gating pulse is shown in Figure 11. A method of deriving the sequence pulses is shown in Figure 12.

The carry control, Figure 13, consists of a 20 ms delay of the output carry so that the last three sequence pulses will have time to be completed prior to when the carry is presented to the adder for the next addition. In the event of a D order digit pulse, the carry storage flip-flop is reset to 0. This precludes the possibility of having an input carry to the adder when adding in the lowest order.

There are two control keys in addition to the ten digit keys. The "clear" key resets all register flip-flops to 0's and sets the accumulator to 0011, which is zero, the first augend required for beginning a series of additions.

The "add" key serves to notify the machine that the number being entered has been completed. This eliminates the need for entering superfluous high order zeros for numbers having less than four digits. The "add" key is to be depressed following each complete number entry.

The remainder of the logical design consists of interconnecting the various sections of the machine already described. Specific verbal detail will not be given, but reference is made to Plate I, Appendix. This plate shows the complete interconnection of the logic and storage elements.

CHAPTER IV

CIRCUITS USED IN MODEL

"And" and "Or" Gates

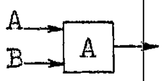
The Boolean "and" and the "or" are mechanized by diode gates as shown in Figure 14 and Figure 15, respectively. These provide a simple, economical, and dependable way for mechanizing "and" - "or" logic. (4).

The basic "and" gate, having inputs A and B, obeys the following Boolean equation and truth table:

$$\text{Output} = AB$$

A	B	Output
0	0	0
1	0	0
0	1	0
1	1	1

The presence of 1's on all input lines is required for a 1 output. The "and" gate is symbolized by

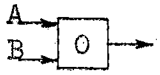


A basic "or" gate, having inputs A and B, obeys the following Boolean equation and truth table:

$$\text{Output} = A + B$$

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

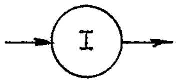
The presence of a 1 on any input line results in a 1 output. The "or" gate is symbolized by



The circuits of Figures 14 and 15 are for negative input signals. By interchanging the two circuits, the corresponding circuits for positive signals will result. For the logic of this machine, a 0 is represented by zero volts and a 1 by -25 volts. The E + supply voltage should be positive with respect to zero volts, and the E- supply voltage should be negative with respect to -25 volts. For convenience, the supply voltages of the model are + 25 and -30 VDC.

Inverter

The only purely logical element using a transistor, in this machine, is the inverter of Figure 16. The input voltage divider reduces the input level to one suitable for operation of the transistor. A 0 input turns the transistor "off" and yields the supply voltage as an output 1. An input 1 turns the transistor "on" and gives an output 0 of approximately zero volts. The inverter is symbolized by



Full Adder

The adding method described in the preceding chapter uses seven full adders. The full adder that was developed has been mechanized by using the "and," "or," and inverter elements just described.

