

SYSTEMS ANALYSIS OF THE ALWAC III-E DIGITAL COMPUTER

by

GEORGE ANTHONY DUBINSKI

A THESIS

submitted to

OREGON STATE COLLEGE

in partial fulfillment of
the requirements for the
degree of

MASTER OF SCIENCE

June 1960

APPROVED:

[REDACTED]

Professor and Head of the Department of
Electrical Engineering

In Charge of Major

[REDACTED]

Chairman of School Graduate Committee

[REDACTED]

Dean of Graduate School

Date thesis is presented 2/17/80

Typed by Janette Crane

ACKNOWLEDGEMENT

This project was performed under the direction of Professor Louis N. Stone of the Department of Electrical Engineering. Assistance was obtained from Dr. Arvid Lonseth of the Department of Mathematics in regards to equipment and sponsorship through a National Science Foundation Research Grant. The author wishes to express his sincere appreciation to these two men and to the staff of the Oregon State Computing Laboratory for their cooperation and aid. A special acknowledgement is in order for Gilbert Bachelor, whose generous assistance proved invaluable, and for my wife, who made this work possible through her encouragement and self-sacrifice.

TABLE OF CONTENTS

	Page
Introduction	1
The Digital Computer	2
Definition	2
Classification of Digital Computers	4
Business Machine	5
University Machine	5
Process Control Machine	6
Commercial General Purpose Machine	6
Machine Functions or Instructions	6
Instruction Form	8
Arithmetic	9
Transfer of Information	9
Transfer of Control	9
Modifying Information (Non-Arithmetic System Operation)	10
Input Output	11
Start-Stop	11
Basic Design Features	11
Number Systems	11
Arithmetic	14
Modes of Operation	17
Circuitry	19
Logic Design	21
The ALWAC III-E Digital Computer	22
General Description	22
Design Concept	24
Circuitry	24
Order Code	29
Machine Structure Notation	34
Physical Structure	36
Basic Block Diagram	42
Memory System	46
Arithmetic System	51
Input Output	56
Control System	59
Modification of the ALWAC III-E Digital Computer at Oregon State College	71
Application of the ALWAC III-E	71
Purpose of Work Performed	73
General Procedure	74

	Page
Logic Structure of Modifications	76
Modifications Performed	83
I. Improvement of Existing Operations	83
A. Code Delete	83
B. Fast Channel Transfer (Memory to Working Storage)	84
C. Input Following Output	85
D. Stop Switch With an Optional "Jump" (IBW)	86
II. "Time Sharing" of Flip Flop Functions and Redundant Logic	86
A. Combine Gg Function Into X1 Flip Flop	87
B. Elimination of Y _p Flip Flop	87
C. Redundant Logic	88
III. Manual Machine Operation Aids	89
A. Command Break Point	89
B. Address Break Point	89
C. Subroutine Protection	90
IV. Orders with New Concepts	91
A. Count E Up and Compare Magnitude (16N)	91
B. One Character Output (DFXX)	93
C. "Flag" Operations (D9XY, D1W, D3W, DEW)	94
D. Count A by 1 (24, 26)	97
V. Convenient Orders	98
A. Exchange B and W (45W)	98
B. Absolute Addition between A and W (73W)	99
C. Compare Absolute Values of A and W (53W)	100
D. B and D Registers (08, 0A)	100
E. Flags Used as Optional Jump Switches	101
Conclusions	101

	Page
Additional Activities	104
Bibliography	108
Appendices	109

LIST OF FIGURES

Fig.		Page
1a	The ALWAC III-E	23
1b	The Control Station	23
2	Elimination of Pulse Reflection by Impedance Matching	26
3	Plug-In Boards	38
4a	Front View of Plug-In Boards in the Logic Cabinet	38
4b	Rear View of the Logic Cabinet Showing the Cable-Like Wire Formation	39
5	The Drum and Associated Plug-In Boards Located in the Memory Cabinet	39
6	Basic Block Diagram of the III-E	43
7	Memory System Block Diagram	49
8	Arithmetic System Block Diagram	52
9	Input-Output System Block Diagram	57
10	Control System Block Diagram	60

LIST OF TABLES

	Page
I. Characteristics of Medium and Large Commercial General Purpose Digital Computers	7
II. Basic Characters of the Binary, Decimal and Hexadecimal Systems	13
III. Operational Speeds of the ALWAC III-E	25
IV. ALWAC III-E Order Code Before Development of the Modifications	30
V. Circuit and Tube Distribution Throughout the Machine's Systems	41
VI. List of Elements Used in Conjunction with the Modifications	79
VII. Idle Order Codes that Existed before the Modifications were Developed	80
VIII. B Register Recirculation Chart	109
IX. D Register Recirculation Chart	110
X. E Register Recirculation Chart	111

SYSTEMS ANALYSIS OF THE ALWAC III-E DIGITAL COMPUTER

INTRODUCTION

This thesis is a report of the investigation of the ALWAC III-E digital computer serial No. 20, which is located at the Oregon State College Digital Computer Laboratory. From the information obtained, a group of modifications were developed and installed into the machine for the purpose of reducing programming labor and to improve machine operation efficiency.

There are three major sections. The first introduces the concept of the digital computer in respect to a systems structure. The computer is defined, its characteristics and features discussed. This information is intended to form a foundation for the analysis of the III-E which is presented in the second section.

Basic features of the III-E are reviewed in order to construct the philosophy of its design. The machine is divided into four systems and each is examined as a separate item and as a portion of the entire structure. Many details are not given because such information is available in the major reference used, which is the "Maintenance Manual of the ALWAC III-E, #20 (3), written by the author. For a clearer understanding of the material presented in this thesis, it is suggested that this manual and the Coding Manual (1) published by the ALWAC Company be studied.

The final section describes the methods used to develop the modifications, form the logic required, perform the installation, and evaluate the results. The modifications are described; their logic and the number of elements needed are given.

THE DIGITAL COMPUTER

Definition

This electro-mechanical machine has made a tremendous impact on our society during the past decade (1950-1960). Fabulous achievements have been accomplished with it. Scientific computations, data processing, and process control, have been performed at fantastic speeds with unbelievable accuracy. It has been compared to the human brain in concept. Yet, it is a machine, and as all machines, it is created and controlled by man.

The machine is a mechanization of the process of computation and data handling under preset and active control. The fundamental properties of the machine are shown below.

1. Information may be placed into or taken out of the machine through the use of the input-output system. This information consists of data and control instructions.
2. Information is stored internally in the machine's memory system.

3. The arithmetic operations of addition, subtraction, multiplication, and division may be performed using data obtained from the memory system. Transcendental functions are achieved through numeric methods utilizing the basic operations stated above. The arithmetic system is the center of the machine's activities.
4. The information supplied to the control system from the memory is interpreted and performed as directed.

Operations are executed by the machine at rates up to and beyond 100,000 a second. It can operate at this pace for days without error in instruction interpretation, execution, and computational results. It must be noted that most machines contain several automatic error detection and correction devices. Errors caught by these methods are not considered if they are of a random nature.

The digital computer is able to perform sets of specific operations utilizing data groups. The end result is always a mechanization of a man-created process. The advantages are speed, accuracy, and dependability. However, the outstanding achievement of its development is the releasing of man from the drudgery of bulky, tedious, type of labor. This machine is indeed an important tool in the modern industrial revolution of automation.

Classification of Digital Computers

The digital computer, when incorporating its peripheral equipment consisting of input-output and auxiliary storage stations, buffers, and control units, is termed a system. The computer or central processor is the major component of this system. It is the master control and processing unit, and itself is composed of the memory, control, arithmetic, and input-output systems. This paper is concerned only with the computer, hence the peripheral equipment is not directly considered.

Digital computers may be placed into one of four categories in regards to application and design features. The data processing or business application machine is concerned with handling a mass of data with minor computations. The university machine is used for scientific work where the amount of data is limited, but the calculations are intricate and numerous. A third type is the process control machine. It has recently appeared (1959) and will be used widely in industrial application. The final category is the commercial general purpose machine. It may be used for scientific computations, for business applications which are not gigantic in nature, and also for process control, when economically feasible. The characteristics of each type of machine will be reviewed.

Business Machine

Its physical size is extremely large because of the many types of input-output equipment and auxiliary storage units utilized. The speed of the peripheral equipment is high, but the computational rate is moderate. The design structure tends towards simplicity, yet multichecking features are incorporated for reliability. The cost is in the multi-million dollar range because of the large amount of peripheral equipment required. An example of such a system is the RCA Bismac (4).

University Machine

This type of machine will be found at various universities and also in government sponsored scientific laboratories. The input-output equipment is limited in quantity, but the other portions of the machine are bulky in size. The internal speeds are extremely fast in order to make many types of scientific computations practical. The design is complex, in order to achieve high speed of operation. The cost is in the range of \$500,000, which includes the peripheral equipment. Examples are the Illiac, Maniac II, George I, etc. (12).

Process Control Machine

The input-output system incorporates digital-analogue conversion. Auxiliary storage is usually not necessary. The speed of operation is moderate, for the rate of data sampling is the controlling factor. The machine is of a moderate design, hence the cost is in the vicinity of \$100,000. An example is the Ramo-Wooldridge RW300 (2).

Commercial General Purpose Machine

Since the characteristics are variable in this group, further classification is warranted. Sub-grouping into a medium and large class results. The small type of machine has not been found to be economical at this time. The characteristics of these sub-groups are shown in Table I (12). There is a gap between these two groups in which a third type of commercial machine could be placed. At present, two new machines could fit into this category. They are the Remington Rand Solid State 80, and the Electrodata 220.

Machine Functions or Instructions

The machine is constructed to perform a group of specific functions. These functions usually take form as instructions which are "obeyed" when directed. Construction is principally with electronic circuits. In order

TABLE I. CHARACTERISTICS OF MEDIUM AND LARGE
COMMERCIAL GENERAL PURPOSE DIGITAL COMPUTERS

<u>Characteristics</u>	<u>Medium Type</u>	<u>Large Type</u>
Cost	Less than \$150,000	Over \$500,000
Speed	Up to 1000 operations a second	Up to and over 100,000 operations a second
Direct Information Storage	Less than 4000 items	More than 8000 items
Designs	Simple to Moderate	Moderate
Input-Output Equipment and Auxiliary Storage	Medium quantity	Large quantity
Examples	IBM 650 ALWAC III-E Datatron 205 Bendix G15 Librascope LGP-30 ReComp II	IBM 704 and 709 UNIVAC II, 1103, and 1105

to achieve the performance of an instruction, the proper circuits must be energized. The sequence of operation may be arranged with repetition in any set of instructions desired. The particular task that is to be accomplished determines the set used. The set of instructions is termed a program and it is operated upon from the memory system in a stored program computer. The program is executed in a sequence directed by preset, or active control. The conditions of the active control have been determined prior to the program execution, making all control derived directly or indirectly from the man who initially formulated the program.

In order to understand the concept of a digital computer, the various types of machine functions or instructions will be discussed (10, p. 30-62).

Instruction Form

An information item is termed a "word". A word contains numeric data or machine instructions. An instruction is formulated numerically, therefore, both data and instructions have the same representation. The word is fixed in length, thus the number of digits available is constant. If the instruction length is small, more than one instruction may be placed into a word. An instruction usually consists of two or more portions. A specific operation, termed an order, is defined, and associated information

may be included. The types of orders will be discussed. The added information consists of one or more of the following: reference address of information, index requirements for modifying addresses, number of positions to shift information, number of input-output characters, timing references for instruction execution, etc.

Arithmetic

The types of arithmetic instructions have already been reviewed. The major variations that may exist in a single type are: the magnitude and precision of the numbers involved may change, and the operation may be algebraic or absolute in nature.

Transfer of Information

Information items or words may be transferred in part or whole, between various systems. These instructions are referred to as "housekeeping" operations.

Transfer of Control

The automatic preset sequencing of the program may be changed internally by the control system, with transfer of control instructions. This action may be conditional or unconditional. A conditional type gives the machine an opportunity to make a decision dependent upon magnitudes

of numbers, external switch positions, etc. The transfer of control instructions are used for branching purposes in program execution.

Modifying Information (Non-Arithmetic System Operation)

Information may be modified by indexing, logical, and shifting instructions. Indexing is a non-arithmetic system operation that assigns correct reference addresses for information items that are stored in the memory. This process reduces the quantity of instructions required for a program by enabling a single instruction to refer to many separate items at various occasions during the execution of the program.

Logical instructions are multi-arithmetic operations with binary numbers. The results are termed extract, masking, or transplanting (10, p. 31-34). Portions of a single word are easily isolated by this type of instruction.

Shifting will move the information in a word left, or right, considering a reference point. Aligning decimal or binary points in regards to arithmetic operations are achieved with this operation. Shifting may also be used to isolate a portion of a word.

Input-Output

Input-output instructions place information into or take information out of the machine utilizing the peripheral input-output equipment.

Start-Stop

The machine operation may be started or stopped by start-stop instructions. These may be unconditional or conditional. The conditional type gives the operator a chance to evaluate specific phases in the program execution, in order to check the validity of operation and the results, and to make decisions affecting control of the program.

Basic Design Features

The general design features will be reviewed briefly in regards to the university and commercial general purpose machine.

Number Systems

The choice in the number system structure used in computers, has been limited to binary and decimal. Theoretically, the most efficient is the number system to the base e , however, a physical element that could be used in such a case is not practical (8). The nearest number system to

e with a practical representation has been used extensively. This is the binary system; the system to the base two, for a two state element is easily obtained.

The binary system in itself, is impractical in regards to human use because of the great number of binary digits (bits) needed to represent numbers of only small magnitude. The binary system may be interpreted directly in any number system whose base is a multiple of two. This operation reduces the length of a number and provides a means of expressing binary numbers directly in usable notation. The system generally adopted for this practice is the number system to the base 16, termed hexadecimal or sexadecimal. The internal structure of the machine is binary, but the information may be considered in hexadecimal notation for convenience. Equivalent representations of the decimal, hexadecimal, and binary digits are shown in Table II.

In order to use decimal information, a binary machine must convert it into binary notation. This conversion is easily accomplished directly or indirectly, or by a combination of both. A direct method involves special circuitry, while the indirect method uses the machine itself to perform this task by executing a decimal-binary conversion program. In many instances, the conversion by program is very time consuming.

Because of this conversion time or added circuitry required, many machines have an internal decimal structure.

TABLE II. BASIC CHARACTERS OF THE BINARY,
DECIMAL AND HEXADECIMAL SYSTEMS

<u>Binary</u>	<u>Decimal</u>	<u>Hexadecimal</u>
0	0	0
1	1	1
10	2	2
11	3	3
100	4	4
101	5	5
110	6	6
111	7	7
1000	8	8
1001	9	9
1010	10	a
1011	11	b
1100	12	c
1101	13	d
1110	14	e
1111	15	f
10000	16	10

Instead of using a ten-state element, groups of binary digits are used to represent decimal digits. Efficiency is reduced, particularly in the arithmetic and memory systems. Although the decimal machine structure is not as efficient as the binary type, the advantages of working directly with decimal data may in some instances outweigh the disadvantages. A prime example is in the field of machines designed for business applications. The most popular decimal representational system used is the binary coded decimal. This system uses the first ten digits of the hexadecimal system. With binary coded decimal the advantages of the binary system arithmetic is kept in part. The arithmetic may be performed as with the binary system, then a correction is made to obtain the true result (7, p. 270-275). Some modern machines have incorporated both the binary and the binary coded decimal systems into their structure in order to make it more versatile and a step closer to becoming a true general purpose type of machine.

Arithmetic

Most computers are constructed to use a fixed point type of arithmetic. A word contains a constant number of binary digits; therefore, the magnitude and precision of numeric information contained in a word is limited by its length. The binary point is positioned before or after any bit in the word desired. This operation is termed

scaling and the numeric position of the binary point is called the scale factor. The man who formulates the program, or the programmer, is responsible for this task, for the machine does not recognize a binary point. The position of a binary point may vary from word to word, depending upon the magnitude of the number contained in the word and the precision desired. The various positions of binary points are set and then controlled by the programmer. The latter action is accomplished by the shifting instruction. Typical situations that may require a shift of the binary point are:

1. Addition and subtraction require that the two numbers or words involved have their binary points at identical positions.
2. In order to obtain the precision desired in the result of multiplication or division, the binary point of one or both words may have to be shifted before the operation is executed.
3. After multiplication or division the binary point of the result may not be in the same position as that of either of the words involved in the operation, therefore a shift may be required in order to place it to a standard position.

