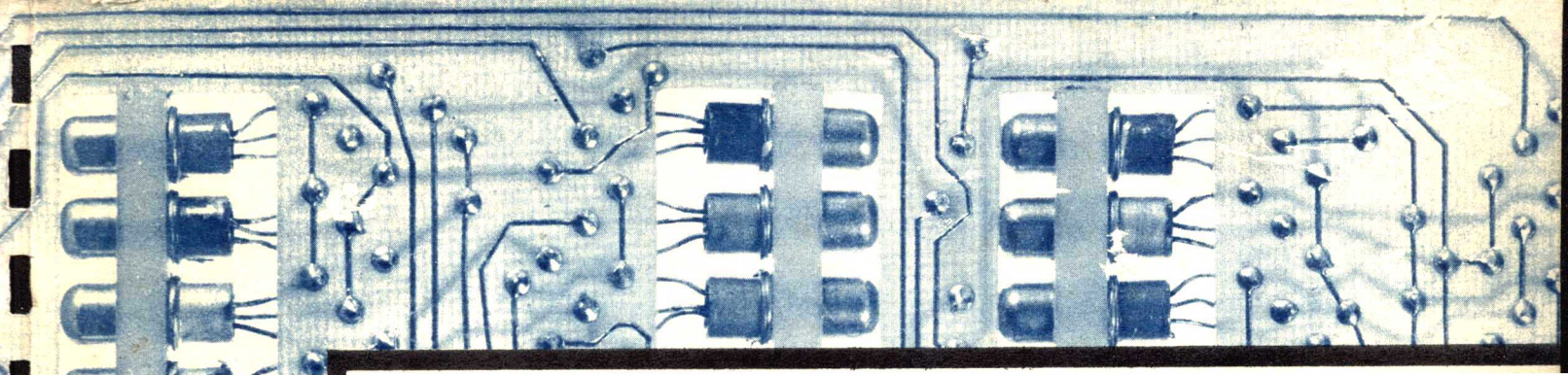
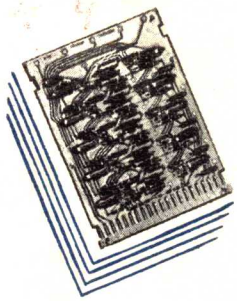


TRANSAC S-2000
SYSTEM DESCRIPTION



PHILCO CORPORATION

COMPUTER research design manufacture

TRANSAC S-2000

SYSTEM DESCRIPTION

January 1958

PHILCO CORPORATION
GOVERNMENT AND INDUSTRIAL DIVISION
Philadelphia, Penna.

TRANSAC S-2000 SYSTEM DESCRIPTION

TABLE OF CONTENTS

APPENDICES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont'd)

S-2000 Computer System Installation

TRANSAC S-2000

SYSTEM DESCRIPTION

1. INTRODUCTION

The TRANSAC S-2000 System is Philco's answer to the challenge of the modern world for more advanced digital computing and data processing systems. It embodies new exclusive concepts. The S-2000 is designed for optimum utilization of transistor characteristics. It is not a transistorized version of a vacuum tube system.

1.1 Advanced System Design

The TRANSAC S-2000 solves business, scientific and control problems. Its speed of operation keeps pace with the data processing requirements of today and tomorrow. Its unique features of expansibility permit a basic installation to be increased in capacity to accommodate more complex or demanding problems at any time after installation without redesigning the existing system.

1.2 Reliability

The reliability achievements of the S-2000 are made possible by the Philco Surface-Barrier Transistor, the active electronic element in the S-2000. The 90% life expectancy of these transistors is well over 200,000 hours. Such transistors are 100 times more reliable than either diodes or vacuum tubes.

1.3 High-Speed Operation

The use of asynchronous parallel transistor logic results in an equivalent clock rate of 10 megacycles. The average time required to add two 15 decimal digit numbers is less than 1 microsecond.

1.4 Multiple Sequencing

The rate of data handling is further increased by processing as many as four magnetic tape units simultaneously. A system using this technique is capable of processing up to 360,000 binary coded decimal (alpha numeric) characters per second.

1.5 Compactness

A typical large-scale S-2000 occupies only 600 square feet including working and access area or about 1/4 of the floor area required by equivalent vacuum tube systems.

1.6 Economy

This S-2000 Computer requires less than 2 kilowatts of power; or about 1/10 the power required by a vacuum tube computer. There is no need for costly air conditioning or costly power distribution systems.

1.7 Versatility

The programmer may choose from 64 arithmetic, 16 transfer of data, 14 shift, 16 transfer of control, and 16 special instructions. The instructions have been carefully designed to produce ease in programming and efficiency in the use of Compilers.