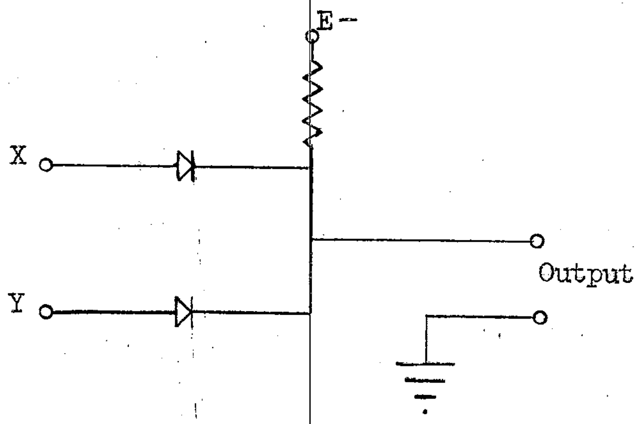


Figure 14. "And" Gate

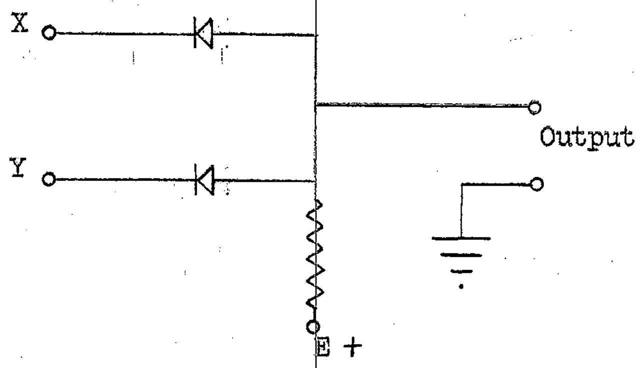


Figure 15. "Or" Gate

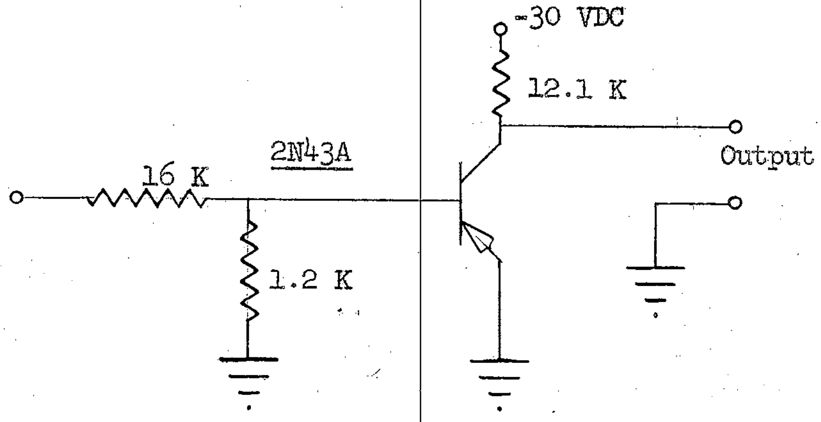


Figure 16. Inverter

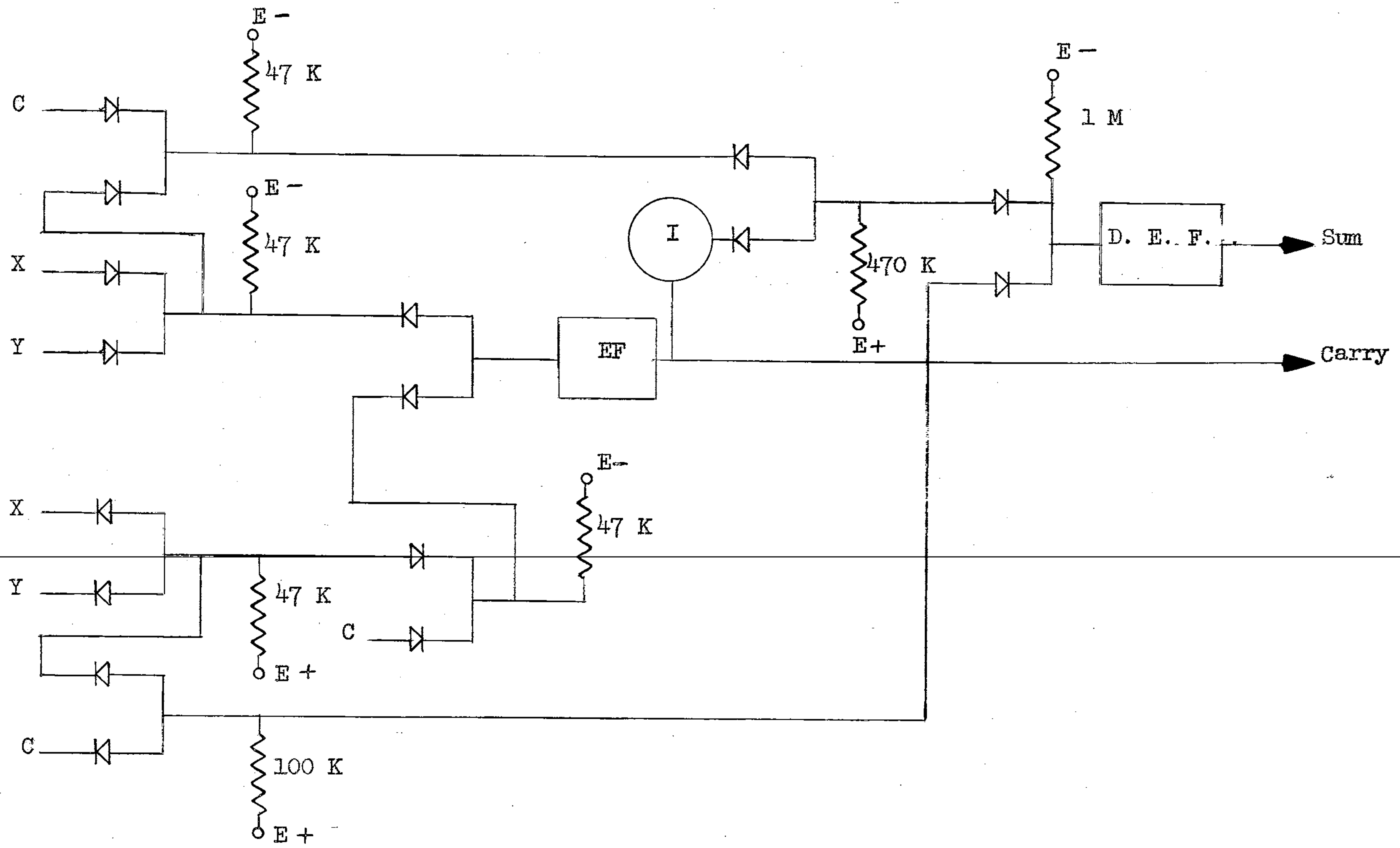
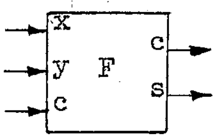


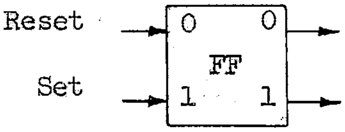
Figure 17. Full Adder Circuit

The resulting circuit diagram is shown in Figure 17. The full adder is symbolized by



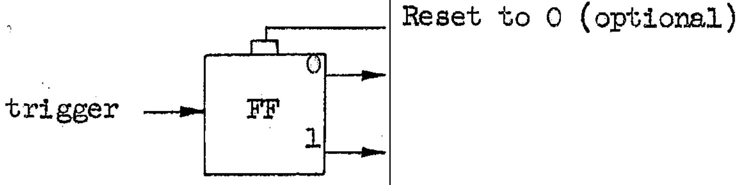
Flip-Flops

There are two basic trigger types of flip-flops used in this machine. The storage flip-flops of the registers and accumulator use the "reset - set" type. This type has two trigger inputs, one for setting the flip-flop to an output 1 and one for resetting to an output 0. This flip-flop may be represented, as by Richards (3),



Two means of actual triggering of this flip-flop are used. Collector triggering, as illustrated in Figure 18, consists of applying a positive trigger pulse, through an input diode, to the collector of the "off" transistor. (4). This positive pulse is applied, through the coupling network, to the base of the "on" transistor. The net result of these two conditions is a change of state of the flip-flop. Another variation is shown in Figure 19. By this method, a negative trigger pulse is applied to the base of the "off" transistor. This results in turning the transistor "on", thereby changing the state of the flip-flop. This last method is used for low-signal applications since it required less driving power from the trigger source. It will be noted that a voltage divider is used in the input trigger lines of Figure 19. This is to provide a method of discriminating against low voltage trigger "noise" which results from deteriorated 0's. For example, the output 0's of the excess 3 adder range from zero to -5 volts. The output 1's range from -15 to -30 volts. Therefore, the voltage divider is chosen so that the flip-flop will not trigger over the range of 0's, but will trigger dependably from the 1's. This difference in amplitude is sufficient to insure dependable operation.