Before or after a fixed point arithmetic instruction, a shifting operation may have to be performed. This is time consuming and sometimes difficult to program because of the

number of different scale factors used. The range in magnitude and precision of the numbers are limited by the fixed number of bits in a word. Magnitude and precision may be extended by using two or more words to contain one number. However, this adds to the labor of programming and also extends the time required for program execution because multi-word operations are not usually included directly in the machine's instructions.

In order to eliminate some of the less desirable characteristics of fixed point arithmetic, many machines have floating point arithmetic operations available. This type of arithmetic is similar to the system used when performing computations with a slide rule. All numbers consist of a fraction and a characteristic which is some power of two in the binary machine. The value of the number is equal to the fraction multiplied by the characteristic. The major advantages of this system are: the magnitude of the number which may be stored in a word is greatly increased, and scaling is controlled automatically resulting in less labor in programming. However, the precision obtainable is reduced. Floating point operation is usually necessary for scientific work, while fixed point arithmetic is satisfactory for data processing. The general purpose machine, with only fixed point arithmetic, is used for floating point work by performing the floating point operations through the use of a special sub-program

termed a subroutine. This is a slow and cumbersome method. The university type of machine has incorporated automatic floating point and fixed point instructions since its primary use is in the field of scientific computation. The automatic floating point operation is more complicated than the fixed point type because scaling is included. Therefore, the execution speed is slower except when unique design features have been applied. Besides loss of precision, another disadvantage of floating point structure is that data which is being placed into or taken out of the machine, may require a conversion considering the decimal type machine. If the machine is binary in basic structure, a conversion is already necessary; extra work is not required. Precision may be increased by using multi-word operations; but this is awkward and inefficient because of the basic single word systems structure.

Modes of Operation

Desk calculating machines perform most of their arithmetic functions by a counting process. This method is slow, especially when the numbers involved contain many digits. Man has the capability to memorize rules and tables as those concerned with multiplication. This factor increases the efficiency of multiplication because it is raised from counting to a logical process. Most computers perform

their functions by the same methods as used by man, thus the operation has simply been mechanized.

Many machine systems perform an instruction by completing a group of individual steps in successive order. Such an operation is termed "serial". This may not be efficient for a machine, while for man, it is a result of his structure. Man is restricted in many aspects, as in recording information. He can only write a single character with one stroke, while a machine can be made to print groups of characters in a single stroke. Progressive design developed the parallel type of system where instructions are performed in a few steps, while in the serial type, groups of steps are necessary. Although the parallel type of machine can execute its functions at higher speeds, the added complexity of design resulted in a large increase in the quantity of components required, raising the cost.

The modern machine has carried the parallel concept into systems design. Previously, operations were executed singularly. In the concurrent systems structure, multi-operation is possible. Again, complexity yields higher speeds and additional expense.

The serial system is moderate in speed and cost. Parallel types are fast but expensive. A combination of serial and parallel operation could be a desirable compromise. The concurrent mode of systems operation may be applied to

all three types, resulting in the increase of speed, design complexity, and cost.

Circuitry

A computer performs its functions by logical methods. Circuits are arranged in various configurations in order to achieve the desired operation. Basically, three types of circuits are used. They are: storage units as the flip-flop, gating circuits as the "and" and "or" type, and amplifiers (9).

Each specific function may require a group of individual circuits plus a group which is shared with other functions. The result is that many of the circuits are "time shared" by numerous functions. Fewer components are therefore required, reducing total cost and increasing reliability.

The circuits used are interconnected directly with D.C. coupling or are A.C. coupled using transformers, R.C. networks, or other isolating devices. A combination of D.C. and A.C. coupling methods may also be used effectively. The D.C. type has the advantage of requiring fewer coupling components. This coupling is used primarily with asynchronous type of circuit operation, while the A.C. type tends toward synchronous control. Asynchronous circuitry operates at speeds determined by the individual switching characteristics of the circuits used. Total

speed or switching time is equal to the summation of individual circuit operational times. The synchronous type uses a central timing signal termed the "clock" or signals derived from the clock, which determine the circuit operational speed. In order to obtain effective switching, the individual circuit speeds must be greater than the clock repetition rate. The circuits are energized during the clock pulse or by its rise or fall condition. If the clock pulse width is small compared to the repetition period, the duty cycle will be low in value. One active element may then drive or be driven by many circuits. The number of input and output paths or "fan-in" and "fan-out" numbers may be large compared to the asynchronous type. Thus the synchronous system requires less active elements. Average circuit energy requirements are low in synchronous operation, but high peak levels may exist (11).

Systems may be composed of both types of circuit operation in separate or the same sections. A synchronous controlled system operates at a basic speed determined by the clock rate. The asynchronous system may use local asynchronous operation with single or sectional synchronous control timing signals. This type of structure is time controlled in a circuit group rather than in individual unit operation. If no external timing occurs, true asynchronous operation is obtained and is called speed independence.

Logic Design

The approach utilized in logic design may be primarily by the block diagram method, or through the use of logic equations (3, p.126-145). Usually, a combination of both are involved, the primary method being dependent upon the specific task and experience of the designer concerned. The objective of each method is to communicate between the systems designer and the engineer who must construct the final physical machine. The block diagrams, or logic equations describe in detail the functions performed by the machine in regards to logic organization and circuitry required. The order of development generates from the machine's requirements. The specific functions of each system are determined and formulated into a working structure with circuit characteristics considered. The structure takes form in block diagrams or logic equations. The end product of this work may then directly be used to construct the various systems that compose the machine.

THE ALWAC III-E DIGITAL COMPUTER

General Description

The ALWAC III-E is a medium size commercial general purpose digital computer utilizing drum storage. Photographs of the machine are shown in Figure 1. It is basically serial and synchronous in circuit and systems operation with automatic control sequencing. The clock line which is located on the drum has a repetition rate of 62.5 Kcps. The machine contains approximately 275 tubes and 5400 silicon diodes which are arranged in plug-in board construction. The arithmetic mode is the fixed point type. Internally, the machine is binary in structure with a 34 bit word. The first bit identifies the sign, the following 32 contain information, and the last bit is used for phase detection. A single address, or a self-addressed, or two self-addressed instructions occupy a half word length. Before this investigation, 81 orders were decoded for programming purposes. One indexing register is available and is activated for address modification by subtracting 1 from the standard instruction. The remote memory contains 8192 words of drum storage which are addressable in blocks of 32 words each. Quick access drum storage, termed working storage, contains four blocks of directly addressable words. A block transfer is used to communicate between the remote memory and working storage.

Figure 1a. The ALWAC III-E (right to left in rear - Power Supply, 23
Logic, and Memory Cabinets).

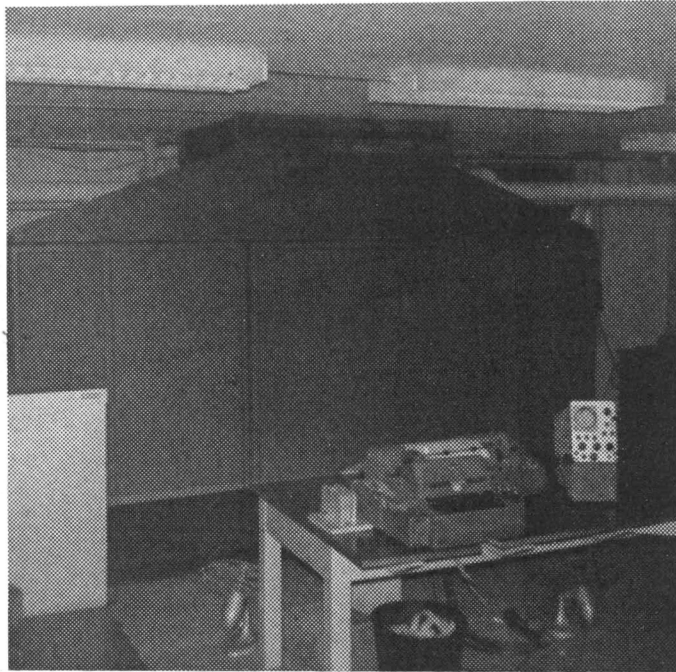
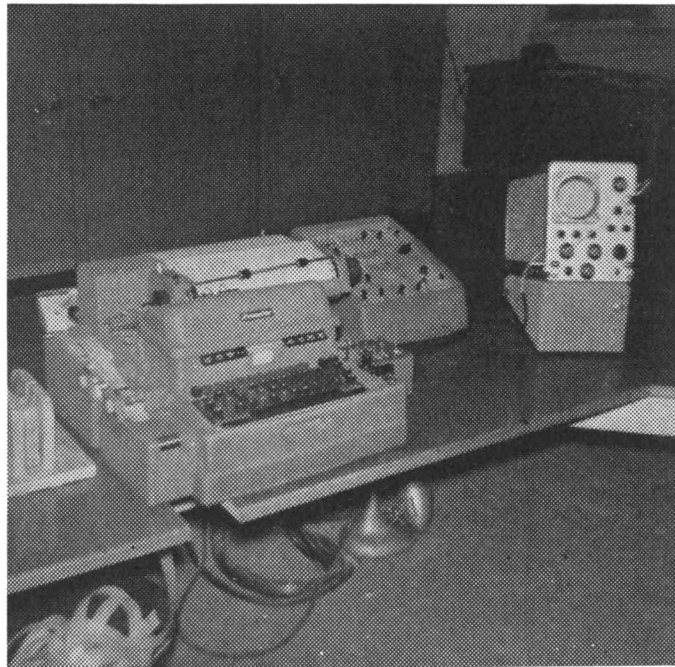


Figure 1b. The Control Station (right to left - Monitoring Oscillo-
scope, Control Panel, Break Point Box, and Flexowriter).



The input-output equipment, which will accept alpha-numeric information, consists of a flexowriter with a paper tape punch and reader. Basic operational speeds are shown in Table III (12, p. 16-22).

Design Concept

Simplicity and economy have been stressed in the design. The result is a basic yet versatile serial synchronous machine that may be produced at low cost (\$60,000). The III-E is a development of earlier models. Many important features were added without radically altering the initial design. This evolution improved the machine, however, many of the original concepts retarded its development. The point has been reached where major redesigning is warranted, but valuable information that results in systems improvement may be obtained from the investigation of the existing structure.

Circuitry

The major circuits utilized are: flip flops, "and-or" gates, and amplifiers. A modified flip flop (one shot circuit) and a pulse generator are employed sparsely. Three types of vacuum tubes are used in quantity (more than two). The absence of cathode followers, inhibit gates, and other types of computer circuits, reduces the complexity of the total circuit network. Coaxial cables are placed directly

TABLE III. OPERATIONAL SPEEDS OF THE ALWAC III-E

Arithmetic System

	Including storage average access time milliseconds	Excluding storage access time milliseconds
Addition	5.25 or 5.75	0.5 or 1.0
Subtraction	5.25 or 5.75	0.5 or 1.0
Multiplication	21.25	17.0
Division	21.25	17.0

Storage

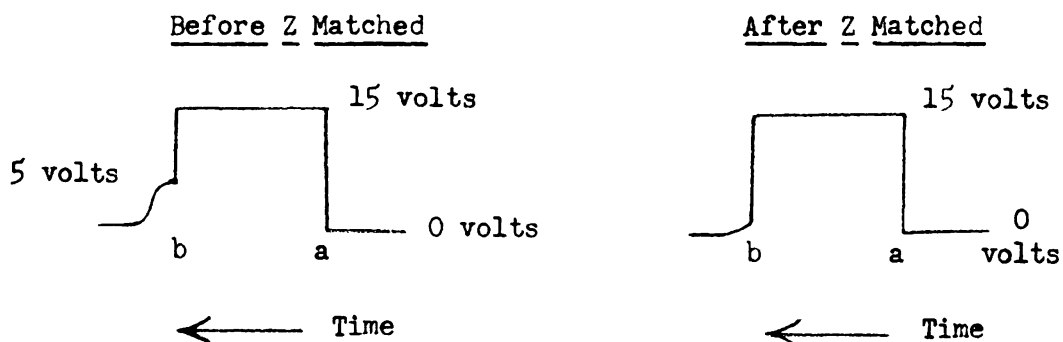
	Average in milliseconds
Block transfer from Working Storage to Memory	110.5
Block transfer from Memory to Working Storage	92.5
Access time in a Working Storage	4.25
Access time in a Register	0.25

Input-Output

Flexowriter typing	8 characters per second
Flexowriter paper tape reader	10 characters per second
Flexowriter paper tape punch	10 characters per second

into gates, and in some instances, pulse reflections result. By investigating various entrance points for the cable, a position which produces negligible reflection may be found. This procedure was used by the author to terminate the clock line cable in the logic cabinet. The result is shown in Figure 2.

Figure 2. Elimination of Pulse Reflection by Impedance Matching.



Note: Pulse width from a to b is 16 microseconds.

The gates are limited in most instances to the "and - or" or two level configurations. In order to effectively construct this type of gating, the "and" gates are cascaded and D.C. coupled (3, p. 8-20). This results in the use of a minimum number of diodes. A positive pulse will predominantly have its rise time extended by the "and" gate, while its fall time is increased by the "or" gate. When both types of gates are interconnected with D.C. coupling, the rise and fall characteristics of the output

pulse are largely determined by the final gate. This is caused by the loading effect of the circuit to be driven. Cascaded "and" gates have similar characteristics as that of a single "and" gate, considering the 16 microsecond period of the clock pulse. In order to obtain effective operation, the maximum number of "and" gates that may be cascaded is approximately five.

A positive potential of 15 volts indicates a 1 position in the binary system, while a zero or ground potential represents a zero. Flip flops are triggered by the fall of the clock pulse. The final "or" gates of the input gating networks for flip flops are A.C. coupled and are composed of an R-C combination and a limiting diode. The clock line is placed at the termination of each cascaded "and" gate in order to reduce the capacitance affecting the fall time of the gate's output pulse. When the "and" gate input configuration is satisfied, a clock pulse of 15 volts in amplitude and 0.15 microseconds in fall time is applied through the "and" gate to the input "or" circuit. A negative spike type of pulse, approximately five volts in amplitude, is then obtained from the input circuit and used to trigger the flip flop, which changes state in about five microseconds. The flip-flop driving networks consist of groups of cascaded D.C. coupled "and" gates collected together with A.C. coupled "or" gates which form the required triggering pulses for the flip flops.

Since the flip flop is a bistable element, two complete sets of driving circuits are required.

An important amplifying element termed a driver is used to regenerate pulses and reduce the total number of gating circuits required. The driver is a three stage D.C. coupled amplifier with two outputs that are 180 degrees out of phase. By using a driver instead of a flip flop as a collection point for circuitry that involves complex machine functions, the required number of gates is reduced by a factor of two. This is possible because the driver assumes a relaxed or zero position when the input is zero potential or not applied (3, p. 31-34). The driver input network consists of cascaded "and" into "or" type gates which are D.C. coupled. Since the driver is an amplifier, switching is not necessary; thus the clock line is not involved in the driver's input circuitry. The driver outputs have rise and fall times in the order of five microseconds. This condition can be tolerated because they are applied to flip flop driving or input networks, where the sharp fall of the clock pulse supplies the necessary trigger. In most applications the rise or fall time of the driver outputs must not exceed the clock pulse period of 16 microseconds. With the type of circuitry used, machine operation is dependent upon the rapid fall of the clock pulse. Flip flops are switched at the fall of the clock, while

the driver may change state during a clock pulse. These are the major circuit characteristics considered in logical design.

Conventional pulse amplifiers are employed to write and read information in the drum operations. Tube stages are placed in parallel in order to obtain power amplification, as in the construction of the clock line power amplifier, where as many as 24 stages may be in parallel. This is a crude but simple method and characterizes the general theme of the design.

Order Code

An instruction or command is composed of an order and an address portion. Each is eight bits in length, and together occupy a half word. A group of instructions contain orders that are self-addressed. Since address portion information is not necessary, they require a quarter word length (eight bits). One or two self-addressed orders may be placed into a half word, therefore, a word may contain from 2 to 4 instructions. Eighty-one orders were available before this analysis was undertaken. They are formed into groups and shown in Table IV (1).