1.8 Simplicity

Electrical and mechanical modular design results in simplicity of fabrication and maintenance. All logical functions are performed by two Philco developed circuits; common emitter and common collector type switches.

1.9 Ease of Servicing

The reliability of Surface Barrier transistor reduces the frequency of component replacement. Fullest use is made of automatic diagnostic techniques for rapid preventive maintenance. Modular assemblies are easily accessible for quick replacement.

1.10 Human Engineering

Controls are designed and positioned to minimize operator fatigue. Color styling increases operator efficiency.

2. TRANSAC S-2000 SYSTEM

A TRANSAC S-2000 installation consists of:

2.1 Computing and Control Unit MF-2000

Optional Features:

(a) Floating point and/or fixed point arithmetic FP-20
(b) Variable number of index registers; 4 to 16 index registers in groups of 4 (IR-2004, 8, 12, 16)

2.2 Magnetic Core Storage Unit MC-2001 to 16

4096 Word by 48 Bit capacity per Unit.
Expansible up to 8 Units. (32, 768 Words)

2.3 On-Line Input-Output Optional Equipment

2.3.1 Magnetic Tape System

(a) Magnetic Tape Units (up to 256) TT-2150
(b) Magnetic Tape Control Unit TC-2150
(c) Magnetic Tape Sequencing Unit TS-2150