The toggle flip-flop, Figure 20, is used for counting applications. (4). Here it is used for counting key pulses in the digit control section. For every trigger pulse on the input trigger line, the flip-flop undergoes a change of state. Then, differentiation of the output results in pulses that are decreased in frequency by a scale of two. This flip-flop may be symbolized by



Monostable Multivibrator

The monostable multivibrator is a special case of a flip-flop having one stable state. (4). When an input trigger pulse is applied, the circuit makes a transition to the other state and remains there for a predetermined length of time before returning to the stable state. The differentiated output then results in the delay of the trigger pulse by this predetermined length of time. In this machine, this circuit is used as a time delay. The circuit of Figure 21 gives a delay of approximately 5 ms. A change in circuit time constants gives the 20 ms delay used in the carry control circuit. The monostable

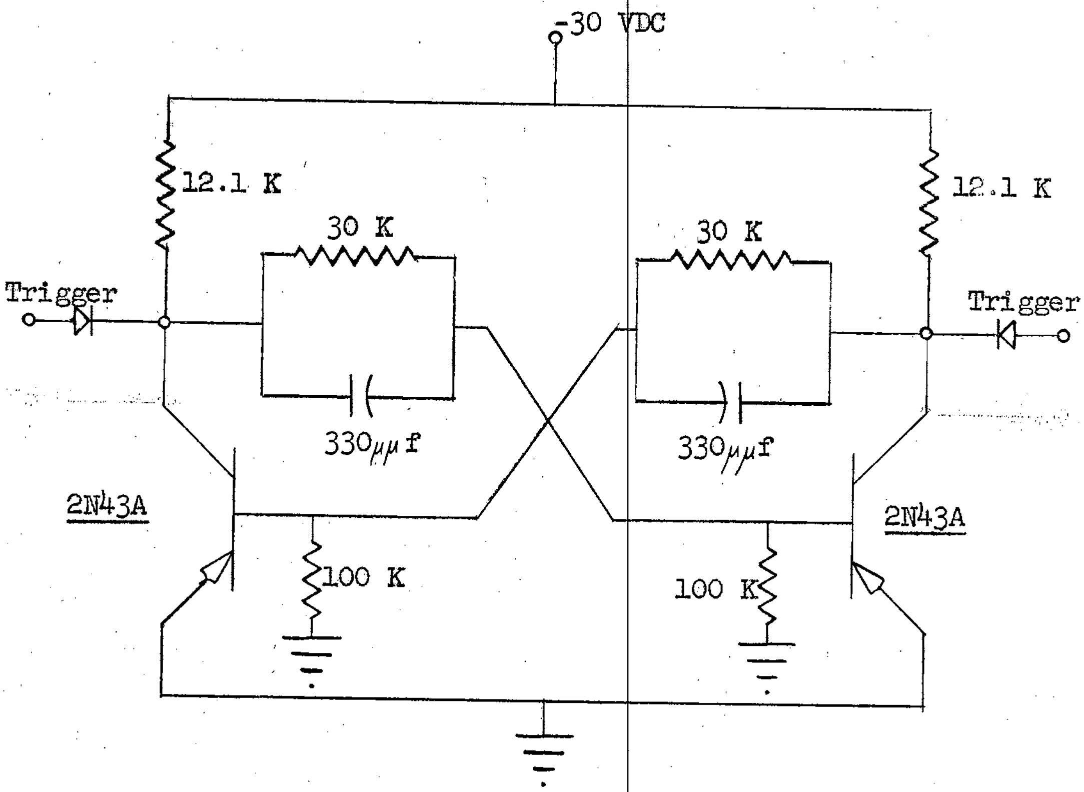