An order occupies eight bits, therefore 256 configurations are possible. Two hexadecimal digits are used to represent each configuration. All even orders indicate a modification of the address portion by the index register,

TABLE IV. ALMAC III-E ORDER CODE
BEFORE DEVELOPMENT OF THE MODIFICATIONS

A. Arithmetic

1.	61	$A+W = A$
2.	63	$-(A+W) = A$
3.	65	$-(A-W) = A$
4.	67	$A-W = A$
5.	bd	$AB+W = AB$
6.	bf	$AB-W = AB$
7.	e1	$(BxD)+A = AB$
8.	e3	$(BxW)+A = AB$
9.	e5	$BxD = AB$
10.	e7	$BxW = AB$
11.	e9	$AB/D = B$ Remainder in A
12.	eb	$AB/W = B$ Remainder in A
13.	ed	$B/D = B$ Remainder in A
14.	ef	$B/W = B$

D. Change of Control

1.	11	Jump to W
2.	13	Control Jump 1 to W
3.	15	Control Jump 2 to W
4.	17	If $(E-1 \neq 0)$ to W
5.	19	If $A \neq 0$ to W
6.	1b	Stop or to W
7.	1d	If $(-A)$ to W
8.	1f	If Z on to W
9.	51	Z on if $ A < W $
10.	02	Reverse Z
11.	04	Z on if A is odd

B. Shifting and Extract

1.	A1	AB shift right
2.	A3	AB shift left
3.	A5	A shift right
4.	A7	A shift left
5.	AB	AB float left - Count in D
6.	71	Extract with D mask - $(AxD')U(DxW) = A$
7.	75	Extract $(AxW) = A$

C. Input-Output

1.	F1	Hexadecimal in
2.	F3	Alphabet in
3.	F5	Hexadecimal out
4.	F7	Alphabet out
5.	F9	Sign in
6.	d5	Sign out
7.	dd	Number out
8.	9b	Type
9.	9d	Punch
10.	9f	Punch and Type
11.	99	Neither

E. Special Accumulator (A)

1.	22	Round A from B
2.	28	$A = +0$
3.	2c	Make sign of A plus
4.	2e	Reverse sign of A
5.	3e	A is complemented
6.	0c	Sign of A to B
7.	0e	Sign of A to B and $B = 0$

TABLE IV. Continued

F. "House Keeping" or Information Transfer

1.	49	A to W
2.	69	A to W, W to A (Exchange)
3.	79	W to A
4.	b5	W to A (W is in Memory 00)
5.	C5	E to W
6.	41	W to B
7.	C7	D to W
8.	5b	W to D
9.	C3	E to W
10.	57	W to E
11.	C1	Address E to W
12.	55	Address W to E
13.	4d	Address of A to W
14.	4f	half A to W
15.	6d	Address W to A
16.	6F	half W to A
17.	30	A to B, B to A (Exchange)
18.	32	B to A
19.	34	E to A
20.	36	A to E, E to A (Exchange)
21.	38	D to A
22.	3a	A to D, D to A (Exchange)
23.	81	Memory to Working Storage I
24.	83	Memory to Working Storage II
25.	85	Memory to Working Storage III
26.	87	Memory to Working Storage IV
27.	89	Working Storage I to Memory
28.	8b	Working Storage II to Memory
29.	8d	Working Storage III to Memory
30.	8f	Working Storage IV to Memory

Note: 00 code indicates "no operation". Group E, orders 17 through 22 in group F, and "10 and 11" in group D are the self-addressed type. A, B, and D are one word registers. E is a half word register. W is a word in working storage.

reducing the total number to 128. Eighty-one were used as shown, leaving many open configurations for additional orders. As seen in Table IV, the codes in some groups were arranged in a non-related nature, resulting in an awkward structure. A prime example is the "house-keeping" group. This senseless placement resulted in many instances from the random addition of orders without present or future consideration of the order code structure.

The ALWAC III, which was the predecessor of the III-E, did not contain an index operation. Its order code consisted of even hexadecimal codes. When the III-E was developed and indexing placed into its control system, the orders that require address portion information were made odd. Even codes were used to indicate index modification. If an order is even, the address portion is modified, without exception. Orders that did not require address portion information (self-addressed) were not changed to odd codes. Since the address portion of this type of instruction is not used, the modification that takes place is meaningless. If two self-addressed orders are placed into a half word, or are doubled, the first order must be made odd. This prevents the second order, which is in the address portion of the half word, from being modified by indexing. In most instances, such a modification would result in erroneous operation and can not be tolerated.

Leaving the group of self-addressed order codes even was a result of the structure of the ALWAC-III, and it also labels this special group. As a result, the following rules exist:

1. Index operation is indicated by subtracting a one from an order than requires address portion information.
2. Doubling of self-addressed orders demand that a one be added to the first order used.
3. The address portion of an instruction using a self-addressed order is irrelevant.

This structure is not convenient and creates confusion for the novice programmer. It could be altered without involving any physical changes in the machine by adding a one to the self-addressed order codes, making them normally odd. Rule 1 would then apply to all orders and rule 2 and 3 would be replaced by the following:

- a. A self-addressed instruction must contain a doubled order or a "no-operation" in its address portion.
- b. If a one is subtracted from a self-addressed order, the contents of the address portion is of no consequence.

All orders would then have odd codes, and any change in their operation would be indicated by making the order even through subtraction. The procedure would be

standardized. It should be noted that in either code structure the second order of a doubled command situation may be the type which requires address portion information. The order configuration then acts as both, the order and address; however, this is not a normal situation.

The III-E has approximately 30 more orders than the average machine in its classification. By providing more tools for the programmer, his labor is reduced, and faster operation is obtained in program execution. This concept will be developed in the later portion of this paper. The order structure is large and versatile, but the hexadecimal representation and non-related order codes may cause difficulty in memorization. However, once the general and particular rules concerning the use of the order code have been mastered, the machine may be programmed with ease.

Machine Structure Notation

Logic equations are used to describe the machine's activities. These equations indicate the sequence of events of the specific functions that are performed, and the circuitry required. Once the equations have been mastered, the machine's logic may be inspected and its functions understood. A procedure that is recommended to be followed in order to acquire the knowledge necessary to comprehend the logic equations of the III-E is listed:

1. Be familiar with the general operations of "and-or" gates, flip flops, drivers, amplifiers, and one-shot circuits.
2. Relate the logic equation to circuit structure which basically involves "and-or" gates, flip flops, and drivers.
3. Understand the concept of the computer, its simplified and detailed block operation.
4. Review the symbols used to represent various elements in the machine, and also the general functions of each element.
5. Investigate the logic equations, developing their patterns in relation to systems operation starting with the basic timing section of the control system.

After these steps have been completed, with emphasis on items 3 and 5, the machine structure will become familiar. Initial knowledge of programming methods is not necessary, but it will aid in the development of the machine's block operation.

Since the logic equation book gives a reasonably complete description of the machine, it is the major source of information. In this book, a reference is given to the physical location of individual diodes of "and-or" gates, and is cross-referenced in respect to the logic equations. Thus, the logic book contains the equations and gives

reference to the wiring book and machine as to the physical location of the circuitry defined; the wiring book has individual diode and circuit locations listed and refers to logic equations that are constructed. In these two books, the functions of the machine are described and exact circuit structure shown. Separate schematics of individual circuits as flip flops, drivers, amplifiers, etc., are also provided. The machine structure is presented in a precise cross-referenced form that is not bulky in nature. This method has been found to be well suited for the III-E.

When using this scheme of logic equation representation, flow or logic diagrams are not usually necessary; however, a complex operation may be presented more clearly with a combination of equation and block diagram forms. A timing chart consisting of pulse lines in time reference, that are developed from the equations being investigated, is an important aid in the analysis of the equations of complex functions. The various applications of these methods are dependent upon the individual involved and the circumstances.

Physical Structure

The machine consists of the power supply, logic, and memory cabinets, and an operator's station. The power supply cabinet contains the necessary rectifying and regulating circuits that are needed to generate the proper

voltages for the computer. The D.C. voltages supplied are: +130V. regulated, +15V., +11V., -112V., -115V. regulated, -121V., and -280V.. The logic cabinet contains the major bulk of the logic type of circuitry, while the drum and its reading and writing circuits are in the memory cabinet. The control station consists of the input-output unit (a flexowriter with paper tape reader and punch), a control panel, and a monitoring oscilloscope. Basically, the arithmetic, input-output, and control systems are in the logic cabinet, while the memory cabinet houses the memory system.

The physical structure of the logic and "memory read and write circuitry" is similar. Circuits such as flip flops and amplifiers are placed on plug-in boards. Diodes used for gate construction and input circuits are also on plug-in boards. The diode plug-in contains 16 diodes and the input plug-in has 15 input circuits. Each plug-in board is terminated with an amphenol connector with 16 pairs of gold plated contacts. A flip flop, diode, input, and a working storage read amplifier board are shown in Figure 3. Two complete flip flop circuits and four input circuits are on each flip flop board. Figure 4 illustrates the boards (front and rear view) positioned in the logic cabinet rack. The drum and some of the boards used in the construction of its associated circuitry are shown in Figure 5.

Figure 3. Plug-In Boards (starting top left - Input, Flip Flop, Read Amplifier, and Diode).

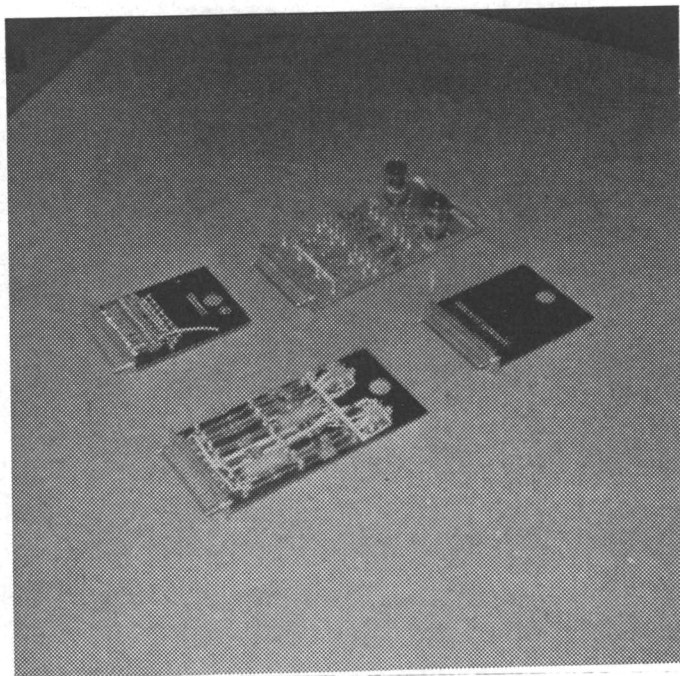


Figure 4a. Front View of Plug-In Boards in the Logic Cabinet.

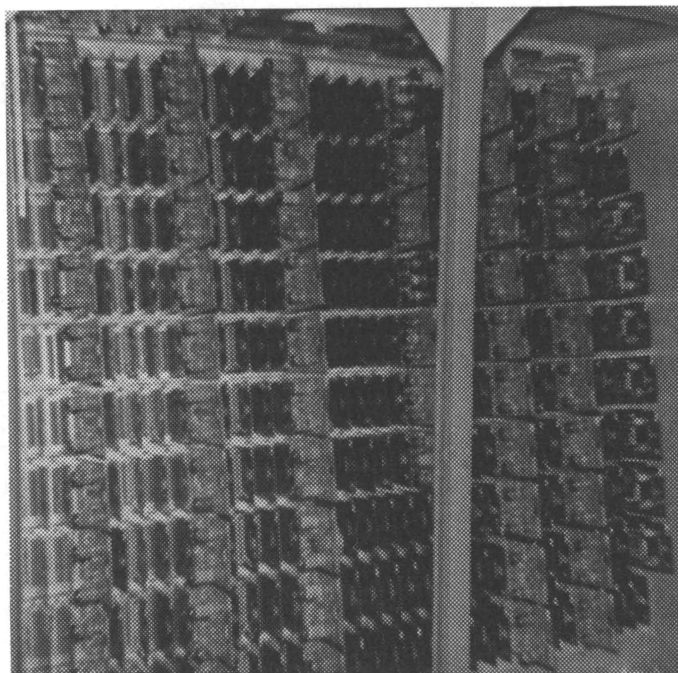


Figure 4b. Rear View of the Logic Cabinet Showing the Cable-Like Wire Formation.

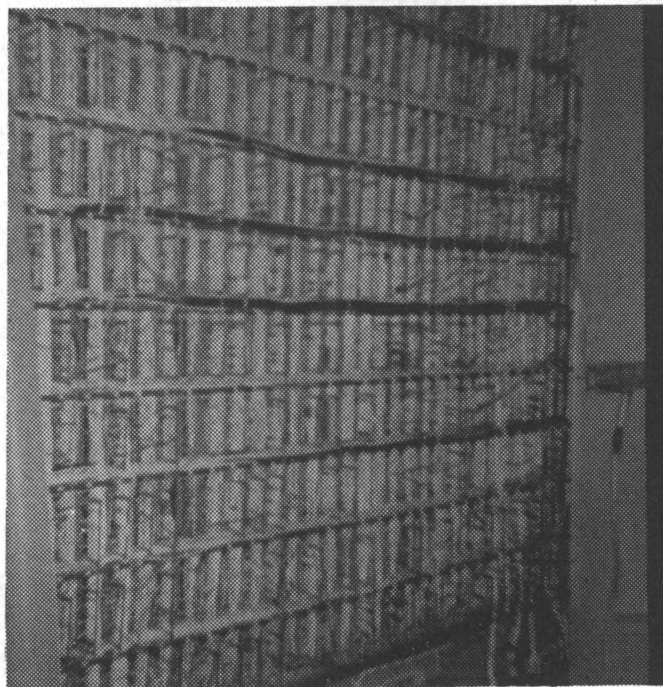
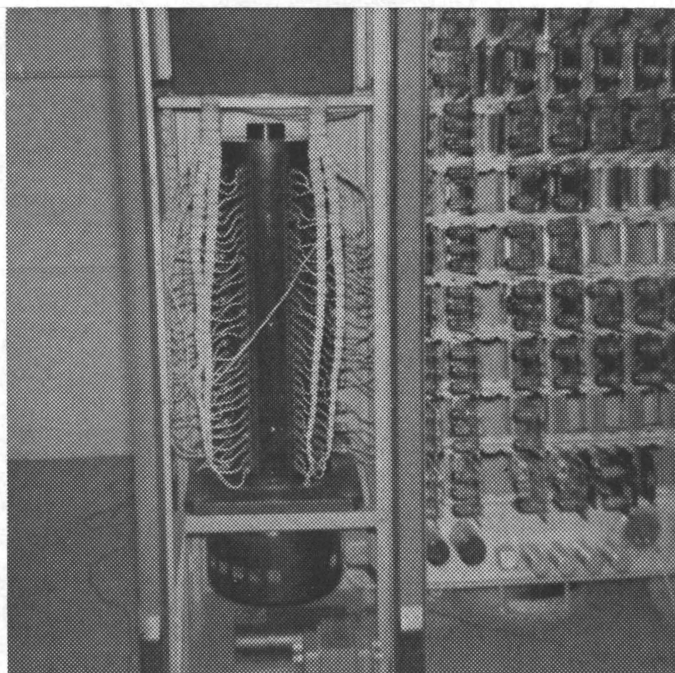


Figure 5. The Drum and Associated Plug-In Boards Located in the Memory Cabinet.



The D.C. gates are constructed from the diodes on the diode plug-in boards. The gates resistors and inter-diode wiring are placed on the rear of the plug-in rack. The result is a maze of wires in cable-like formations. This is seen in Figure 4. Individual wires approach lengths up to eight feet because the gates input sources, diodes, and output leads may be spread throughout the rack. This type of construction is feasible because of the relatively low frequency of the clock.

There are about 275 tubes (mostly dual triodes) and 5400 diodes in the machine. These are the principle elements used in the construction of 134 flip flops, approximately 85 amplifiers, 8 one shots, 794 input circuits, and numerous gates. The distribution of the major circuits in regards to sub-divisions of the four machine systems are shown in Table V. Quantities of tube types are also listed.