2.3.2 Magnetic Drum System

(a) Magnetic Drum MD-2032
(b) Magnetic Drum DC-2032

2.3.3. On-Line Punch Card System PC-2801

(a) Card Reader
(b) Card Punch
(c) Card Controller

2.3.4 On-Line Paper Tape Console

(a) High Speed Paper Tape Reader
(b) High Speed Paper Tape Punch
(c) Flexowriter

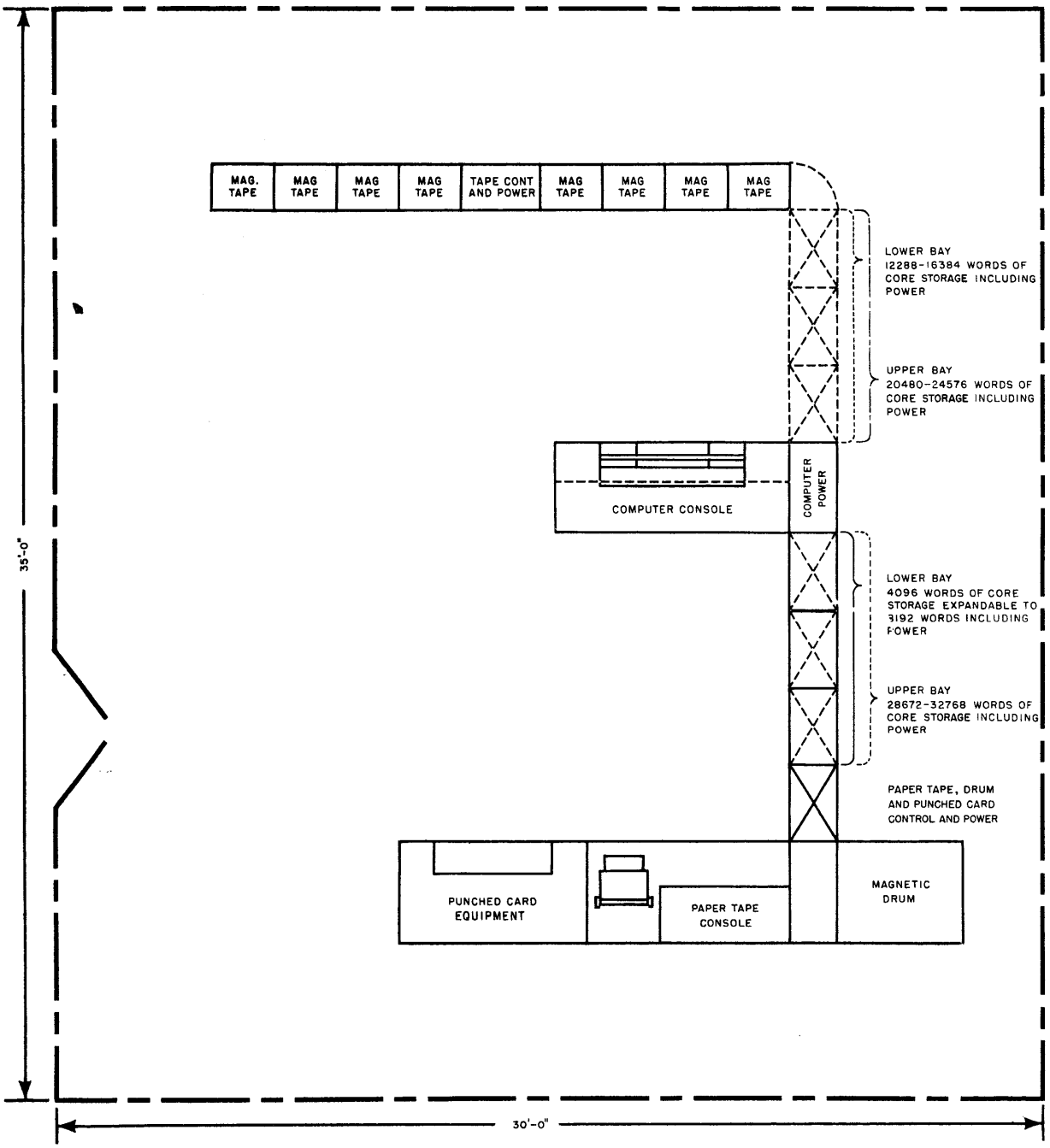
2.3.5 Console Typewriter CT-2000

2.4 Off-Line Input-Output Optional Equipment

2.4.1 High Speed Printer CPS-2912

2.4.2 Card to Magnetic Tape - Magnetic Tape to Card Converter CPC-2802

These devices are described in detail in the following sections. A typical S-2000 installation is illustrated in Figure 1.



TYPICAL TRANSAC S-2000 SYSTEM LAYOUT
(ON LINE EQUIPMENT)

FIG. 1

3. COMPUTING AND CONTROL UNIT MF-2000

The Computing and Control Unit performs the arithmetic and programming operations for the system.

3.1 Arithmetic Operations