Figure 18. Flip-Flop, Collector Triggering

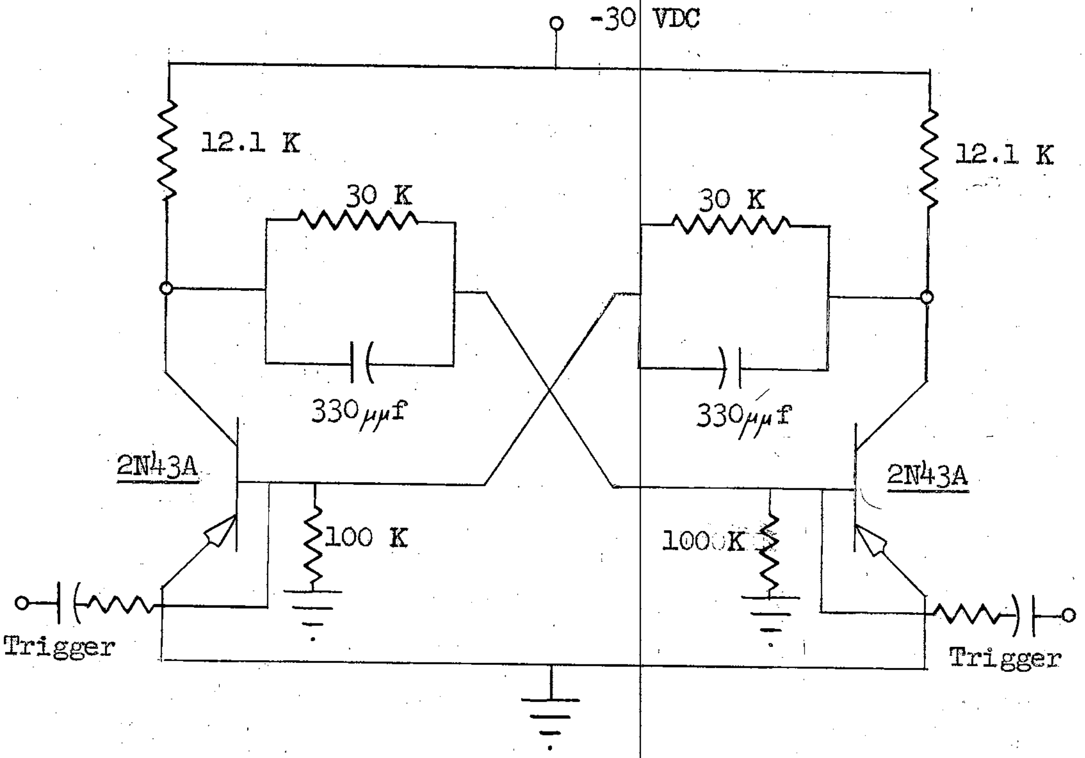


Figure 19. Flip-Flop, Base Triggering

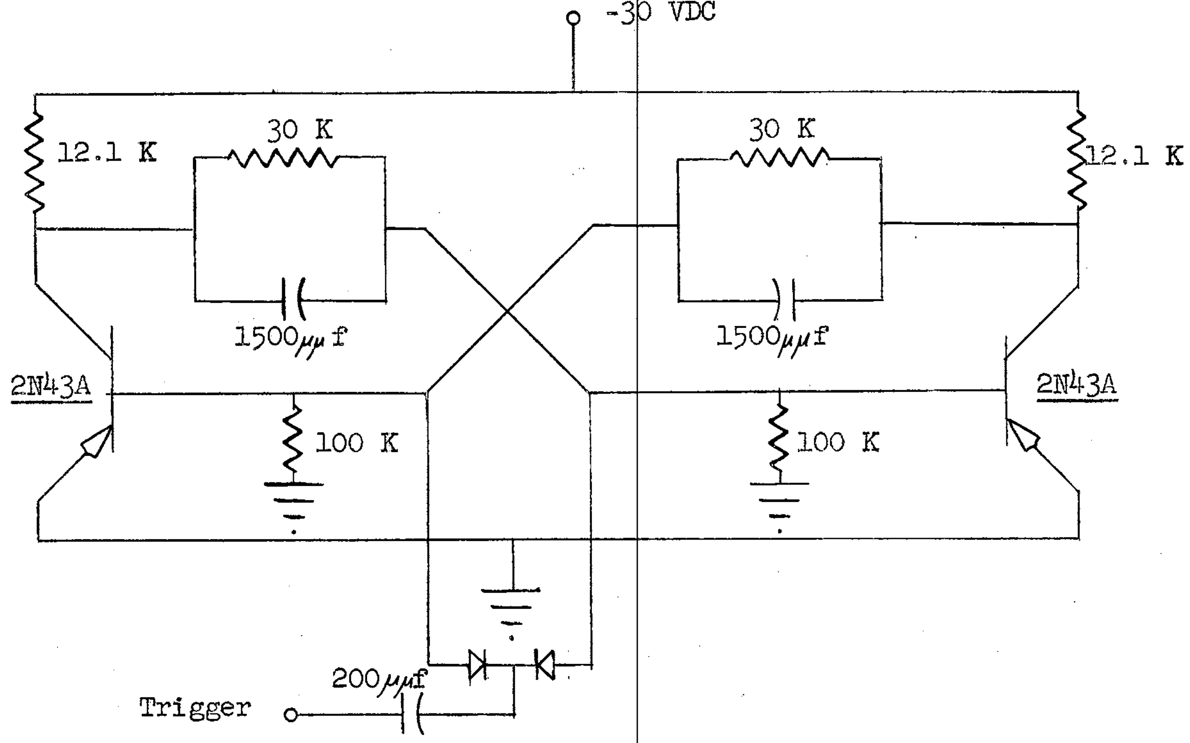


Figure 20. Toggle Flip-Flop

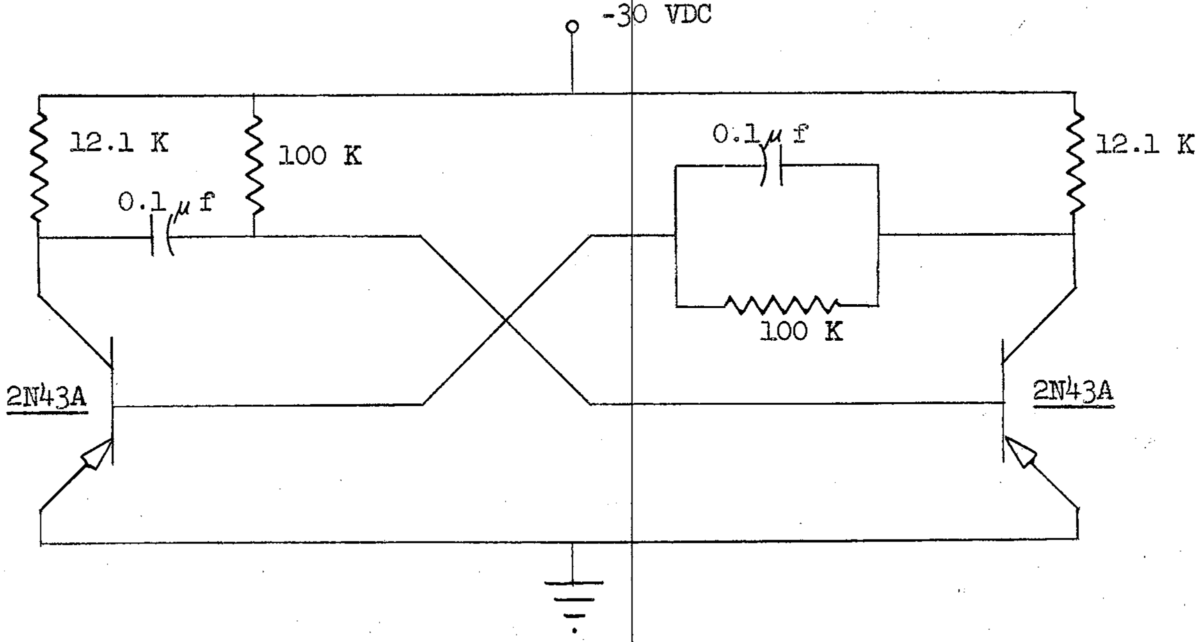
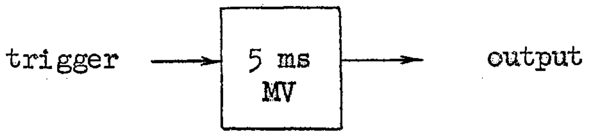


Figure 21. Monostable Multivibrator

multivibrator is symbolized by



where the delay time is included in the square block.

Emitter Followers

The emitter follower is used as a current amplifying device having a voltage gain of slightly less than unity. (5). The simplest of the emitter followers herein used is diagramed in Figure 22. The degenerative feedback of the circuit makes possible a wide range of input voltages.

The emitted follower of Figure 23 makes use of the Darlington connection. (5). This connection results in a much higher input impedance than the previous circuit. This circuit is used as a buffer into which the signals from the full adders are fed. The high impedance input is essential to keep from unduly loading the diode logic circuits of the full adder.