The construction used in the III-E, is economical and electronically satisfactory for the type of circuitry and operational speeds employed. The stray wire and load capacitances cause slow switching times which may approach the clock pulse period. However, the required flip flop trigger pulses are supplied by the fast fall of the clock, and reliable operation is attained. Care was not taken to use a minimal total wire length in gate construction and excess wire length is a normal, rather than an exceptional occurrence. The clock line distribution system and the

TABLE V. CIRCUIT AND TUBE DISTRIBUTION
THROUGHOUT THE MACHINE'S SYSTEMS

	F.F.	Dr.	Amp.	O.S.	B.O.
	(Flip flop, Driver, Amplifier, One Shot, Blocking Oscillator)				
<u>Memory</u>					
Working Storage Recirculation	32		16		
Remote Memory	2		2		
Buffer (Register and W.S.) . .	1	2			
<u>Arithmetic</u>					
Register Recirculation . . .	14	3	6		
Operation	8				
<u>Input-Output</u>					
Buffer	6				
Operation	2			7	
<u>Control</u>					
Master Station	24				
Synchronous Timing	12		3	1	2
Word Location	8				
Sequence Control	8	2			
Operations Control (Special).	13	6	75		
Indexing	5	1			
Alarm	3	1			
<hr/>					
Total circuits	132	14	102	8	2

Tube Types	F.F.	Dr.	Amp.	O.S.	B.O.		
5687	132	16	59	8	2	=	217
5963		8	6			=	14
12AT7			34			=	34
12AY7					2	=	2
6AH6			2			=	2
2D1				7		=	7
<hr/>							
Total							275

Note: Seven circuits are placed in two categories.

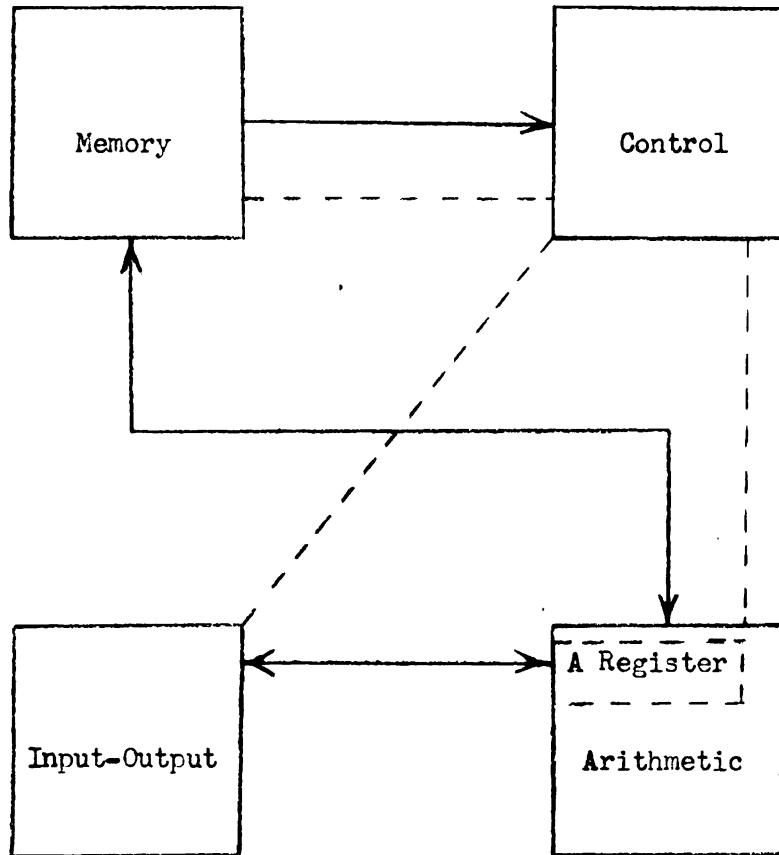
flip flop trigger pulses taken from input circuits use wiring that protrudes away from the main rack in order to reduce stray capacitance. Printed circuits which are placed on both sides of the flip flop and driver plug-in circuit boards, are not solder dipped and frequently have an open-circuited condition caused by radically changing temperatures, general usage, and age. With the exception of the printed circuits, the construction of this machine has proven to be suitable for reliable operation, and is extremely flexible in regards to additions and deletions made to the machine's systems.

Basic Block Diagram

The basic block diagram with information flow between the machine's systems is shown in Figure 6. Standard machine operation will be described using this diagram as a basis.

Consider the machine in an idle state. It is activated in order to place information into the memory, or to execute a program that is already stored in the memory. Both of these operations are performed by a program called the start routine. This routine is always kept in a specific location in the memory and is called upon by pressing the clear switch which is located at the operator's station. This action automatically places the first instruction of the start routine into the control unit. Upon the

Figure 6. Basic Block Diagram of the III-E



execution of this command, a statement consisting of four hexadecimal digits must be placed into the machine through the use of the input-output system. The statement is decoded and performed by the subsequent instructions of the start routine. A set of information items consisting of instructions, data, or both, is placed into the memory under the direction of the start routine, or the first instruction of a program that is already in the memory is loaded into the control unit. The latter initiates the operation of this program. Its instructions are then loaded from the memory into the control system and executed in a standard preset sequence. This sequence may be interrupted by a change of control or jump command; however, its pattern is continued, only the initial starting point has been changed.

Since the start routine is a program, many other operations may be performed by it other than the two mentioned; however, these two are the minimum required for activating the machine. The machine is considered under executive control when the start routine is being used because this routine interprets a simple statement and performs it through the use of many direct machine instructions.

The operation of the machine follows a constant pattern. A preselected instruction determined by the normal sequence or a change of control command, is loaded into

the control unit. The instruction is then performed by the control system utilizing one or more of the other systems. The execution of the instruction involves information transfer between systems or within an individual system, or information modification by the arithmetic system, or both. While an instruction is being executed, the physical memory location of the next instruction is sought. At the termination of the present instruction, the next instruction is immediately loaded, if its location has been completed.

The loading and execution of an instruction is a serial systems operation. A limited concurrent systems operation is obtained during the execution of an output command. The input-output system may send a character of information out of the machine while other instructions are being executed. If more than one character is involved in the output command, the machine will not proceed on to the next instruction until all but the last character has been taken out. Thus, the concurrent mode is only obtained during the output of the last character. Even though this example appears trivial, it is used to advantage in the binary to decimal conversion of output data. In this process, one digit of the decimal number is obtained for each step. While the digit is being taken out of the machine, the next digit is generated. Thus, the conversion does not require extra time. This example

illustrates the tremendous advantage of concurrent operation. The concurrent mode of systems operation is rarely used in this machine; hence, it will not be considered beyond this section. Concurrent operation may be achieved internally in a single system as in the control unit where execution of an instruction and the location of the next instruction are performed simultaneously; however, the general mode of system operation is serial. The machine is then serial in all systems operation and basically serial with slight concurrent operation considering an individual system.

Memory System

The memory is divided into two portions, the main and the working storage. The main consists of 256 blocks or channels, where each channel contains 32 words. The channels are addressable and are identified by the hexadecimal digits 00 through FF. The main is used for remote storage and a channel transfer exists for communication with the quick access working storage. An exception is channel 00, where individual words may be transferred into the arithmetic unit by a special order. The working storage is the active unit in the memory system. It contains four channels of 32 words each that are directly addressable. The entire main is on the drum, while portions of each working storage are in flip flops. A working storage is

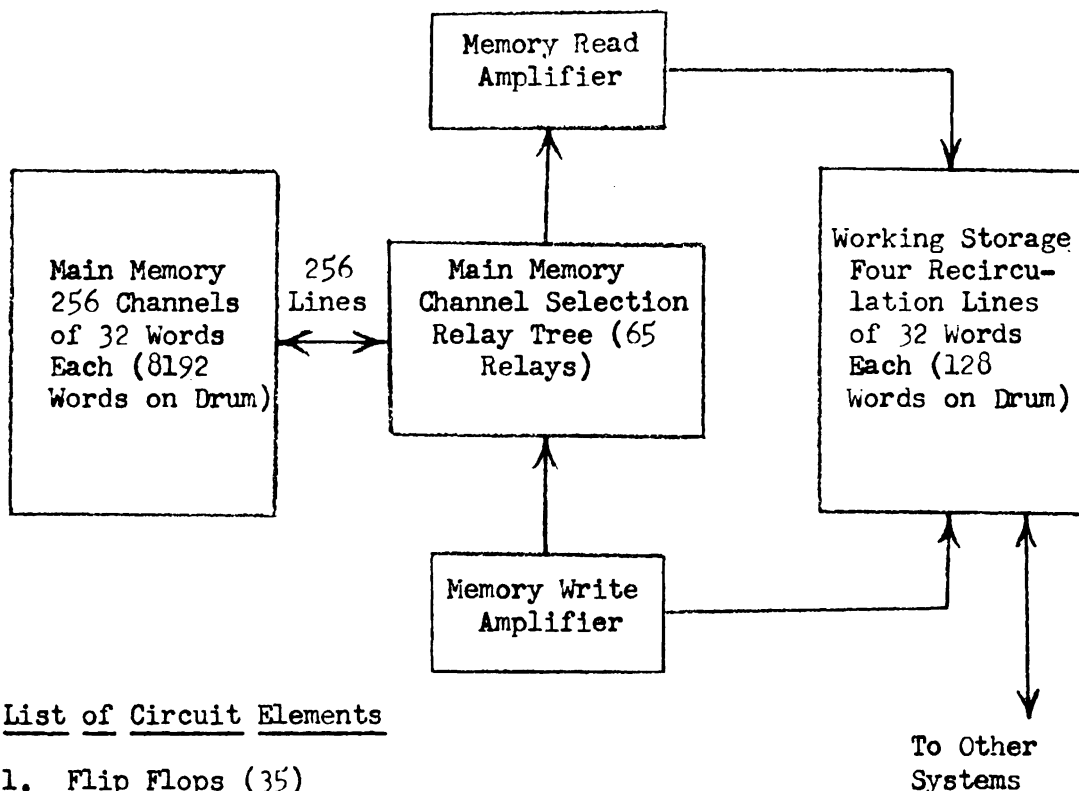
divided into two lines, the long and the short. The information recirculates from a line on the drum, to flip flops, and back to the other line in a continuous circular pattern. The two electronic portions of a working storage channel contain $4\frac{1}{2}$ bits and $2\frac{1}{2}$ bits respectively (3, p. 73-78). Since there are two breaking points in each working storage, the maximum waiting period for a specific word is $\frac{1}{2}$ a drum revolution ($8\frac{1}{2}$ milliseconds); and the average is $\frac{1}{4}$ drum revolution ($4\frac{1}{4}$ milliseconds). The main is a non-volatile storage, while the working storage is a volatile type since it recirculates. A return to zero phase type of recording is used in the main with the same head involved in both the reading and writing process. Working storage uses a non-return to zero method of recording with two read and two write heads because of its recirculating characteristic. The 34th bit of each word in the memory is permanently in the zero state in order to distinguish between a word that contains an all-one or an all-zero configuration, considering the working storage non-return to zero method of recording. Thus, the 34th bit of the words in memory are not accessible to the programmer.

Considering the average machine application, the main contains the program and data which usually occupy many channels. The program and data are used in blocks of 32 words, for both must be transferred into working storage in order to be available for usage by the machine's other

systems. Thus, blocks of information are constantly being transferred in both directions between the main and working storage during the execution of a program. In some instances the initial data is not stored in the main, but is taken when needed, directly into the arithmetic unit by the input-output system; however, the program is in the main and many block transfers are still required. The simplified block diagram of the memory is shown in Figure 7, with a listing of its specific circuits and their individual identification symbols.

Complication in programming and slow operation are results of this memory structure. A programmer must be very careful that the numerous block transfers are placed correctly in his program. The transfer takes about 0.1 second to perform, thus, it is time consuming in comparison to other operations. There is only one path existing between memory and working storage. A selection network that contains 65 multi-contact relays and 73 driving amplifiers, is used to connect the desired channel of the 256 available to the working storage. The information transferred is checked bit by bit for errors after the transfer is completed. If an error occurs, the operation is repeated if a switch has been placed in the required position. This check is necessary, for errors result when many channel transfers are made in a short period of time.

Figure 7. Memory System Block Diagram

List of Circuit Elements

1. Flip Flops (35)
 - a. Working Storage Recirculation (32)

W.S. I	Hb,He,H1,H2,H3	Ib,I2,I3
W.S. II	Jb,Je,J1,J2,J3	Kb,K2,K3
W.S. III	Lb,Le,L1,L2,L3	Mb,M2,M3
W.S. IV	Nb,Ne,N1,N2,N3	Ob,O2,O3
 - b. Buffer into Main Memory - Mo and Mp
 - c. Buffer from Registers to Working Storage - Q8
2. Amplifiers (18)
 - a. Working Storage Recirculation Lines (16)

Hr,Hw,Ir,Iw,Jr,Jw,Kr,Kw,Lr,Lw,Mr,Mw,Nr,Nw,Or,Ow
 - b. Main Memory Read and Write Amplifiers - MR and MW
3. Drivers (2)
 - a. Buffer from Working Storage to Control Unit - Ws
 - b. Buffer from Working Storage to Registers - Wn

Note: Selection relay amplifiers are considered a part of the control system.

In this situation, the contacts of the switching relays may not have relaxed properly in the time allotted for reliable operation.

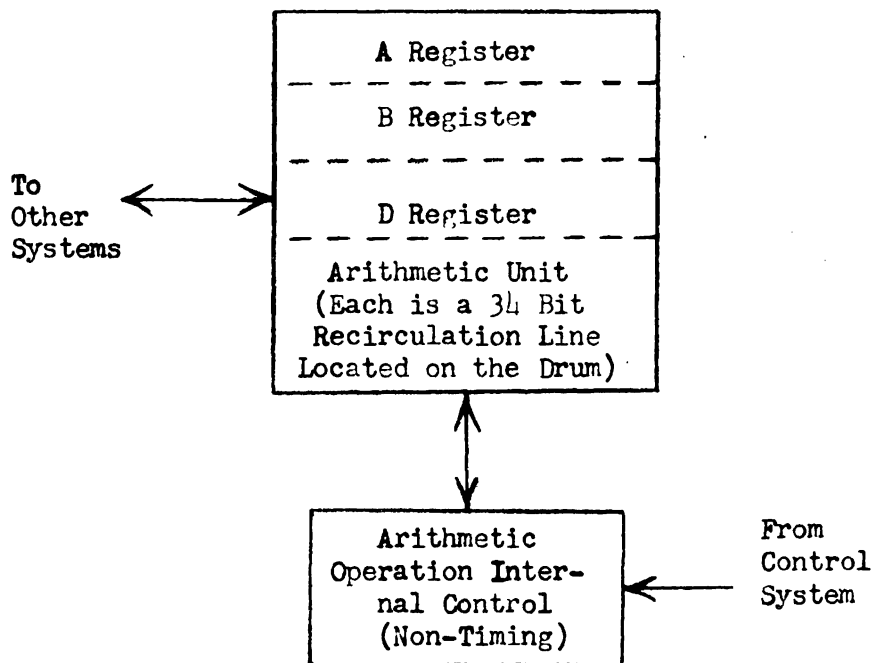
The remote main and slow block transfer makes this memory system undesirable. If each word in the main were directly addressable and the block transfer was fast, the memory system would be vastly improved. The quick access lines of the working storage are useful and would not be obsolete with a directly addressable memory. In fact, a few four word recirculation lines could be added and used to advantage. The command word structure does not allow enough bits for all the words in the main to be addressable; however, a preselect main channel command and the addressability of the words in this channel would make the main memory words directly addressable in banks of 32. The preselect would have to use electronic switching in order to be efficient. Any main channel could then be selected, and quickly connected to the path between the main and working storage. The words in this channel would be addressable. Since the size of the main memory is adequate (8192 words), with the improvements stated, the memory system could be greatly improved without re-designing the basic structure of the machine.

Arithmetic System

The major items of this system are the A, B, and D registers. Each is a one-word recirculating line with a $3\frac{1}{2}$ bit electronic portion (3, p. 66-73). Fourteen flip flops, 6 amplifiers, 3 drivers, and the drum are involved in their construction. Eight more flip flops are used for arithmetic operation elements, utilized for duties as carry, borrow, and inter-operation control. The latter applies to items as addition in multiplication and subtraction in division. The recirculation lines and operation elements are tied together with numerous "and-or" gates to form the arithmetic system. The basic block diagram with a listing of the elements used is shown in Figure 8.