Binary arithmetic is performed with a parallel logic. Negative numbers are represented in the two's complement form. Numbers are handled in an algebraic sense. For fixed point arithmetic, numbers are normalized such that they are in magnitude less than unity and equal to or greater than -1. The register arrangement for fixed point arithmetic is illustrated in Figure 2. Each register has a capacity of 48 bits. The far left bit designates the sign, and the remaining 47 bits contain the magnitude of the number.

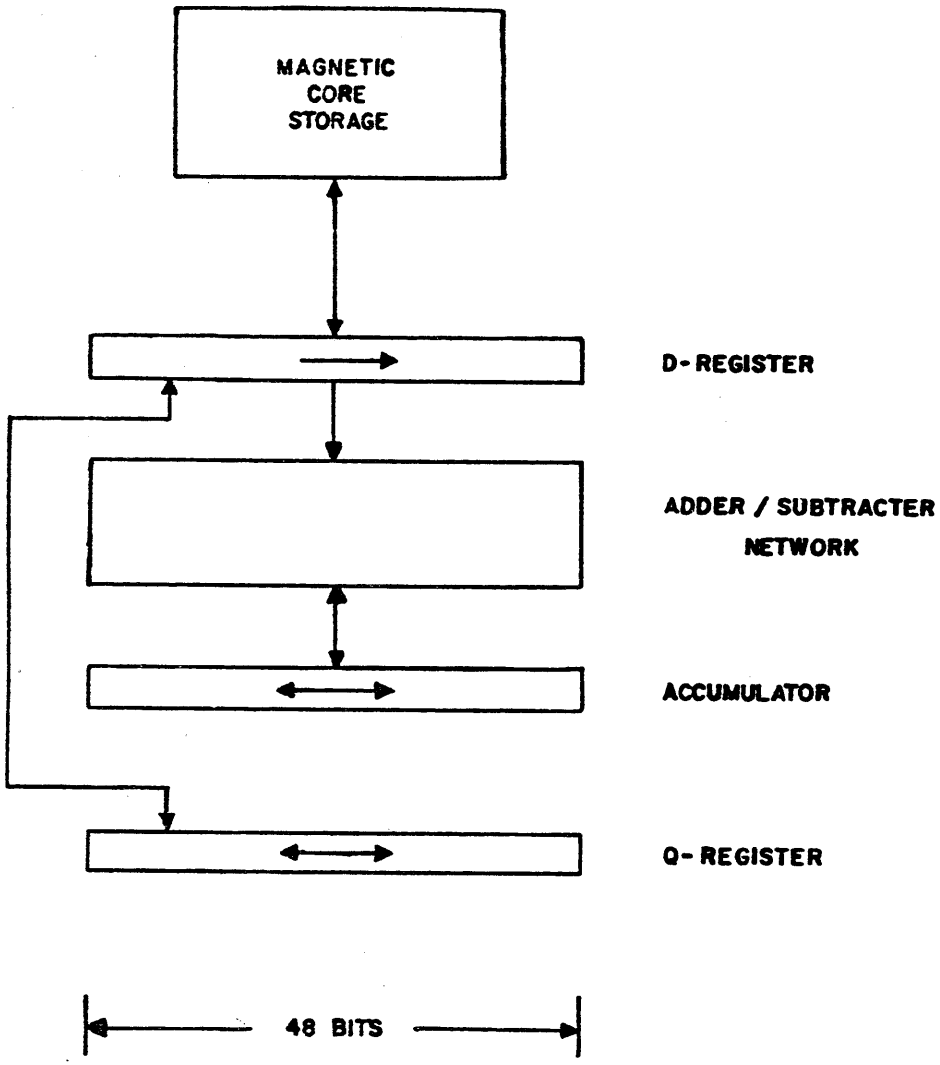
Floating point numbers consist of a mantissa (magnitude) and the characteristic (exponent), where both are represented in the two's complement form. The mantissa is normalized such that in absolute value it is less than unity but greater than one-half. The register arrangement for floating point arithmetic is illustrated in Figure 3. There are 36 bits allotted to the mantissa and 12 bits allotted to the characteristic.