Readout

The "Nixie" decimal readout device, manufactured by the Burroughs Corporation, is used as the sum indicator. The tube is shown, diagramatically, in Figure 24. The tube has a common anode and is gas-filled. There are ten cathodes, and each cathode is in the shape of a different decimal digit. The digit to be displayed is selected by putting a negative 150 volts on the cathode corresponding to that digit. This results in ionization of the gas and causes the ions to collect around the cathode. This glow describes the digit to be displayed. The

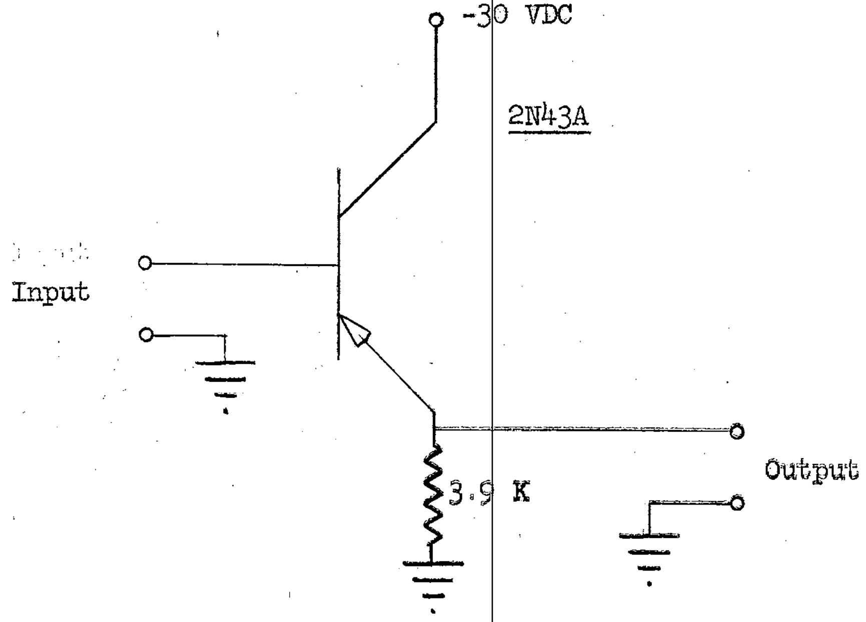


Figure 22. Emitter Follower

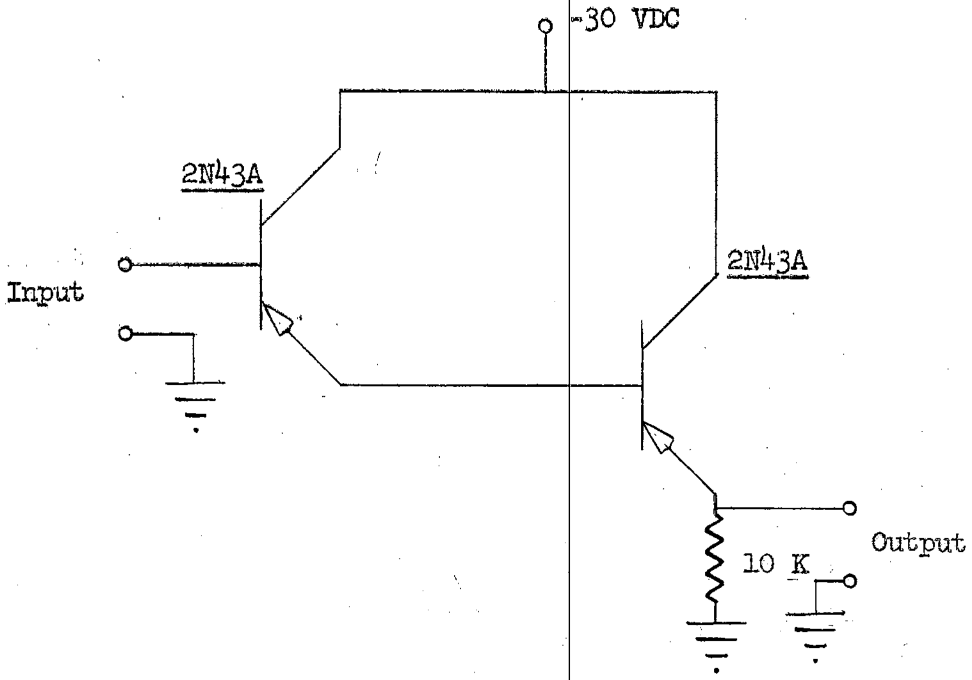


Figure 23. Darlington Emitter Follower

tube is constructed so that all cathodes are visible from the front of the tube. A bias of approximately + 125 volts is applied to the "Nixie" anode. The 1 from the decoding matrix is a negative voltage. When this is applied to the cathode, the ionization potential is reached, and the digit may be read.

Power Supplies

Three power supplies are required for the model:

(1) -30 VDC at .75A, for the transistor circuits and "and" gates

(2) +25 VDC at 100 ma, for the "or" gates

(3) +125 VDC at 5 ma, for the "Nixie" bias

Circuit diagrams of these power supplies are shown in Figure 25. Silicon diodes are used as the rectifying devices. Operation is from a conventional 117 VAC, 60 cycle source.