A few of the flip flops are time shared with the control system. They are classified as multi-function elements since they perform more than one task. Elements of this type do not necessarily leave the area of a single system. Since these non-related functions take place at different time periods, many of the machine's elements are used in this manner. This multi-function arrangement reduces the total number of elements required, resulting in lower cost and greater reliability. Another outstanding feature of the arithmetic system structure is the application of the driver elements. The characteristics of the driver are: a zero state exists when no input is applied, and when an

Figure 8. Arithmetic System Block Diagram



List of Circuit Elements

1. Flip Flops (22)
 - a. Register Recirculation (12)
Ab, Ae, An, Ar, Bb, Be, Bn, Br, Db, De, Dn, Dr
 - b. Arithmetic and other Operations Control (10)
Ua, Ub, Uc, Ud, Ur, Us, Ut, Uv, Al, Bl
2. Amplifiers (6)
Register Recirculation - Ar, Aw, Br, Bw, Dr, Dw
3. Drivers (3)
Arithmetic and other Operation - Ad, Bd, Dd

input is received, the driver immediately changes to a one-state for the duration of the input signal. Since the drivers are collection points for the bulky logic circuitry of the arithmetic system, the total number of "and-or" gates used is reduced by a factor of two considering normal flip flop circuit construction without drivers. This reduction is possible because only the one state input logic is necessary for a driver, while the flip flop demands input logic for both the one and zero states. The non-storage feature of the driver is also valuable for switching during a clock pulse period. The major application of this characteristic is in the region of control.

Considering fixed point arithmetic, the A register is the prime accumulator, the B register contains the multiplier and dividend, and the D register the multiplier and divisor. The product is placed in the double length AB register which is the A and B register combined into a 65 bit register, and the B register contains the quotient and the A register the remainder, if the scaling of the dividend and divisor is identical. Addition and subtraction may also be performed with the AB register and a single information word. The dividend may occupy the AB register if the absolute value of the A portion is less than the absolute value of D, considering their scaling equal. The arithmetic operations that use the registers directly are basically single precision and fixed point. Since the

ALWAC is a single address machine, the standard procedure of arithmetic operation is: load a register from working storage, optional shift for scaling, perform the operation using a second word from working storage, an optional shift to obtain desired scaling of the result, and transfer of the result from a register into working storage. Since there are numerous exceptions to this procedure, many types of orders are available besides those defined for typical operation. Fifty orders deal directly with the arithmetic system. These orders may be grouped as follows: 14 arithmetic, 8 special A and B register, 7 shift and logical, 11 information transfer between registers and working storage, 4 inter-register transfers, 3 transfers between register, control and main memory, and 3 conditional transfers of control. This large and versatile group of orders reduces the labor in programming and decreases the program operation time.

The ALWAC is convenient for work using single precision fixed point arithmetic. Since the information word length is only 33 bits, the range of magnitude and precision of approximately 9 decimal digits, is not adequate for many applications. Double precision eliminates some of these deficiencies, but double length arithmetic is not directly available, therefore it is very difficult to program and is slow in execution. If single precision is sufficient, but a large variation in number magnitude is

desired, floating point arithmetic is satisfactory. The floating point arithmetic operations are performed by a subroutine, thus, they are slow in execution and cumbersome to use. The pattern of operation is as follows: place the subroutine into working storage IV, load the two numbers involved from working storage into the registers, place an information word called the key word into a register, and then transfer control to the subroutine in order to perform the specific operation that is indicated by the key word. The result of the operation is inserted in a register and the control is returned to the main program at the position determined by the key word information. Although floating point arithmetic is inconvenient and inefficient, the advantage of having a large range of magnitude available makes it the prime arithmetic mode used for scientific computations. Transcendental functions are evaluated by subroutines for both the fixed point and floating point arithmetic modes; therefore, the operation of each is similar. The floating point type performs the arithmetic required internally; the floating point arithmetic subroutine is not involved. Operational times of both arithmetic modes will be discussed in the latter portion of this paper.

The ALWAC arithmetic system is versatile for single precision fixed point computations, but applications are limited. The floating point mode is used for most work,

but its operation is slow and programming is awkward. Double precision is performed through intricate programming and is very time consuming in execution; therefore, it has not proven to be practical. The desired arithmetic system should be trivial to program and quick in operation for fixed point, floating point, and multi-precision arithmetic.

Input-Output

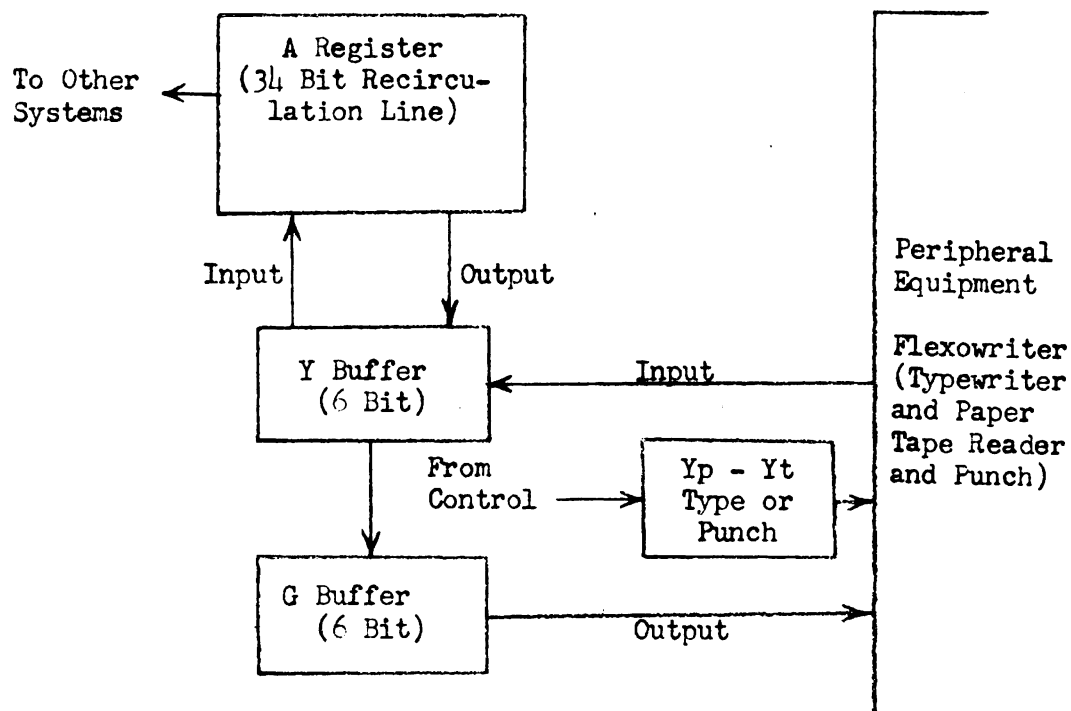
This system is composed of the peripheral input-output equipment and a six bit static buffer. The buffer consists of six flip flops in order to accommodate 4 bit numeric and 6 bit alphabetic or flex control characters. The general block diagram of the system with a listing of elements is shown in Figure 9.

Typical operation for the input situation is as follows:

1. Upon the execution of the input instruction, a character is placed into the buffer from the peripheral equipment.
2. The buffer (4 or 6 bits) is then inserted into the A register recirculation line, and the character enters the front of the register causing the contents of the register to be shifted one character length.

This process is continued in a circular pattern until the number of characters, as indicated by the address portion

Figure 9. Input-Output System Block Diagram



List of Circuit Elements

1. Flip Flops (8)
 - a. Input-Output Y Buffer -Y1 ... Y6
 - b. Peripheral Equipment Position Control -Yp and Yt
2. One Shots (7)
 - Output G Buffer - Go ... Gb

of the input instruction, are placed into the A register. The limit is 8 hexadecimal or 6 alphabetic characters because the A register is one word in length. The output operation is the dual of the input process described, with the addition of a second buffer consisting of seven one-shot circuits. This buffer supplies the required signals to the peripheral equipment.

The alpha-numeric property and limited automatic decimal-binary conversion are two convenient characteristics of the input-output system. Decimal integer to binary integer conversion (input), and binary fraction to decimal fraction conversion (output), are advantageous for many applications. This specific pattern is a result of the conversion requirements, where both involve a multiplication by ten; therefore, use the same logic. There are ten input-output orders which make this system flexible and comparatively easy to use except for the alpha or flex control input-output, which is cumbersome in character code identification.

The input-output system is serial in system operation, with the exception of parallel transfer into buffers. A slight concurrent mode of systems operation may be achieved as previously discussed. The input-output operations are considered serial because the control unit and A register are directly concerned. Since the peripheral equipment is basically mechanical, its speed is slow compared to the

electronic portions of the machine. This equipment drastically limits computer operation if a large quantity of information is processed, because the machine must wait in an idle state while the peripheral equipment performs its functions. Considering the III-E, the major methods that could be employed to eliminate this condition are: the use of fast operating peripheral equipment, and the adoption of the concurrent mode of systems operation. With concurrent operation, the computer could then simultaneously utilize the other systems while the input-output system is activated, eliminating the wasted idle periods. For efficiency, a buffer that has the minimum length of one word would be required. The only other additions are in the region of control, and they would not be too complex in nature. Since the III-E does not have the concurrent mode in regards to input-output, it is input-output burdened, causing its efficiency and capabilities to be greatly reduced.

Control System

The control system is the most diversified of the machine's four systems. It consists of the control unit, a one-word recirculating register, reference timing, period sequencing, and individual instruction execution control. The block diagram of the system and a listing of its principle elements are shown in Figure 10.

Figure 10. Control System Block Diagram

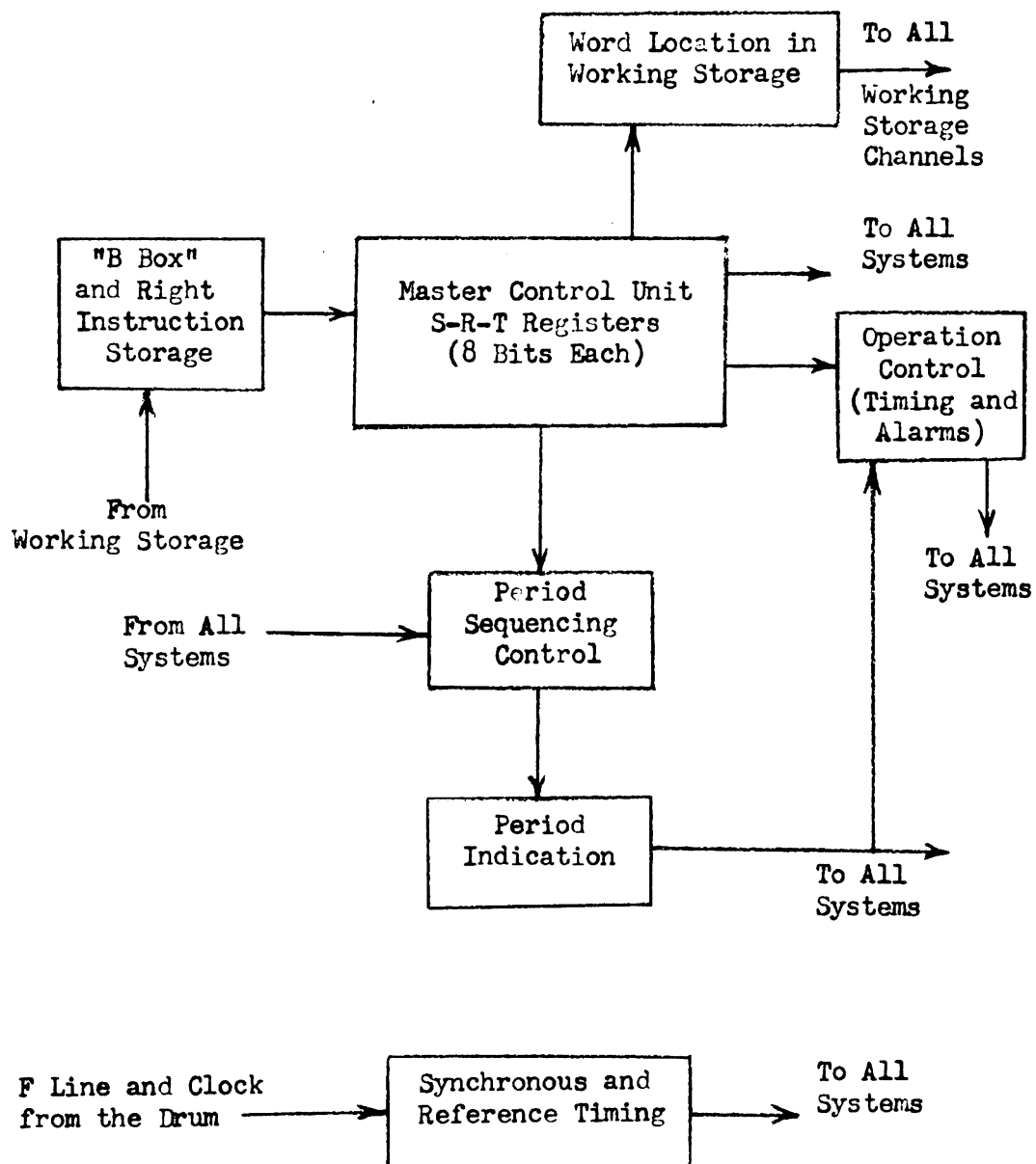


Figure 10 - Continued

List of Circuit Elements

1. Flip Flops (70)
 - a. Synchronous and Reference Timing (12)
Fb, F, F1, F2, F3, F4, F5, Po, P1, P32, P33, Vo
 - b. Word Location (6)
Or, Os, Va, Vw, Vr, Vs
 - c. Period Indication (8)
 $\phi_0 \dots \phi_7$
 - d. Operation Control (Timing and Alarms) (15)
Uc*, Ue, Uk, Um, Ut*, Cr, Fr', Fw', Gg, Md, B4, B5, A2, X1, Z
 - e. "B Box" and Temporary Right Instruction Storage (5)
Eb, Ee, En, Er, Ue*
 - f. Master Control Unit (24)
S1 ... S8, R1 ... R8, T1 ... T8
2. Amplifiers (4)
 - a. "B Box" and Instruction Temporary Storage - Er and Ew
 - b. Synchronous Timing (clock) power amplifier and Reference Timing - Cd and Fr
3. Drivers (9)
 - a. Period Sequencing - Q3 and Q4
 - b. Operation Control (Timing and Alarms)
Q1, Q2, Q6, Q7, A4
 - c. Control Buffer - Fr' and Fw'
4. Relay Amplifiers and "Blocking Oscillator and Cathode Follower" (2)

Synchronous Timing and Main Memory Write Timing - CA#1 and CA#2, Selection Relay Tree Amplifiers - (10 preamplifiers and 67 relay drivers)
5. Blocking Oscillator (1)

Main Memory Write Timing - Mos

* These elements are time shared between systems.

The control unit is composed of the S, R, and T registers which are eight bits in length and are constructed of flip flops. S contains the working storage address of the next instruction to be executed; R and T hold the address and the order portion of the instruction that is currently being executed. S is automatically counted in a preset sequence before the execution of an instruction. With a change of control command, the address in R is transferred in parallel into S, and the S sequence pattern is continued from this new position. The type of address information in R has previously been reviewed, but the predominant application of R is for working storage address identification of the information used in the execution of the order that exists in T.

An important function of the control system is the location of words in working storage lines as indicated by the R and S registers. Each has a separate system for this purpose, which are identical in structure. The register configuration is compared directly with the F line tags (from drum) and when they are matched, the next word is the one sought, thus a physical word location on the drum surface is found (3, p. 55-60, p. 92-99). Since this inspection is continuous, a concurrent mode is attained. This operation has proven to be the most difficult in the machine for comprehension because two different types of

sources are used in the comparison, a line from the drum and a flip flop register.

The T register configuration is sent throughout the machine to aid in activating the circuitry required for the order execution. This T configuration is the prime factor of control, while the B and S configurations have specific minor activities. The various T configurations that are present in the machine structure account for a large bulk of its logic, for the majority of the equations which describe the systems structure, contain T terms. These control T terms are usually satisfied by a group of complete T register configurations leading to multi-instruction usage of many major logical gates. Since the outputs of the T flip flops are applied to numerous gates, they tend to be over-loaded. This situation has been remedied by placing two tubes in parallel in some of the T flip flops. The R address information may also be used for order control; however, this is not the normal situation. Thus, the T register, which contains the order configuration, extends into every portion of each system and directly guides the computer's operation.