In either case, the operations performed in the arithmetic section include:

1. Addition: Contents of A-Register added to contents of D-Register, with sum stored in the A-Register.
2. Subtraction: Contents of D-Register subtracted from contents of A-Register, with difference stored in the A-Register.
3. Multiplication: Contents of the D-Register multiplied by the contents of the Q-Register, with the product stored in the A and Q-Registers.
4. Division: Contents of the A-Register divided by the contents of the D-Register, with the quotient stored in the Q-Register and the remainder stored in the A-Register.

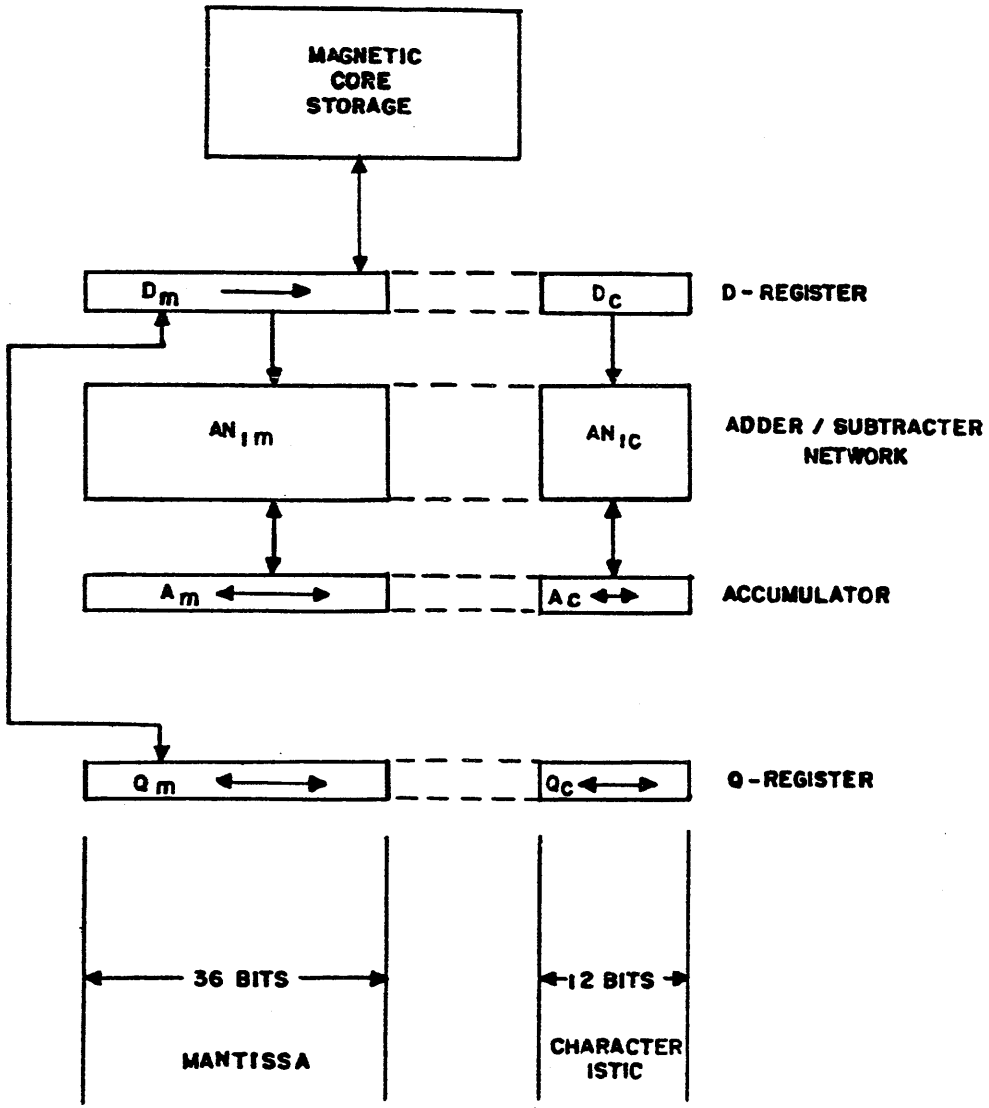
3.2 Speed of Arithmetic Operations

Some of the arithmetic instructions obtain operands from the magnetic core storage and others use operands stored in the registers of the arithmetic section. The speed of operation is determined by the access time of the magnetic core storage and the switching speed of the arithmetic instructions. The following tables illustrate some of the arithmetic speeds.



BLOCK DIAGRAM OF FIXED POINT ARITHMETIC SECTION

FIG. 2



BLOCK DIAGRAM OF FLOATING POINT ARITHMETIC SECTION

FIG. 3

Operands Initially in Arithmetic Registers

Operation		Fixed Point	Floating Point
Addition/and Subtraction	Av.	1.0 μsec	8 μsec
	Min.	0.5 μsec	0.5 μsec
	Max.	5.1 μsec	19 μsec
Multiplication and Division	Av.	28 μsec	22 μsec
	Min.	19 μsec	15 μsec
	Max.	245 μsec	140 μsec

Operands Transferred from Magnetic Core Storage

Operation		Fixed Point	Floating Point
Addition and Subtraction	Av.	12 μsec	20 μsec
	Min.	12 μsec	13 μsec
	Max.	12 μsec	41 μsec
Multiplication and Division	Av.	33 μsec	27 μsec
	Min.	24 μsec	20 μsec
	Max.	250 μsec	145 μsec

Result Replaces Operand in Magnetic Core Storage

Operation		Fixed Point	Floating Point
Addition and Subtraction	Av.	13 μsec	20 μsec
	Min.	12.5 μsec	13 μsec
	Max.	17 μsec	31 μsec
Multiplication and Division	Av.	40 μsec	34 μsec
	Min.	31 μsec	27 μsec
	Max.	257 μsec	152 μsec

3.3 Programming Operations

Programming instructions are read from the magnetic core storage and stored in the PR-Register for interpretation and execution. The PR-Register has a capacity of 48 binary bits. An instruction word requires 24 binary bits, which includes the command, and address information. Single address instructions are used. The PR-Register therefore contains two instructions, side by side. The left-hand instruction is executed first except for the case where only the right-hand instruction is to be executed.

The instructions of the program are sequenced by a Program Address Register (PA).

3.4 Addressing and Index Registers

The format for an instruction word is illustrated in Figure 4. Eight bits determine the command (function of the instruction), one bit designates whether index registers are used or not, fifteen bits are used to address the magnetic core storage and designate which index register is involved. When the word capacity of the magnetic core storage is 4096 and 8 index registers are used, twelve bits address the magnetic core storage and 3 bits designate which index register is involved. When index registers are involved, the contents of the designated index register is added to the 12-bit address portion of the instruction to address the magnetic core storage.

When 8192 words of magnetic core storage are used in the system, the same instruction word format is used except that 13 bits instead of 12 bits are necessary to address the magnetic core storage. In this case, if index registers are not involved, 13 bits are taken from the addressing portion of the instruction. If an index register is involved, 12 bits of the address portion of the instruction are added to a 13-bit index register to perform the addressing.

Besides address modification, the index registers are used for counting, address placement or substitution, or accumulating.