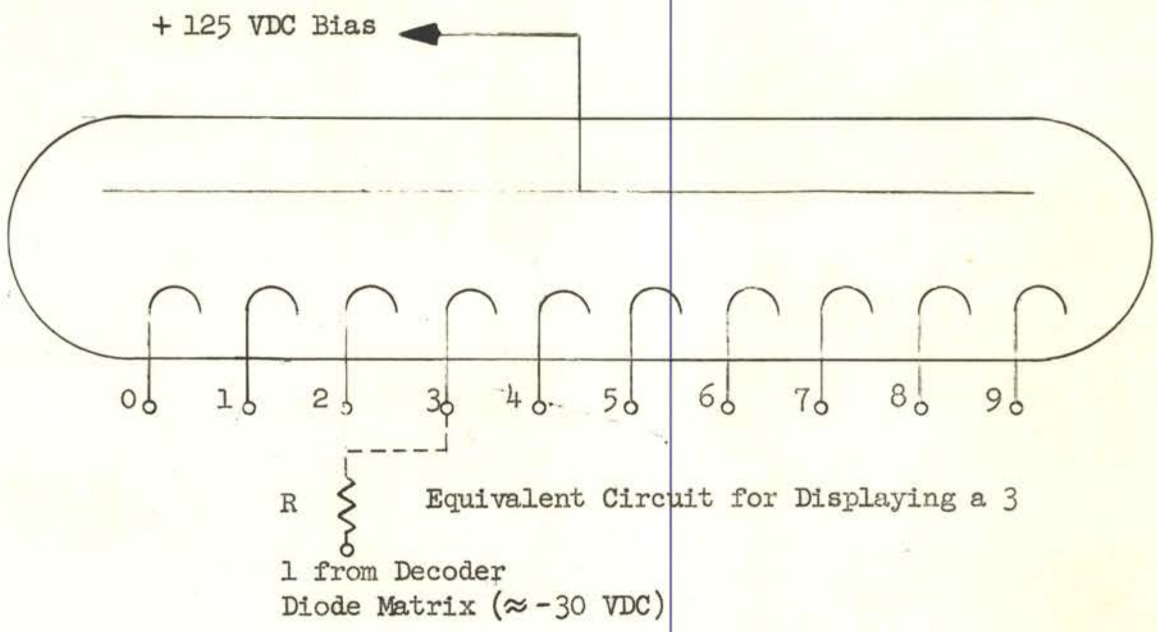


Figure 24. "Nixie" Readout Circuit

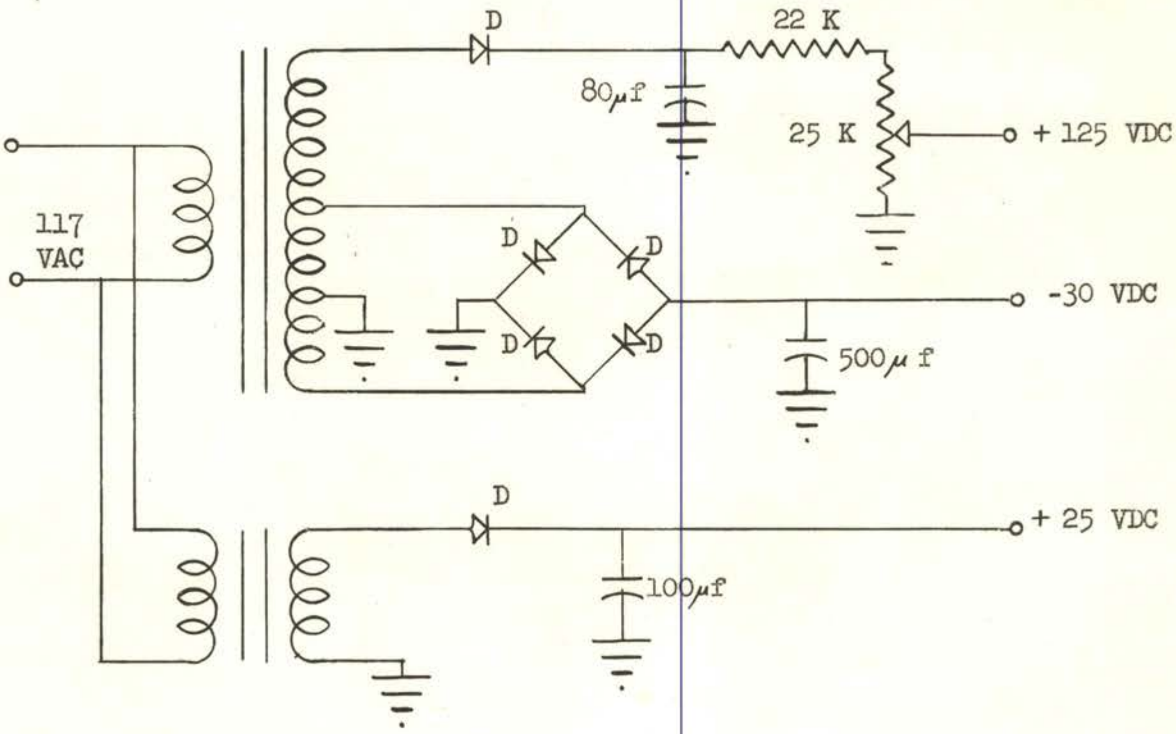


Figure 25. Power Supplies

CHAPTER V

OPERATION AND PERFORMANCE OF TEAM

It has been pointed out previously that the first number to be added in any series is to be added to an initial zero in the accumulator. Explanation has also been given as to the necessity, in this machine, for entering the digits of a number in reverse order. With this consideration in mind, the operation of the machine is as follows:

(1) All four digits of the accumulator are set initially to zero by depressing the "clear" key.

(2) The first number to be added is entered from the keyboard one digit at a time, the digits being entered in reverse order.

(3) At the end of each complete number, the "add" key is depressed.

(4) The next number to be added is entered in similar fashion, and so on for each other number to be added.

As an example, consider the addition of 379 and 1648. The sequence of keyboard operations required is: Clear -- 9 -- 7 -- 3 Add -- 8 -- 4 -- 6 -- 1 -- Add. This would cause the sum, 2027, to appear in the readout section.

The design of the sequence control circuits causes the actual operation of adding a digit to continue for approximately 20 ms. This is the time required for the four successive 5 ms delay circuits, of the sequence control, to operate. This means that digits could be entered from the keyboard at a maximum rate of 50 per second. In actual practice, it is doubtful that the rate of depressing any two consecutive keys would exceed the equivalent rate of 10 per second. Thus, the safety factor of five seems to be adequate for insuring sufficient adding time.

A photograph of the operational model is shown in Figure 26. The "Nixie" readout tubes are to the left of the keyboard. The "Clear" key is on the upper-left of the keyboard, and the "Add" key is on the lower-left.