The S register is counted to indicate a left half word, then the right half of the same word, and then the left half of a word which is located four words later, etc. This pattern was adopted in order to automatically reduce the time involved in locating instructions that are to be

placed into the control unit. Its purpose is fulfilled; however, the odd counting sequence results occasionally in an undesired distribution of data in a working storage which contains a mixture of data and instructions. The disadvantage of referring to every fourth word could be eliminated if S counted by 1 instead of 4, and also by rearranging the working storage word identification in the F line, which is recorded on the drum. In the F line, consecutive address tags would be placed in every fourth word location, causing consecutively addressed working storage words to be physically located in every fourth word position on the drum. With this structure, instructions are positioned in consecutive working storage words and the advantage of automatic reduction of look-up time still retained. However, a programming guide rule would be necessary to determine which working storage words, that are to be used in instruction execution, have minimum access time. With the present arrangement, working storage word locations that are optimum, or have no look-up time, are easily found by the programmer. This modification would then involve the recording of a new F line track on the drum, and the changing of the S counting logic. Instruction words would follow consecutively instead of jumping to every fourth word, but a hand guide has to be provided for the programmer in order to determine optimum working storage addresses.

A machine cycle consists of a look-up-load and execution phase. The instruction to be performed, as indicated by S, is located and loaded into R and T from working storage, or from the right half of the E register. When a left half word instruction is taken from working storage, the next instruction which is normally located in the right portion of the same word, is placed into the right half of E. S is counted at the termination of the look-up and load phase. The order in T is executed using the reference information in R. While this is taking place, the next instruction as indicated by S, is being located in working storage. It may already exist in E, however the working storage location is still performed although it is meaningless and does not have to be completed.

The III-E is designed to automatically operate at maximum efficiency with instructions that take one-word-time or a multiple of 16 word-times for execution. Since many instructions do not directly fit into this category or the working storage information word that is required may not be immediately available, extending execution time, the machine can not perform at full efficiency. However, the normal sequence pattern does help to increase efficiency independent of the type of instruction being performed. This is possible because two instructions are taken from working storage as a result of the location of a left half-word instruction. The left instruction is

placed into the control unit, and the right instruction is stored in the right half of the E register. The right instruction is immediately available for loading into R and T after the left instruction has been executed. Effectively, every other instruction is taken from E with no location delay involved. The loading and execution of the "two instruction word", which contains instructions that are each executed in one-word-time, takes a total of four-word-times. Hence, consecutive "two instruction words" are placed in every fourth word in working storage to eliminate instruction location time. As stated previously, this "fourth word" structure is not effective in many instances.

Provisions that have been provided for reducing the location time for instructions are:

1. Forming a working storage line into two 16-word portions.
2. Picking up two instructions from working storage in a single "look-up" operation.
3. Having every other instruction immediately available in the E register.
4. Adopting a sequence involving every fourth word in working storage to take advantage of instructions that are completed in one-word-time or a multiple of 16-word-times.

When an instruction requires the use of a working storage word, there are 8 of the 128 words that are immediately

available during the word time following the loading process or are optimum. It is difficult and usually impractical to utilize them. Therefore, many instructions that may be performed in one-word-time have their execution time extended because of the use of non-optimum addresses. The location time of the next working storage instruction is also increased. Since the working storage has a split construction, average delay in word location is eight-word-times. It is impossible to take complete advantage of the normal sequence pattern; however, for efficiency, important subroutines are constructed to apply optimum words when feasible. It has been found that these subroutines are a great asset because of their constant use, where one may be performed thousands of times in the execution of a single program.

The right half of the E register is used for instruction storage and the left half is principally involved in address modification or is a "B Box". While each instruction is loaded into the control unit, the contents of the "B Box" is subtracted from the address portion of the instruction, thus, the address is modified. The modified and unmodified addresses are placed into the control unit during each loading process. The desired address, as determined by the odd or even state of the order portion of the instruction, is held and the other is destroyed. The III-E was one of the first commercial machines to

incorporate a "B Box" as a part of its basic control system. It was added to the machine when it advanced from the III to the III-E. Excellent work was performed in this task, for the "B Box" was effectively fitted into the existing structure. The utilization of the remaining portion of the B register as a temporary storage for right-half word instructions was also an outstanding achievement, and took place in this same transition period.

The basic timing is derived from the clock line which is permanently recorded on the drum. It is frequency divided to form five other timing pulse lines, which with the clock, mark a word, half-word, quarter-word, two-bit period, and a single bit. Four special timing lines that are derived from the six lines described above and the F line identify the 1st, 2nd, 33rd, and 34th bit of each word. The F line contains a timing pulse for every eighth bit of a word and also address identification configurations for the 32 words in a channel. The clock line, F line, and the subsequent pulse lines are used for synchronous timing in regards to a single reference. The F line provides this reference with its individual word address identification tags. The basic timing of the machine is dependent upon the mechanical drum because the electronic circuit operation is a slave of the clock and F line. Thus, the drum and the electronic portions of the machine are tightly locked together to form an effective, single unit.

Machine operation is divided into eight possible periods. The first is for instruction look-up and loading; the second for an inspection period used to halt the machine if an alarm or operator's stop condition exists, and the last six are involved in instruction execution. The machine sequences consecutively through the periods. The length of time required and complexity of an instruction will determine its specific termination period. This multi-step operation is advantageous, for in design various activities are easier to isolate for control and elements can be time shared more readily since their separate functions may be restricted to specific control periods. Also, the causes of machine breakdown can be located with less effort by noting the period in which the machine may have stalled. Most machines employ a three-period structure, but multi-phase operation has proven worthwhile for the III-E. Three flip flops can indicate eight periods; however, eight individual units are used. This construction prevents overloading of the flip flops concerned because the quantity of logic needed to specify a single period or groups of periods is reduced. Also, with eight elements, the machine structure is simplified since one element is used for a single period indication.

Instruction execution control concerns various timing functions needed in the operation of particular instructions. Instruction internal control is considered a portion of the

system in which the activity takes place. It is difficult to separate these two groups; however, when general timing is involved, the control system has charge. The elements in this group are usually fitted into the "time shared" class, for their functions are restricted to one or more groups of similar instructions.

The control system is serial in systems operation, for only one instruction may be executed in a single cycle of machine operation. In order to enter the concurrent mode, each system must be provided with a separate control unit. To achieve this type of structure, many system position indicators are necessary in order that the major control unit can effectively direct all activity. This advanced type of operation could be approached in many sections of the present structure. Independent input-output has been considered. Concurrent preselect of a main memory channel and concurrent transfer of a block of information between memory and working storage are practical considerations. Minor improvements, as complimenting the result of subtraction during the instruction "look-up and loading phase" instead of using more time in the instruction execution periods, are also possible. The continuous process of working storage word location and address modification while loading an instruction into the control unit are two examples of concurrent operation that presently exist in the control system. It is seen that parallel structure,

when economically feasible, leads to a higher degree of achievement in digital computer design.

There are ten orders that deal directly with this system and are in the area of change in control. These orders are used to make decisions within the program, dependent upon variable information that is generated internally or by fixed information that has been preset, such as a numeric constant or a control panel switch position. The valuable "B Box" order is in this group and is effective in automatically determining a desired sequence of addresses, and for program looping. As in all of the systems of the III-E, the large group of orders provided, aids the programmer to be efficient in regards to program formation and operation.

MODIFICATION OF THE ALWAC III-E DIGITAL COMPUTER AT OREGON STATE COLLEGE

Application of the ALWAC III-E

The activities of the Oregon State College Digital Computer Laboratory, which is located within the Department of Mathematics is centered around the ALWAC III-E serial number 20. Principle applications are in the area of education and scientific computation. An undergraduate course in computer coding uses the computer for approximately 20 hours a week for laboratory periods, where the students have direct contact with the machine under the guidance of

an instructor. Approximately five to ten graduate mathematics students and other personnel from the Mathematics Department use the computer in various phases of their work. Many individuals from other departments on campus, as Chemical Engineering and Soils, process data, develop programs, and use program systems that are available in the program library. An algebraic compiler which was completed recently (May 1959), is very popular because of its simplicity in programming, but its utilization is restricted to the region of recurring formula evaluation. Since its installation in March of 1957, the machine has averaged over 12 hours daily running time.

In general, a multitude of programs of various lengths are constantly being written, but only a few have been employed on a production operation basis. These exceptions are practical subroutines and major program systems as Statistical Correlation, Linear Programming, Business Games, etc. Most programs use floating point arithmetic with its many subroutines. Since these subroutines have frequent application, optimum programming techniques are employed in their construction to increase operation speed. Thus, besides the coding laboratory periods, the machine is normally occupied by the time consuming and difficult task of program development with limited production operation of the programs already in existence. It should be noted that problems exist in various departments on the

Oregon State College campus which are not practical for application of the III-E because of its relatively slow speed.

Purpose of Work Performed

The development of a program involves the following steps:

1. Determine the exact procedure to be used in obtaining the desired solution of the problem to be programmed.
2. Know how to program at the minimum level required for the particular task.
3. Derive a complete detailed flow diagram in block form of the entire project.
4. Code the program loosely in block form directly from the flow chart.
5. Check each block singularly for proper operation on the machine (debugging).
6. Tie the blocks together to form the complete program, test its performance, and then proceed (as time allows) to tighten the structure for efficient operation.

Items 4, 5, and 6, deal directly with the machine and are the major points considered in this work. Convenient orders added to the order code give the programmer more useful tools, reducing the time and labor in programming.

Item 5 concerns "debugging", which is performed to some extent on the machine. "Debugging aids" in the form of manual "break points" provide the programmer with a larger degree of direct machine control, eliminating many awkward procedures. Program efficiency is increased by adding orders that perform functions which previously would take groups of orders, and also by improving the performance of existing operations.

The specific goals of this analysis are:

1. Reduction of labor and time of programming by placing new, useful orders into the machine's structure.
2. Provide "debugging aids" which reduce the machine time needed for this purpose.
3. Improve the efficiency of program operation, especially subroutines, through the addition of orders that perform the present duties of groups of orders, and by improving the performance of existing functions.

It is seen that most new orders will fulfill the objectives of items 1 and 3.

General Procedure

The mechanics of the work outlined involved the following: a detailed analysis of all phases of the present physical structure, the reviewing of the existing order

code, the forming of new orders and other items that are useful for the programmer and are efficient in regards to machine operation, and the placing of the modifications developed into the machine's structure. This project was terminated when the idle elements available within the logic cabinet were completely utilized in the construction of the modifications.

In order to prepare for the final stages of this work, a thorough detailed study was conducted in regards to the machine's applications, order code, systems structure, and circuitry. Other computers, as the Electrodata Datatron 205, and Bendix G-15, were also investigated in many of these aspects. Through this knowledge and the information obtained from consultation with the advanced programmers of the computing laboratory, specific items were formulated and placed into the machine. They were tested for usefulness and electronic reliability during an evaluation period, and from the information gathered adjustments were made when necessary. The difficult phase of this work concerned the selection of the specific modifications that were most beneficial from the large number considered. A detailed knowledge of the machine and practical programming experience were the major assets used in making the decisions. The final group was chosen for their usefulness and ease of construction. The logic representation and the

installation of the particular items were secondary, for the knowledge required resulted from the physical systems analysis performed.

Each modification of the order code had to take form as an addition, rather than a change. If this rule was not maintained, existing programs would not be able to be performed. Confusion always resulted when new orders were used before the acceptance procedure was completed, for some items were changed or eliminated in the evaluation period. Thus, the programs that contained them would not operate properly.

Logic Structure of Modifications

The logic equation is the notation used in the description of the machine; hence it was the principal method employed to express the detailed form of the various modifications. The equations derived for each item* indicates the quantity and type of elements needed and the labor involved in its construction. As previously stated, the number of elements required and the ease of installation were important factors in determining which items of the group considered were finally selected. Therefore, each was investigated thoroughly in these respects.

Before a specific item was chosen, its general logic formation was closely reviewed. After an acceptance, the

* An item is a modification from the group developed.

detailed equations were developed. The steps involved in this task are the same for most types of items. In forming the equations, logic blocks already existing in the machine were employed when possible in order to reduce the number of new elements required. In many instances, the logic structure and item functions were altered from the most desired form to take advantage of the machine's present logic. Sound electronic construction also dictated a structure which would cause deviations from the initial logic formation. The physical placement of the logic circuits in the computer was arranged to use a minimum wire length to keep stray capacitance small in value. The optimum distribution of elements and of wire length was difficult to achieve, because the entire group of modifications were developed over a four-month period, and the installation was carried on continuously. At all times the general pattern of the modifications was known; however, changes which would cause revisions of the overall plan would occur as the work progressed. It was impractical to keep revising installed modifications; therefore, an ideal utilization of elements and wire distribution was not attained, but there was slight waste.

There was no specific investigation for redundant logic because the time which would be required did not justify the end results; but small groups of redundant logic were located in connection with other work. The

removal of this logic provided more elements for the modifications. The functions of two flip flops were incorporated with those of a third flip flop, releasing them for other purposes. This "time sharing" change was performed because two flip flop elements were needed for a modification and none were available. Table VI shows a tabulation of the elements involved in the total group of modifications. It should be stressed that efficient use of these elements was a prime consideration because there were more items to be installed than were possible with the number of elements on hand. However, with careful planning and the redundant logic found, the situation was relieved.

The code for a new order was chosen from the idle group of order codes which are shown in Table VII. The function of the prospective order, its similarity with an existing group, and the present operation of the suitable open codes were the major factors which determined the final selection. From the description of the present operation of a code, the extent of the logic required to bring it to the desired "no operation" state is easily determined. Thus, available "no operation" codes or ones with trivial functions are more convenient because no or minor logic is needed to eliminate the existing erratic operation. By careful selection the new orders were placed in similar groups, and the logic required was as minimal in form as feasible.

TABLE VI. LIST OF ELEMENTS USED
IN CONJUNCTION WITH THE MODIFICATIONS

Diodes available in main rack (prior to October 1958)	95
Diodes released	24
Diodes used in main rack	119
Diodes used outside of the main rack	16
Input circuits released	1
Input circuits used	27
Flip Flop circuits released	2
Flip Flop circuits used	2

Note: All elements concerned were located in the logic cabinet.

TABLE VII. IDLE ORDER CODES THAT EXISTED
BEFORE THE MODIFICATIONS WERE DEVELOPED

<u>Order</u>	<u>Functions</u>
00 No operation.
06 No operation.
08 E into B and bits 0 to 16 are made zeros (B).
0A E into D and bits 0 to 16 are made zeroes (D), and -zero into B.
2A -zero into A and D.
3C Complement A.
47 B U W into B.
4B A into W and -zero into D.
5D D U W into D.
5F Same as 5D.
6B A into W, W into A, and -zero into D.
73 Same as 71*.
77 Same as 75* except -zero or all ones into E.
7B W into A and -zero into D.
7D Same as 6D*.
7F Same as 6F*.
A9 Stop, +zero into A, -zero into B, and all ones into D.
AD Same as BD* except R will count down.
AF Same as BF* except R will count down.
B1 Same as B5* except -zero into B.
B3 Stop, -zero into A, zero into B.
E7 Stop and -zero into A.
B9 Stop, zero into A, Zero into B.
C9 Z on if absolute value of A is equal or greater than the absolute value of D.
CB Same as C9.
CD Same as C9.
CF Same as C9.
D1 Stop.
D3 Stop.
D9 Stop.
DB Stop.
FB Same as F9*.
FD Same as DD* except the first four bits of A are made zero (sign of A is not changed).
FF Same as DF* except the first four bits of A are made zero (sign of A is not changed).

* These orders are normal types and are listed in Table IV.

The logic for a modification generally consisted of the principle operation and control sequencing equations. The principle equations contain the T order configuration, the order execution period or periods indication, the operation mechanics terms, and if necessary, word portion or other timing. The sequence equations are composed of the T configuration, order execution period, and work timing. Occasionally more timing terms or equations were necessary, but usually the timing provided by the execution period specified and the sequencing equations were sufficient. For most items, the total logic formed consisted of new equations and changes in one or more of the machine's existing equations.