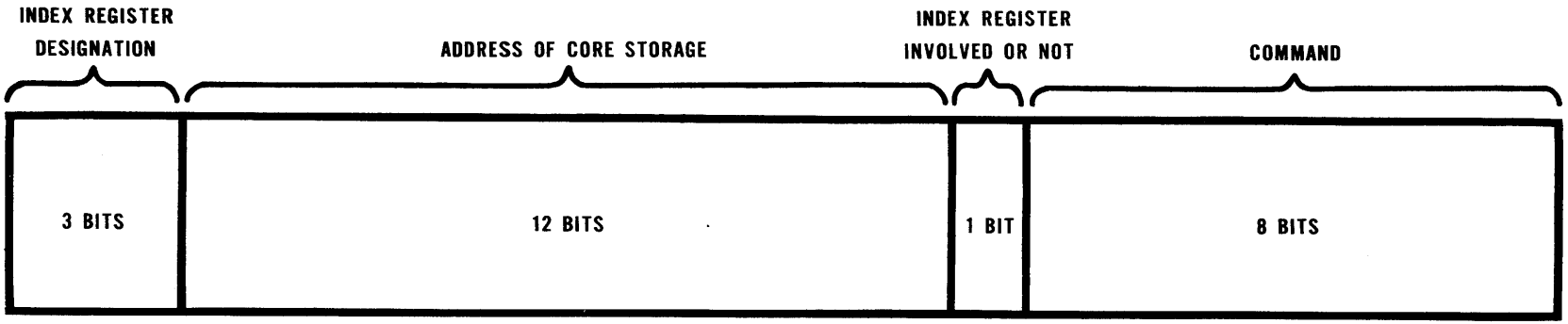
3.5 Computer Instructions

Computer instructions are subdivided into five groups:

1. Arithmetic,
2. Transfer of data,
3. Shift,
4. Transfer of control (jump),
5. Special.

To describe the instructions in detail, a symbolism is adopted, which is explained as follows:

A	refers to the A-Register
D	refers to the D-Register
Q	refers to the Q-Register
V	refers to the magnetic core storage.
||	signifies 'absolute value of'.
( )	signifies 'contents of'.



COMPUTER INSTRUCTION WORD FORMAT

FIG 4

$\longrightarrow$ signifies 'place in'.

d_j designates the j^{th} bit of D, where j=0 for the sign position, and

j= -1 through -47 for the remaining positions to the right of the sign.

3.5.1 Arithmetic Instructions

There are a total of 32 addition and subtraction instructions which may be summarized as follows:

$$(A) \pm [(V) \text{ or } (Q)] \longrightarrow [A \text{ or } A \text{ and } V]$$

$$0 \pm [(V) \text{ or } (Q)] \longrightarrow [A \text{ or } A \text{ and } V]$$

$$(A) \pm [1\ (V)\ 1 \text{ or } 1\ (Q)\ 1] \longrightarrow [A \text{ or } A \text{ and } V]$$

$$0 \pm [1\ (V)\ 1 \text{ or } 1\ (Q)\ 1] \longrightarrow [A \text{ or } A \text{ and } V]$$

There are a total of 16 multiplication instructions which may be summarized as follows:

$$(Q) \times [(V) \text{ or } (A)] \longrightarrow AQ \text{ or } AQ \text{ and } (A) \longrightarrow V,$$

$$(Q) \times [1\ (V)\ 1 \text{ or } 1\ (A)\ 1] \longrightarrow AQ \text{ or } AQ \text{ and } (A) \longrightarrow V,$$

where A contains the more significant half of the product without round off, and Q contains the lesser significant half of the product.

$$(Q) \times [(V) \text{ or } (A)] \longrightarrow A \text{ or } A \text{ and } V,$$

$$(Q) \times [1\ (V)\ 1 \text{ or } 1\ (A)\ 1] \longrightarrow A \text{ or } A \text{ and } V,$$

where A contains the more significant half of the product <u>with round off</u>. The multiplier is left in the Q-Register.

There are a total of 16 division instructions which may be summarized as follows:

$$[(AQ) \text{ or } (A)] \div [(Q) \text{ or } (V)] \longrightarrow Q \text{ or } Q \text{ and } V$$

$$[(AQ) \text{ or } (A)] \div [1(Q)1 \text{ or } 1(V)1] \longrightarrow Q \text{ or } Q \text{ and } V,$$

where the remainder is stored in the A-Register.

The overflow flip-flop is normally set to zero before an arithmetic operation. In the event the arithmetic operation should obtain a result which overflows the capacity of the register, the overflow flip-flop is set to unity. As will be seen later, the overflow flip-flop may be used for control purposes.

3.5.2 Transfer of Data Instructions

The transfer of data instructions include:

(V) $\longrightarrow$ A or Q or D

0 $\longrightarrow$ V

(A) $\longrightarrow$ Q or D or V

0 $\longrightarrow$ A

(Q) $\longrightarrow$ D or V or A

0 $\longrightarrow$ Q

(D) $\longrightarrow$ V or A or Q
all 1's $\longrightarrow$ D

3.5.3 Shift Instructions

There are a total of 14 shift instuctions which are summarized as follows:

1. A and Q ordinary shift left
   out $\leftarrow a_0 \leftarrow a_{-1} \ldots \leftarrow a_{-47} \leftarrow q_0 \leftarrow q_{-1} \ldots \leftarrow q_{-47} \leftarrow 0$