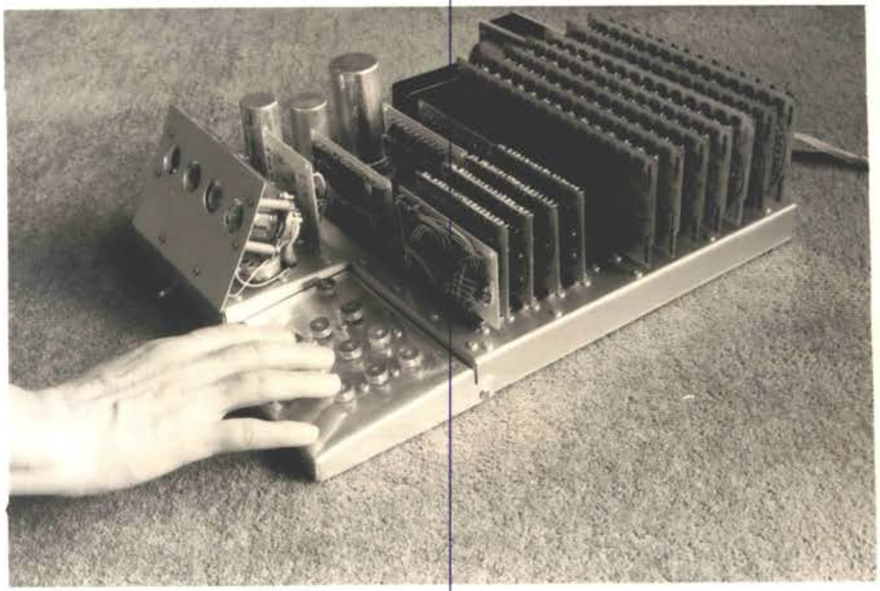


Figure 26

Photograph of Model

CHAPTER VI

SUMMARY AND CONCLUSIONS

The performance of the model has been entirely satisfactory and has borne out the validity of the logical design. There have been no unstable tendancies noted. The thesis, to this point, has dealth with the design of a four-digit machine. The model that was constructed departed from this design in the following ways:

(1) Only two digits were completed, this being sufficient for the stated purpose of the model.

(2) The sequence control was not constructed but was simulated by four pushbutton switches that were depressed, in order, after each digit had been entered. This, too, was considered to subtract little from the usefulness of the model.

(3) Because of (1), the digit control and the carry control sections were modified to serve two digits rather than four. These alternate control techniques are illustrated in the Appendix by Figures 27 and 28.

The model was constructed from components that were readily available. The particular "Nixie" tubes used for the readout circuit were designed to be used with higher switching voltages than those used in this machine. As a result, the readout is not as clear as might be desired. There are tubes designed for lower switching voltages that should be investigated in the event of further work in this area.

The application of transistor logic to a machine of this type would invite exploration. (6). It might also be possible to use gas-tube flip-flops as more economical storage devices, but the question of long-term reliability would possibly prevent this. (7).

The more recent applications of ferrite cores to storage and, more recently, to logic loom as important advances pertinent to a machine of this general type. (8). Extensive use of cores could possibly result in greater reliability, and a higher degree of compactness with greater storage capacity. An increase in the storage capacity would make possible entry of numbers in forward order, rather than reverse. This is explained in Chapter III. Another advancement that would warrant study for a compact computer is that of micro-modules. For example, flip-flops are being made on an experimental basis that have all components, including transistors, arranged on a tiny crystal wafer. The coming of the tunnel diode will, without doubt, create many new, compact computer circuits. With greater compactness of a machine having capabilities similar to this one, it follows that a machine of similar size could be constructed having far greater capabilities. The extension of capability to include subtraction, for example, by means of complimentary addition, would be very simple. (3). More complex capabilities could be included in a general purpose calculator.

There are many instances in which a particular type of equation must be solved, over and over, with different coefficients. Machines similar to this one could possibly find widespread use in the form of specific designs capable of solving specific problems. As an example, a machine to solve standard quadratic equations could be developed. There is perhaps enough demand for this particular operation to create a market for such a machine.

In conclusion, the results of this thesis project have augmented the feeling that electronic digital computer techniques can be successfully applied to the desk calculator field. Electronic desk calculators will, in all probability, soon make their debut on a commercial basis. The success of this endeavor, as with others, is vested in the ingenuity of its proponents and is limited primarily by human imagination.

BIBLIOGRAPHY

(1). *Automatic Control*, A Scientific American Book.
New York: Simon and Shuster, 1955.

(2). Phister, Montgomery, Jr. *Logical Design of Digital Computers*.
New York: Wiley, 1958

(3). Richards, R. K. *Arithmetic Operations in Digital Computers*.
Princeton, New Jersey: Van Nostrand, 1955.

(4). Millman, Jacob, and Herbert Taub. *Pulse and Digital Circuits*.
New York: McGraw - Hill, 1957.

(5). Shea, Richard F. *Transistor Circuit Engineering*.
New York: Wiley, 1952.

(6). *General Electric Transistor Manual*, Third Edition.
Syracuse: General Electric Company, 1958.

(7). Thomas, J H. "Computers and the Neon Lamp."
Radio Electronics, XXX (November, 1959), 111-18.

(8). Richards, R. K. *Digital Computer Components and Circuits*.
Princeton, New Jersey: Van Nostrand, 1957.