Two important considerations in many modifications are: register recirculation for the T order configuration, and the "look-up" of the next instruction. When a new code is used the recirculation of the four registers must be checked. If it is not normal in one or more registers, logic must be added to correct the situations, unless the non-recirculation is of no consequence. By referring to a register recirculation chart, the status of the register for a code is seen, and if required, logic can easily be formed to cause recirculation, or deleted to stop it. The register charts which held true at the start of this project are shown in the appendix.

If a command is terminated during a word period, other than the 1st, 33rd, or 34th bit, the next instruction location process which is in an active state, must be halted. In this situation it is possible that the loading of an instruction would start immediately, resulting in erroneous information transfer because the timing is not correct. Thus, the "look-up" process must be reset if an order is terminated within a word time.

After the logic equations were formed in the III-E standard structure to use the minimum number of elements and wire length, the modification was installed. Installation time is dependent upon the number of connections to be made, where each takes from 10 to 15 minutes to complete. Reliability was evaluated by checking the installed items under marginal conditions. The machine's D.C. voltages were varied within the accepted limits (3, p. 167-171) and its operation observed. If it was not satisfactory, errors were generally caused by large time constants which would not allow circuits to charge sufficiently in the time allotted. This situation is located by viewing the active pulses with an oscilloscope. Resistors are then placed in the gating circuits to decrease the time constants. In most instances the operation of an item was satisfactory with normal circuit construction and adjustments were not needed.

Modifications Performed

This section contains information concerning the group of modifications that have been performed on the ALWAC III-E digital computer located at Oregon State College. The items discussed have been completed by October 1, 1959. Five categories are outlined. They contain 25 items composed of 12 new orders, 5 improvements of existing operations, 3 manual machine control aids, and 5 items concerning "time sharing" of flip flop functions and redundant logic. Each is described; the logic and number of elements involved are presented.

I. IMPROVEMENT OF EXISTING OPERATIONS

A. Code Delete

A code delete configuration would occasionally be read by the tape reader if the address portion of the input instruction contained a zero or an "8" in the least significant position. This malfunction was caused by an error in the code delete logic. Vo was taken out of the following term, which resulted in proper operation of the code delete function. Vo is a control timing term and has no connection with the code delete operation, hence it should not have been placed in the logic shown.

Logic

$$Q3 = (T8T7T6T5T3')\phi Y1Y2Y3'Y4Y5'Y6Vo$$

Elements - Release 1 diodeB. Fast Channel Transfer (Memory to Working Storage)

One drum revolution was eliminated from all channel transfers that copy from memory to working storage. Previously during this operation the information from memory would be written into a working storage for two consecutive drum revolutions. The second revolution was eliminated because it served no known purpose; however, by this action the time given for the memory selection relays to set was shortened. This modification reduced the total execution time from an average of 96 milliseconds to 78 milliseconds, a decrease of about 19 percent. Since the channel transfer is time consuming considering other operations, the increased efficiency obtained could have a noticeable effect on total program speed.

The amount of error detection in channel transfers as indicated by alarm no. 1 was unchanged. It was determined that many errors are caused by the selection relays not relaxing in the time allotted after being set to a configuration. This erratic operation was pronounced in only a few of the 256 memory configurations. From the testing patterns used, where the time given for relaxation was varied by 35 milliseconds (2 drum revolutions), it was found that the contact chatter needed more time to be subdued than originally was provided; therefore, the reduction

of time had negligible effect on the existing chattering configurations. Since no new configurations were produced by reducing the time allowed for relay relaxation, the over-all reliability of the channel transfer was not affected. Also, by tuning the "memory read" system, the normal chattering configurations can be eliminated, thus relaxation time is not the only factor involved.

Logic

1. Change $Q7 = Q2T4'(\phi5+\phi6)$
to $Q7 = Q2\phi5T4'$
2. Change $Q3 = (A2'UtQ2F5F2'F1)\phi6T4'$
to $Q3 = Q2\phi6T4'$

Elements - Release 1 diode

C. Input Following Output

If an input instruction was placed directly after an output instruction, the last output would be taken as the first input. There was not enough time allowed between these instructions to permit the flexowriter to assume the required relaxed position. This problem was usually taken care of by programming a delay before the input instruction. Since this is an undesirable feature, the following logic has been added to create a 150 millisecond delay when desired, through the use of a "one shot" circuit. The delay is obtained by adding "2" to the left digit of the address of any output instruction. The form adopted for the

recognition of this delay was dictated by the existing input command structures.

Example: F502 - 2 hex characters out - No delay

F522 - 2 hex characters out with 150 milli-second delay

Logic

Add - Go = (T8T7T5T3)(Q4C)Ø7R6

Elements - 4 diodes

D. Stop Switch With An Optional "Jump" (1BW)

The normal-start switch which controls the lbw instruction (stop or change control) was changed to a three positional switch. The third position gives the lbw a "no operation" function. The switch's function is now equivalent to a "jump switch" with an optional stop. A third "jump switch" (2 already exist) is desired; however, the optional stop makes it more valuable because of the added ease of external program control.

Logic

Ns is the third position of the Normal-Stop Switch (+15 volts)

Add - Q3 = (T8'T7'T6'T5)(T4T3')(Ø2P33)Ns

Elements - 4 diodes and 1 Federal type switch

II. "TIME SHARING" OF FLIP FLOP FUNCTIONS AND REDUNDANT LOGIC

A. Combine Gg Function Into X1 Flip Flop

The "one step" logic for Gg was incorporated into the logic of the X1 flip flop. Hence, X1 is used for error detection in channel transfers, its previous function, and also to perform the "one step" operation. The Gg flip flop is now available for new duties.

Logic

1. Disconnect Gg logic
2. Change $X1 = P33C$
to $X1 = \phi 6(Q3C) + Pgg$
3. Add $\phi 7$ to all terms in $X1'$ logic
4. Add $X1' = Q4C$

Elements - Release 2 diodes, 1 input circuit and 1 flip flop

B. Elimination of Y_p Flip Flop

The functions of the Y_p flip flop were combined with those of Y_t, thus Y_p was made available for other activities. The following changes occurred in the machine's command structure as a result of this modification.

99 - Punch	9D - No operation
9b - Type	9F - No operation

The loss of the former operations of the 99 and 9F commands are not considered important since they can be duplicated by adjusting the switches on the flexowriter.

Logic

1. Replace Y_p' output line by Y_t output
2. Disconnect Y_p and Y_p' logic
3. Connect $T3'$ to the terms in Y_t and Y_t' that contain a T configuration

Elements - Release 2 diodes and 1 flip flop

C. Redundant Logic

The following is a list of logic that was found to be unnecessary. Since no specific investigation was performed in this area, the logic shown below was located in connection with other work. The elements released were used in the construction of modifications.

$$1. \underline{Ad} = (T8T7T5\phi 6POAn)T6'$$

This term is eliminated by $T6'An$. Also, the timing of the above term is in error, making it unnecessary.

Elements - Release 3 diodes

$$2. \underline{Bd} = (T8'T7)T5Bn$$

$(T8'T7T6)Bn$ and $(T8'T6'T5)Bn$ cause the above term to be redundant.

Elements - Release 5 diodes

$$3. \underline{Dd} = (T8T7T5')(\phi 2Vw)T2'Dn$$

$T2'Dn\phi o'$ eliminates the need for the Dd term shown.

Elements - Release 5 diodes

III. MANUAL MACHINE OPERATION AIDS

A. Command Breakpoint

The machine is stopped in the inspection period before executing a channel transfer (8X) order if the 8X switch, which is located on the breakpoint box (Fig. 1b) is placed to the "on" position. This item has proven valuable for program debugging, maintenance, and general manual machine control.

A similar item was installed for change of control or jump orders (1X). It was made obsolete by the address breakpoint modification which is described below, thus it was removed.

Logic

8X indicates the breakpoint switch position (+15 volts)

Addr - A4 = (T3T7'T6')T5'Ø18X

Elements - 5 diodes

B. Address Breakpoint

The machine may be stopped before executing the instruction located in the working storage word address (right or left) that has been placed into the breakpoint box (Fig. 1b). The box, which is placed near the control panel, contains eight switches used for the desired address, and a ninth switch that controls the execution of the

breakpoint operation. These switches are read in the same pattern as the register position lights located on the control panel. This manual method of breakpoint has proven to be versatile and useful in manual machine control, especially in program debugging. The address breakpoint is undoubtedly the most popular modification of the group developed.

Logically only 11 diodes are necessary for this item. However, noise was developed in the cable used to "remote" this operation, therefore the number of diodes required is increased to 19, and they must be placed at the "logic cabinet" end of the cable. The large reverse resistance of the diodes effectively increases the cable's time constant, preventing noise formation.

Logic

E_i - True position of an address switch
 E_i' - False position of an address switch
 BP - Breakpoint operational switch (+15 Volts)

$$\text{Add A4} = (S1E1 + S1'E1')(S2E2 + S2'E2') \\ \dots (S8E8 + S8'E8')\phi 1BP$$

Elements - 17 diodes (16 not in logic rack)

C. Subroutine Protection

Sixty-three main memory channels (01 through 3F) are used to store subroutines. In order to record information in any of these channels, alarm switch #1 must be

placed in the "restore" position. Since this switch is normally in another position, subroutines will not accidentally be destroyed. The input equipment is slow in operation, therefore the replacing of subroutines in memory is time consuming and bothersome. This protection has proven to be nearly 100 percent effective, and is extremely valuable because a large number of inexperienced people use the machine.

Logic

Change $A2 = (T8T7'T6'T5)T4(R8'R7'R6'R5'R4'R3'R2'R1)$
 $(\phi_0'C)\phi_1'$

to $A2 = (T8T7'T6'T5)T4(R8'R7')(R6+R5+R4+R3+R2+R1)$
 $(\phi_0'C)\phi_1'$

Elements - Release 2 diodes

IV. ORDERS WITH NEW CONCEPTS

The following group of orders are considered a valuable addition because they are extremely useful and their concepts are new to the machine's order structure.

A. Count E Up and Compare Magnitude (16N)

If Z is "on", the machine will stop. If Z is "off", the instruction will be performed as shown:

1. Turn Z on when the address portion (N) is equal to the address portion of D (Z on if $E-N=0$)
2. Count E up by 1 ($|E|+1$)

3. There is no "jump" with this order as with the others in the 1X group.

This instruction has been well received because of its counting ability. Previously, E could only count in a negative direction, thus the "B Box" is more versatile with this added function. The comparison of the magnitude of N and E gives this order more depth and has proven useful. The loss of the modified 17W order has not been noticed. Since it concerned a "double" address modification process, it was never used.

Logic

1. Stop if Z is on

$$\text{Add} - A4 = (T8'T7'T6'T5T4'T3T2)T1'(\phi_1Z)$$

2. Z on if $N - E = 0$

$$\text{Add} - Z = (T8'T7'T6'T5T4'T3T2)T1'(R1'R2'R3'R4'R5'R6')(R7'R8')(\phi_2C)$$

3. No jump with 16N

Add T1 to the following terms

$$U_k = (T8'T7'T6'T5T4'T3T2)En'\phi_2Uc\ C + (T8'T7'T6'T5T4'T3T2)En\ (F4P33')(\phi_2C)Uc'$$

4. $|E| + 1$

- a. Add T1 to the following term

$$Ua' = (T8'T7'T6'T5T4'T3T2)\phi_2(EnC)$$

- b. Add - Ua' = $(T8'T7'T6'T5T4'T3T2)\phi_2(En'C)T1'$

Elements - 18 diodes and 2 input circuits

B. One Character Output (DFXX)

One character as determined by the address portion of this instruction will be typed or punched. Numeric, alphabetic, flex control characters, etc., may be used. The programmer is provided with a convenient instruction to employ in format control. No extra storage is needed for the character used as with the standard output instructions, and the A register is not involved. The 150 millisecond "programmed delay" associated with output instructions may be used. The six bit character configuration is identical to the existing codes, except the sixth bit is placed into the seventh bit position since the sixth bit is used to indicate the output delay. This order reduces the complexity of alpha output, therefore it is very useful in the coding course taught because the novice student may attain format control without having to be introduced to the awkward alpha output order.

Logic

$i = 1, 2, \dots, 5$

1. Add - $Y_i = (T_8 T_7 T_5)(T_4 T_2)(Q_3 C_6) R_1$

2. Add - $Y_6 = (T_8 T_7 T_5)(T_4 T_2)(Q_3 C_6) R_7$

3. Change the D.C. "or" gate in the Y flip flop logic as shown:

Change $(T_2' + T_2 P_1')$

to $(T_2' + T_4' P_1')$

Elements - 15 diodes and 6 input circuits

C. "Flag" Operations (D9XY, D1W, D3W, DBW)

The two flip flops that were formally Gg and Yp are used as "Flags", and a two bit counter. The flag concept is similar to the operations concerning the Z flip flop excluding the "arithmetic" and "comparison" dependency. The two flags are available at all times, and may be used to isolate various levels in programming by acting as change of control indicators. This function is performed by setting a flag to the "on" state (D9XY) and then changing control (jumping) to a new position in the program (D1W, D3W), as determined by the flags position without affecting the state of the flag involved. These are two individual operations, thus in a program, the jump may occur at any point desired.

Initially, the O6 and O7 orders were installed to set the flags, where O6 and O7 would reverse the states of flags #1 and #2, respectively. It was found through programming tests conducted during the evaluation period, that the reverse function of the orders were not sufficient for setting a flag to a desired state, therefore the D9XY order was formed. By incorporating the reverse flags feature and specific position selection into this order, the need of the O6 and O7 orders was eliminated and they were discarded.

As stated previously, the flags may be used as a counter. The operation is patterned after the index register counting instruction (17W). A configuration is placed into the counter (D9XY). Upon execution of the counting instruction (DBW) one is subtracted and a jump to W takes place except when the result of the counting is zero. A maximum count of four is obtainable, which may be conveniently used for program loop control, format control, etc. Since the flag and counting operations utilize the same flip flop units, they must be used independently of one another. In order to program both operations effectively, flag configurations may have to be "stored" and "restored". These functions are performed by a group of instructions.

The flag and counting operations have been widely accepted and have proven to be one of the most valuable groups of orders developed. Utilization of the same elements for two different functions, through "time sharing", increases the efficiency of the elements used. The counting order is very popular and has been placed into the "Start Routine" as a format control for input-output of information. Another distinct feature of this group is the sharing of the functions of the set flag order (D9XY), by the counting order (DBW) and "jumping by the flag" orders (D1W, D3W).

The orders involved in the flag and counting operations are reviewed below.

1. D9XY - Set Flags to desired State.

XY in the address portion of this instruction controls the states of the flags as follows:

Y is 1 - Flag #1 is set to the on state.

X is 1 - Flag #2 is set to the on state.

Y is 2 - Flag #1 is set to the off state.

X is 2 - Flag #2 is set to the off state.

Y is 3 - Flag #1 reverses state.

X is 3 - Flag #2 reverses state.

Y is 0 - Flag #1 is not affected.

X is 0 - Flag #2 is not affected.

Thus, the programmer may easily set the flags to a desired position within his program. It should be noted that the address or R configuration is used for order control, which is unusual, but convenient in this situation.

Logic

a. Add - Fg1 = (T8T7T5)(T6'T3')T4T2'(\emptyset 2C)R1

b. Add - Fg1' = (T8T7T5)(T6'T3')T4T2'(\emptyset 2C)R2 + SC

c. Add - Fg2 = (T8T7T5)(T6'T3')T4T2'(\emptyset 2C)R5

d. Add - Fg2' = (T8T7T5)(T6'T3')T4T2'(\emptyset 2C)R6 + SC

Note: The Q4 logic addition shown for the D1 and D3 instructions satisfy the need of the D9. The clear switch sets both flags to "off".

Elements - 12 diodes and 4 input circuits

2. D1W - jump to W if flag #1 is on.

D3W - jump to W if flag #2 is on.

These instructions are controlled "jumps". The state of the flags are not affected.