2. A and Q ordinary shift right
   $0 \rightarrow a_0 \rightarrow a_{-1} \rightarrow \ldots \rightarrow a_{-47} \rightarrow q_0 \rightarrow q_{-1} \rightarrow \ldots \rightarrow q_{-47} \rightarrow$ out

3. A and Q sign shift left
   out $\leftarrow a_0 \leftarrow a_{-1} \leftarrow \ldots \leftarrow a_{-47} \leftarrow q_0 \leftarrow q_{-1} \leftarrow \ldots \leftarrow q_{-47} \leftarrow 0$

4. A and Q sign shift right
   $a_0 \rightarrow a_0 \rightarrow a_{-1} \rightarrow \ldots \rightarrow a_{-47} \rightarrow q_0 \rightarrow q_{-1} \rightarrow \ldots \rightarrow q_{-47} \rightarrow$ out

5, 6 A, Q ordinary shift left
   out $\leftarrow a_0 \leftarrow a_{-1} \leftarrow \ldots \leftarrow a_{-47} \leftarrow 0$

7, 8 A, Q ordinary shift right
   $0 \rightarrow a_0 \rightarrow a_{-1} \rightarrow \ldots \rightarrow a_{-47} \rightarrow$ out

9, 10 A, Q sign shift left
   out $\leftarrow a_0 \leftarrow a_{-1} \leftarrow \ldots \leftarrow a_{-47} \leftarrow 0$

11, 12A, Q sign shift right

$a_0 \rightarrow a_0 \rightarrow a_{-1} \rightarrow \cdots \rightarrow a_{-47} \rightarrow$ out

13. D ordinary shift right

$0 \longrightarrow d_0 \longrightarrow d_{-1} \rightarrow \cdots \rightarrow d_{-47} \rightarrow$ out

14. D sign shift right

$0 \rightarrow d_0 \rightarrow d_{-1} \rightarrow \cdots \rightarrow d_{-47} \rightarrow$ out

3.5.4 Transfer of Control (Jump) Instructions

The program is sequenced by the program address counter. However, the programmer can interrupt the sequence by inserting a jump instruction which decides whether to continue the sequence or to select another instruction from the core storage. The decision is based on certain conditions that exist in the computer. The instruction the program jumps to may be either the left or right half of a word read from the core storage. Jump instructions can be classified into three types: unconditional, conditional, and preparatory. The jump instructions are as follows:

1. An unconditional jump instruction transfers (V) $\longrightarrow$ PA, and designates whether the left or right instruction is to be performed next.

2. An uncondtional breakpoint jump (identical to the unconditional jump) is used when the "breakpoint" switch is OFF. When the breakpoint switch is ON, the unconditional breakpoint jump halts the computer. Then, upon depressing the ADVANCE button on the front panel of the console, an unconditional jump is performed.

3. There are 13 conditional jumps; the decision to jump depends on whether:

   (a) Overflow flip-flop = 0 (no overflow)

   (b) Overflow flip-flop = 1 (overflow)

   (c) Number in A-Register is positive

   (d) Number in A-Register is negative

(e) Number in A-Register equals zero

(f) Contents of A-Register equal to contents of D-Register

(g) Contents of A-Register greater than or equal to contents of D-Register.

In the following jump instructions involving the Q-Register, the contents of the Q-Register are shifted in a circular fashion in the direction of sensing, after the sensing is performed, regardless of the jump decision. An example of the circular shift can be illustrated as follows:

$$q_{-47} \longleftarrow q_0 \longleftarrow q_{-1} \cdots \leftarrow q_{-47} \longleftarrow q_0$$

Hence, the Q jumps are decided by:

(h) The number in the Q-Register is even, $q_{-47} = 0$, circular shift right

(i) The number in the Q-Register is odd, $q_{-47} = 1$, circular shift right

(j) The number in the Q-Register is positive, $q_0 = 0$, circular shift left

(k) The number in the Q-Register is negative, $q_0 = 1$, circular shift left.

A conditional jump which involves the D-Register and the decision to jump is based on whether:

The number in the D-Register is positive, do = 0.

The remaining conditional jump involves a special application of index registers. The programmer first places a control number in the D-Register. For this purpose, the D-Register is divided into parts called the left and right half, such that either half may be associated with an index register or a memory address. The instruction then decides to jump depending on whether the contents of the specified index register increased by one is not equal to the left half of the D-Register. This jump instruction differs from the others in that the right half of the D-Register contains the address of the instruction control is transferred to, instead of the address portion of the instruction itself.

Similar to the jump instructions are two skip instructions. The first says to skip the next consecutive instruction and execute the following one provided,

$$(A) \geq (X_i).$$

Otherwise continue with the next consecutive instruction. (A) is the contents of the address portion of the skip instruction, and X_i is control information from some external source. Which X_i is selected is determined from the three bits which normally select index registers. The second says to skip the next instruction unless an input/output fault has occurred. The specific fault being examined may be specified by the address portion of the skip instruction.