APPENDIX

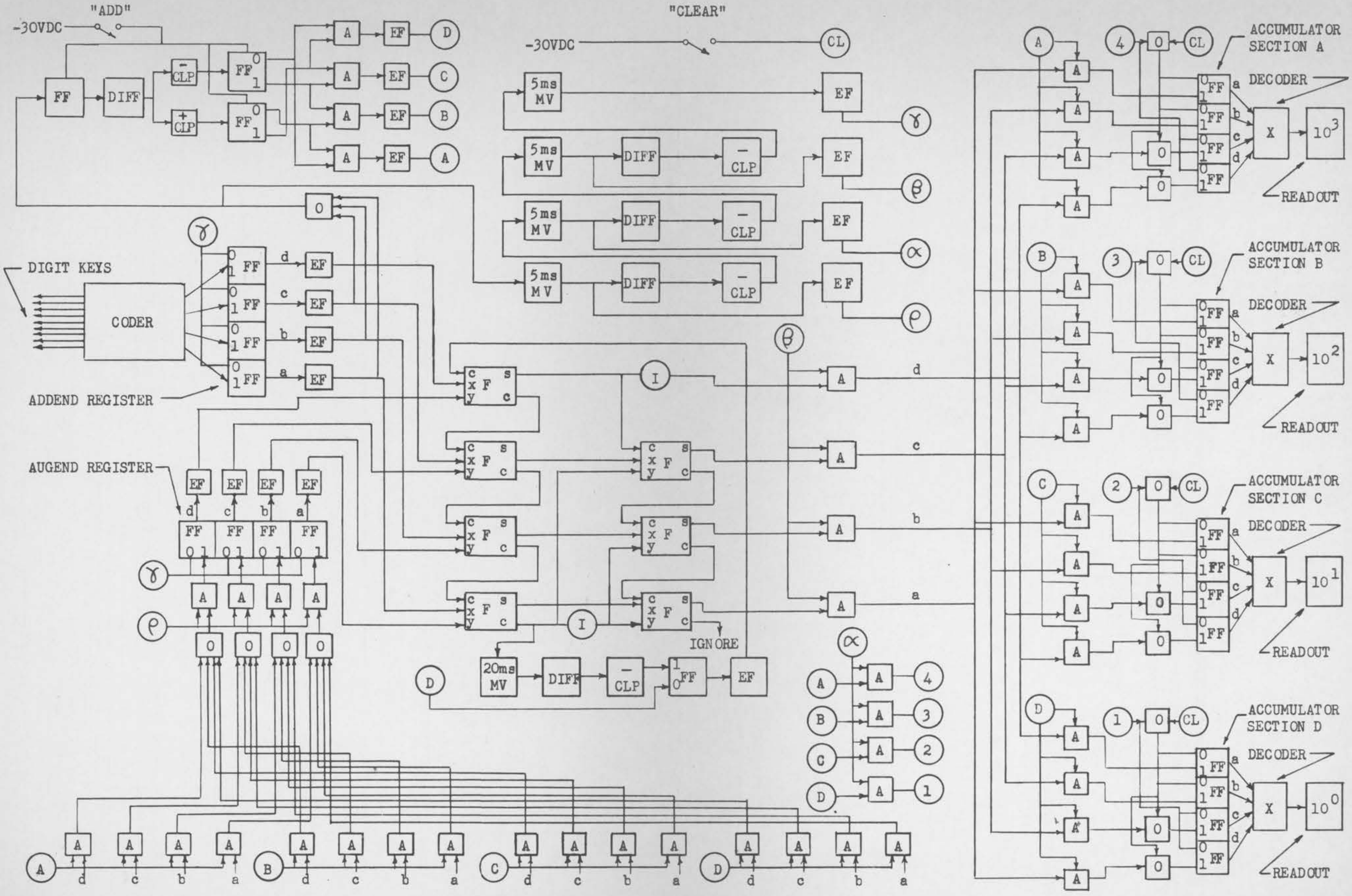


PLATE I. DETAILED OPERATIONAL DIAGRAM

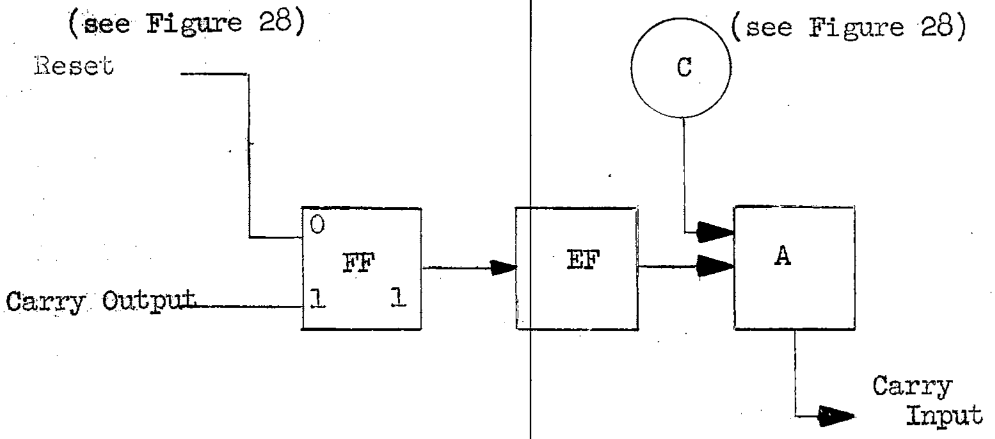


Figure 27. Alternate Carry Control For Two Digits

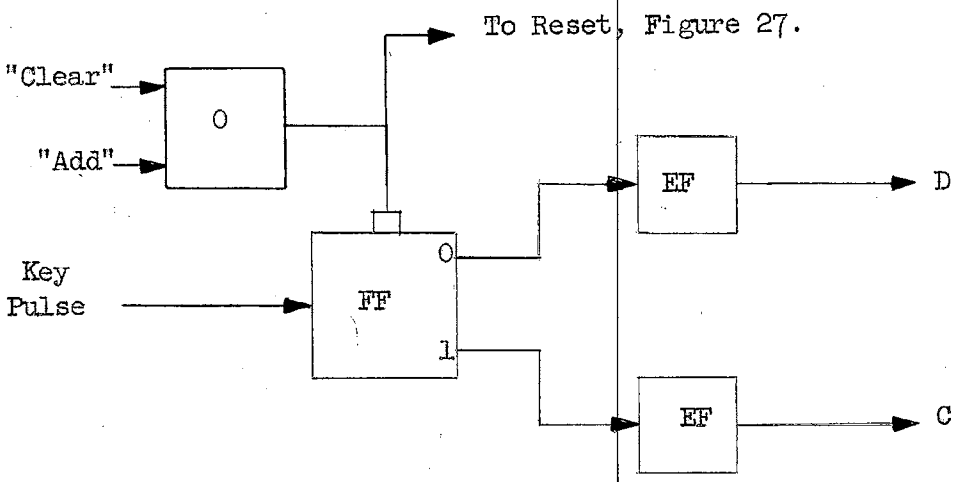


Figure 28. Alternate Digit Control for Two Digits

VITA

William Joseph Watson

Candidate for the Degree of

Master of Science

Thesis: TEAM - A TRANSISTORIZED ELECTRONIC ADDING MACHINE

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Konawa, Oklahoma, September 21, 1936, the son of Joe F. and Ollie L. Watson.

Education: Attended grade schools in Oklahoma, Alabama, Tennessee, and Texas; graduated from Canyon High School in Canyon, Texas, in May, 1954; attended the New Mexico State University 1954-1956; completed requirements for the Bachelor of Science degree from the Oklahoma State University, with a major in Electrical Engineering, in January, 1959.

Professional Experience: Employed part-time by the White Sands Missile Range 1954-1956, by the Pacific Missile Range during the summer of 1958, and by the Autonetics Division of North American Aviation, Incorporated, during the summer of 1959; member of the Institute of Radio Engineers.