Logic

a. Add - $Q4 = (T8T7T5)(T6'T3')\phi3$

b. Add (D1) - $Uk = (T8T7T5)T6'T4'T3'T2'(\phi2C)Fg1$

c. Add (D3) - $Uk = (T8T7T5)T6'T4'T3'T2'(\phi2C)Fg2$

Elements - 16 diodes and 2 input circuits

3. DEW - Count Flags by -1 and jump to W except when the result of the count is zero.

Logic

a. Add - $Fg1 = (T8T7T6'T5T3')(T4T2')(\phi2C)Fg1'$

b. Add - $Fg1' = (T8T7T6'T5T3')(T4T2')(\phi2C)Fg1$

c. Add - $Fg2 = (T8T7T6'T5T3')(T4T2')(\phi2C)Fg1'Fg2'$

d. Add - $Fg2' = (T8T7T6'T5T3')(T4T2')(\phi2C)Fg1Fg2'$

e. Add - $Uk = (T8T7T6'T5T3')(T4T2')(\phi2C)(Fg1'+Fg2)$

Elements - 17 diodes and 5 input circuits

Total number of elements used for counting and flag operations - 45 diodes, 11 input circuits and 2 flip flops

D. Count A by 1 (24, 26)

The A register is made into a convenient counter by the addition of the 26 ($|A| + 1$) and 24 ($|A| - 1$) orders. Since these orders may be doubled, their usefulness is

increased. The machine now has a 32 bit counter (A register), a 16 bit counter (E register), and a 2 bit counter (Flags). A change of control exists directly or indirectly with each counter increasing their usefulness. Previously, the only counter available was the 16 bit "B Box" (E register), and it only counted in the negative direction. The machine is now more flexible and efficient with three separate counters.

Logic

1. Add T2 to the following term

$$Ua' = (T8'T7'T6T5'T4')(\emptyset 2)An'C$$

2. Add - Ua = (T8'T7'T6T5'T4') $\emptyset 2$ (POC)T3

3. Add - Ua' = (T8'T7'T6T5'T4') $\emptyset 2$ AnCT2'

Elements - 7 diodes and 2 input circuits

V. CONVENIENT ORDERS

The following is a group of new orders that have been added to the machine's structure because they are useful and relatively few elements were needed for their construction.

A. Exchange B and W (45W)

The addition of this order completes the B register sequence of orders concerning the working storage (41-45-C5). The B register is now as convenient to use in this respect as the A register with its sequence of

(49-69-79) orders. The 45W has proven to be valuable in increasing the speed of "packed" subroutines.

Logic

1. Add - Bd = (T8'T7)(∅2Vw')Bn
2. Take T8 out of the following terms:

$$Q8 = (T8T7T6'T5'T4)T3T2'BeC$$

$$Q8' = (T8T7T6'T5'T4')T3T2'Be'C$$

$$Q7 = (T8T7T6'T5'T4')T3(\emptyset 2Vw)$$

3. Add T4' to the following terms:

$$Q8 = (T8'T7T5')AeC$$

$$Q8' = (T8'T7T5')Ae'C$$

Elements - 9 diodes

B. Absolute Addition between A and W (73W)

The 73W instruction will perform an absolute addition between the A register and W. The sign of A is unchanged. This instruction uses the present logic of the "63W" for the addition involved. It does this by changing the 73 configuration to a 63 after the sign of A has been preserved. This is a unique method of obtaining a new instruction with a minimum of components, and was possible because the 73 order code was previously not used.

Logic

1. Add T2' to the following term:

$$Ad = (T8'T7T6T5T4'\emptyset 2Vw)T3'WnDnP33'$$

2. Add - Ad = (T8'T7T6T5T4'\emptyset 2Vw)(AnPo)T2

3. Add - T5' = (T8'T7T6T5T4'Ø2Vw)(POC)T2

Elements - 8 diodes and 1 input circuit

C. Compare Absolute Values of A and W (53W)

Z will be turned on if the absolute value of A is not equal to the absolute value of W. This order completes the sequence for comparing the absolute values of A and W as shown below.

Z on if $|A| < |W|$ -51W

Z on if $|A| > |W|$ -51W 0200 (2 instructions)

Z on if $|A| \neq |W|$ -53W

Logic

1. Remove T2' from the following term:

$$Z = (T8'T7T6'T5T4'T2'Ø2Vw)(Po'P33')(An'C)WnT3'$$

2. Add - Z = (T8'T7T6'T5T4')Ø2Vw)(PO'P33')(AnC)

$$Wn'T3'T2$$

Elements - 3 diodes and 1 input circuit

D. B and D Registers (08, 0A)

The 08 order copies D into B and the 0A exchanges D and B. These orders give the programmer a direct means of using the B and D registers together. This action has helped to increase the speed of "packed" subroutines, notable, "Floating Point Arithmetic", where a reduction of approximately 60 milliseconds (33 percent) was attained. The previous 08 and 0A commands that were concerned with

the E register were not used often, therefore their loss is of no consequence.

Logic

1. Change $Bd = (T8'T7')(T6'T5'T4)(T3'T2)\emptyset 2EnF4$
to $Bd = (T8'T7')(T6'T5'T4)T3'\emptyset 2Dn$
2. Change $Dd = (T8'T7')(T6'T5')(T4T3')T2\emptyset 2EnF4$
to $Dd = (T8'T7')(T6'T5')(T4T3')T2\emptyset 2Bn$

Elements - Release 2 diodes

F. Flags Used as Optional Jump Switches

Two switches were installed near the control panel which permit the operator to manually control the states of the flags. Thus, the flags may be used as optional jump switches giving a total of five. Two neon indicators are also present to indicate the position of the flags to aid in machine control for debugging and other purposes. This feature gives the flags a third option in operation.

Elements - 4 input circuits

CONCLUSIONS

The following is a list of conclusions that resulted from the work performed.

1. Digital computers may be classified into four categories: the business machine or data processor, the university type for scientific computations,

the process control machine, and the general purpose computer.

2. The general purpose machine may be subdivided into the large and medium types.
3. The operation of a machine is primarily serial or concurrent (parallel), or a combination of both.
4. Floating point arithmetic is usually used for scientific computations.
5. The ALWAC III-E is composed of four systems: memory, input-output, arithmetic, and control.
6. The basic theme of design of the III-E is simplicity and economy.
7. Since the input-output system is serial in nature, it is very inefficient.
8. The memory system is cumbersome to use and is slow in operation because the main section of 8192 words is not directly addressable, and a relay selection network is used in the block transfer operation.
9. The "B Box", continuous location of the words in working storage, and the temporary storage of every other instruction in a quick access register, are the outstanding features of the control system.
10. The large and versatile group of orders that compose the order code help to reduce the labor of

programming, and tend to make machine operation more efficient.

11. The logic equation notation used to describe the III-E structure is very effective.
12. The principle application of the III-E at the Oregon State Digital Computer Laboratory is for education, and program development.
13. Programming labor was reduced by developing a group of new convenient orders.
14. "Debugging" time on the machine was decreased by the installation of manual "break point" operations.
15. Machine operation efficiency was increased by adding new functions and improving existing ones.
16. The time involved in an installation of a modification can be approximated by the number of connections to be made, where each takes from 10 to 15 minutes to complete.

Additional Activities

The author's association with the III-E may be divided into three phases. From March 1957 to October 1958, maintenance of the machine was the prime objective. To perform this function satisfactory with student labor*, preventative and active maintenance procedures were developed. Since no technical manual concerning the machine was available, one was written by the author, and it has been used successfully as a training aid at the O.S.C. laboratory and other III-E installations throughout the U.S.A., Canada, and Europe.

The investigation of the III-E and the modifications developed, as described in this thesis, were performed from October 1958 to October 1959. The bulk of the work was achieved after June 1959. A National Science Foundation Grant for research in digital computer systems was received in July 1959, thus a portion of the work was supported by this grant.

Starting in August 1959, and extending to June 1960, a new project was undertaken. It is primarily concerned with increasing the efficiency of floating point arithmetic operation. Automatic operation of the arithmetic functions was found to be too expensive at this time,

* The author was a Senior in the School of Electrical Engineering at this time.

therefore another approach was taken. Special modifications (five items) were designed to increase the speed of the floating point arithmetic operations as performed by a subroutine; and the programming labor involved when using floating point arithmetic was to be reduced.

Through the five items formulated the operation time of all floating point arithmetic functions will be decreased approximately by a factor of 3.5. Their speeds will then be comparable to those of fixed point arithmetic operations (shifting for scaling is included in fixed point time). Programming with floating point arithmetic will be simpler than that of the fixed point type because scaling is not required and the need of the "key word", which supplies control information for the subroutine, has been eliminated. Three of the five items are also extremely useful in all phases of programming. An extension of the rack in the logic cabinet was constructed and contains the 21 plug-in boards that are needed for the modifications. The group formed is shown below. The first four have been successfully installed by January 1960.

1. Addressable Registers - All one word registers are made addressable, resulting in more efficient and convenient operation because the use of the working storage is decreased.

2. Floating Point Comparison of the Magnitudes of A and W (7BW)- Z is turned on if $|A| < |W|$ with a floating point word construction as follows:
Bits 1 to 8 (characteristic), bits 9 to 32 (fraction).
3. Unpack and Repack Floating Point Numbers (7D & 7F)- The fraction and characteristic of a floating point number constructed as shown in item two, are separated or combined in one word time.
4. Mark Place and Return to the Marked Place (9DW & 9F)- The 9DW instruction stores the address of the next instruction in normal sequence and "jumps" to W. Upon the execution of the 9F order, the next instruction is taken from the "Mark Place" address that has previously been stored. These two orders have eliminated the need for "key words" in the operation of most subroutines.
5. C Register - A new one word register will be employed for quick access storage. Thus, four one word registers will be available and may be used together effectively without referring to working storage because each is addressable.

The activities of the immediate future include the completion of the fifth item listed, and the testing and evaluation of the entire group. Also, the design of a

memory preselct system which would make the main memory directly addressable in banks of 32 is contemplated.

BIBLIOGRAPHY

1. Alwac Computer Division, El-Tronics Inc. Manual of operation for the ALWAC III-E. Hawthorne, California, 1957. 67 p.
2. Automatic control for industrial process plants. Datamation 3(7):39-40. October 1957.
3. Dubinski, George A. Maintenance manual for the Alwac III-E digital computer #20. Corvallis, Oregon, Oregon State College, Department of Mathematics, 1958. 178 p.
4. Halstead, W. K., et al. Purpose and application of the RCA Bizmac System. American Institute of Electrical Engineers, Proceedings of The Western Joint Computer Conference T-85:119-123. February 7-9, 1956.
5. Humphrey, Watts S., Jr. Switching circuits with computer applications. New York, McGraw-Hill, 1956. 264 p.
6. Millman, Jacob and Herbert Taub. Pulse and digital circuits. New York, McGraw-Hill, 1956. 687 p.
7. Thister, Montgomery, Jr. Logical design of digital computers. New York, Wiley, 1958. 408 p.
8. Richards, R. K. Arithmetic operation in digital computer. New York, Van Nostrand, 1958. 391 p.
9. Richards, R. K. Digital computer components and circuits. New York, Van Nostrand, 1958. 511 p.
10. Smith, Charles V. L. Electronic digital computers. New York, McGraw-Hill, 1959. 443 p.
11. Wanless, C. L. Transistor circuitry for digital computers. Institute of Radio Engineers, Transactions on Electronic Computers Ec-4(1):11-15. 1955.
12. Weck, Martin H. A second survey of domestic electronic digital computing systems. Aberdeen Proving Ground, Maryland, Ballistic Research Laboratories, 1957. 453 p.

APPENDICES

TABLE VIII. B REGISTER RECIRCULATION CHART

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	8	8	78	78	8	8	78	78									0
1	5	5	57	57	5	5	57	57	5	5	5	5	5	5	5	5	5 1
2	8	8	78	78	28	28	278	278	6	6	6	6	6	6	6	6	6 2
3			7	7	2	2	27	27	6	6	6	6	6	6	6	6	6 3
4			7	7			7	7	3	3	3	3	3	3	3	3	3 4
5	5	5	57	57	5	5	57	57	35	35	35	35	35	35	35	35	35 5
6	4	4	47	47	4	4	47	47	346	346	346	346	346	346	346	346	346 6
7	4	4	47	47	4	4	47	47	346	346	346	346	346	346	346	346	346 7
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1 8
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1 9
A					2	2	2	2									A
B					2	2	2	2									B
C	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10 C
D	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9 D
E																	E
F	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9 F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Vertical column indicates the orders least significant digit.
Horizontal column indicates the orders most significant digit.

Recirculation Logic Terms

1. $T_8 T_7 T_6'$
2. $T_7 T_6 T_4 T_3$
3. $T_8 T_7 T_4$
4. $T_8 T_7 T_6$
5. $T_8 T_6 T_5$
6. $T_8 T_6 T_4$
7. $T_8 T_4 T_2$
8. $T_8 T_7 T_5 T_4'$
9. $T_8 T_7 T_5$
10. $T_8 T_7 T_6 T_5'$

Note: The recirculation chart for the A register is not presented because only one term (T_6') is involved, making it trivial.

TABLE IX. D REGISTER RECIRCULATION CHART

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	237	237	23	23	123	123	123	123	7	7			157	157	15	15 0
					57	57	5	5								
1	237	237	23	23	123	123	123	123	7	7	8	8	157	157	15	15 1
					57	57	5	5								
2	237	237	23	23	123	123	123	123	7	7			157	157	15	15 2
					57	57	5	5								
3	237	237	23	23	123	123	123	123	7	7			157	157	15	15 3
					57	57	5	5								
4	37	37	3	3	137	137	13	13	7	7			17	17	1	1 4
5	37	37	3	3	137	137	13	13	7	7			17	17	1	1 5
6	37	37	3	3	137	137	13	13	7	7			17	17	1	1 6
7	37	37	3	3	137	137	13	13	7	7			17	17	1	1 7
8	267	267	26	26	256	256	256	256	67	67	6	6	567	567	56	56 8
					7	7										
9	267	267	26	26	256	256	256	256	67	67	6	6	567	567	56	56 9
					7	7										
A	27	27	2	2	257	257	25	25	7	7			57	57	5	5 A
B	27	27	2	2	257	257	25	25	7	7			57	57	5	5 B
C	79	79	9	9	79	79	9	9	79	79	9	9	79	79	9	9 C
D	47	47	4	4	47	47	4	4	47	47	4	4	47	47	4	4 D
E	7	7			7	7			7	7			7	7		E
F	7	7			7	7			7	7			7	7		F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Vertical column indicates the orders least significant digit.
Horizontal column indicates the orders most significant digit.

Recirculation Logic Terms

1. T8'T3
2. T7'T4'
3. T8'T4'
4. T8T7T5
5. T7'T3
6. T8T7'T6'
7. T2'
8. T8'T7'T6'T5T4T3'T2
9. T8T7T6'T5'

TABLE X. E REGISTER RECIRCULATION CHART

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	245	245	24	24	25	25	2	2	234	234	234	234	235	235	23	23	0
									5	5							
1	45	45	4	4	5	5			345	345	34	34	35	35	3	3	1
2	246	246	24	24	26	26	2	2	234	234	234	234	23	23	23	23	2
									6	6			6	6			
3	456	456	4	4	56	56			345	345	34	34	35	35	3	3	3
									6	6			6	6			
4	24	24	24	24	2	2	2	2	234	234	234	234	23	23	23	23	4
5	4	4	4	4					34	34	34	34	3	3	3	3	5
6	246	246	24	24	26	26	2	2	234	234	234	234	236	236	23	23	6
									6	6							
7	46	46	4	4	6	6			346	346	34	34	36	36	3	3	7
8	124	124	124	124	125	125	12	12	123	123	123	123	123	123	123	123	8
	5	5							45	45	4	4	5	5			
9	145	145	14	14	15	15	1	1	134	134	134	134	135	135	13	13	9
									5	5							
A	124	124	124	124	125	125	12	12	123	123	123	123	123	123	123	123	A
	56	56			6	6			456	456	4	4	56	56			
B	145	145	14	14	156	156	1	1	134	134	134	134	135	135	13	13	B
	6	6							56	56			6	6			
C	124	124	124	124	12	12	12	12	123	123	123	123	123	123	123	123	C
									4	4	4	4					
D	14	14	14	14	1	1	1	1	134	134	134	134	13	13	13	13	D
E	124	124	124	124	126	126	12	12	123	123	123	123	123	123	123	123	E
	6	6							46	46	4	4	6	6			
F	146	146	14	14	16	16	1	1	134	134	134	134	136	136	13	13	F
									6	6							
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Vertical column indicates the orders least significant digit.
Horizontal column indicates the orders most significant digit.

Recirculation Logic Terms

1. T8
2. T5'
3. T4
4. T3'
5. T7'T2'
6. T6T2'