3.5.3 Special Instructions

The special instructions include:

A. Unique control functions
B. Index Register control
C. Special arithmetic instructions
D. Logical operations

The unique control functions include:

1. Halt: This instruction stops the controls of the Computer.

2. Breakpoint Halt: This instruction stops the controls provided the breakpoint halt switch is ON; otherwise, it is a dummy instruction.

3. Input/Output: This instruction only indicates that a system instruction which involves input/output devices, exists in the D-Register and is to be performed.

4. Inhibit clearing of overflow flip-flop. Otherwise when an arithmetic instruction is performed, the overflow flip-flop is first cleared to ZERO, and then set to ONE in the event that an overflow occurred in the arithmetic instruction. By using this instruction, a subroutine can be sampled (instead of a single instruction) for overflow.

5. Address Substitution: Transfer the contents of the program address counter to the address portion of the word contained in core storage location V. Storage location V usually contains a jump instruction which implies that the program address counter includes information concerning whether the left or right instruction is involved. Further, because of the manner in which index registers are used, special consideration is given to the number of bits transferred.

The Index Register control instructions include:

6. Add the contents of the address portion of the D-Register to the specified index register. (Left or right portion is specified by the command).

7. Subtract the contents of the address portion of the D-Register from the specified index register.

8. Transfer the contents of the index register to the address portion of the D-Register. Because of the manner in which the index register is employed, special consideration is given to the number of bits transferred. The bit of the D-Register, which in an instruction sense determines whether an index register is involved, makes the decision.

9. Transfer the contents of the address portion of the D-Register to the specified index register. A similar situation exists for this instruction as in the preceding one.

10. If the contents of the specified index register are equal to the contents of the indicated half of the D-Register, set the overflow flip-flop to unity.

The special arithmetic instructions include:

11. Increase the address portion of the word in memory location (V) by one. (Left or right is specified by the command.)

12. (A) $\pm$ (D) $\longrightarrow$ A

The logical operations include:

13. Logical Product to D. Take the bit by bit logical conjunction of (D) and (V), and leave in the D-Register.

14. Logical Product to A. Same as 13 above except that (D)→A after the logical product is formed.

15. Logical Sum. Take the bit by bit (inclusive or) of (D) and (V), and leave in V.

16. Add without carry. Take the bit by bit logical (exclusive or) of (V) and (A) and leave in A and V.

3.5.6 Speeds of Non-Arithmetic Instructions

The non-arithmetic instructions may be said to include:

1. Transfers which require 1/2 microsecond exclusive of instruction access time and magnetic core access time. The magnetic core access time is 5 microseconds for reading and 7 microseconds for writing. The nature of the magnetic core storage demands that a write cycle must follow a read cycle in order to retain information in the storage. Therefore, the access time of the magnetic core storage is 12 microseconds. Since two instructions are obtained in a 48-bit word from the magnetic core storage, the access time for instructions is 6 microseconds.

2. Shift instructions require less than 1/2 microsecond per bit shifted.

3. Jump instructions require less than 1/2 microsecond each exclusive of access times.

4. MAGNETIC CORE STORAGE UNIT

The high-speed storage of the TRANSAC S-2000 is a coincident current magnetic core device. Transistor circuits are used. The magnetic core device is designed in groups of 4096 words where each word contains 48 bits. The system is capable of using up to 8 such units.

4.1 Expanding the Magnetic Core Storage

To expand the magnetic core storage from one unit to several requires the addressing registers of the program portion of the computer to be expanded accordingly. This includes the -

1. Memory Address Register (MA)
2. Program Address Register (PA)
3. Index Registers (RI)

These registers are packaged in a manner as to permit expansion by inserting additional printed circuit chassis, which are identical to those present. No rewiring is necessary.

4.2 Magnetic Core Storage Control

The MA register is connected to the selection network of the core storage. To operate the storage the address is set in MA and either Read or Write control is activated. Information read from or written into the storage passes through either the D-register or PR-register, depending on the situation. To transfer data from the D-register to the storage requires first a Read cycle, which clears the storage location, followed by a Write cycle, which transfers the contents of the D-register to the cores. To transfer data from the cores to the D-register requires a Read cycle to transfer the data, and a Write cycle to restore the data in the cores. The Write cycle does not have to immediately follow the Read cycle, hence the Write cycle is held up during the replace type arithmetic operation.

The time required for the Read cycle is 5 microseconds, and for the Write cycle is 7 microseconds.

5. ON-LINE INPUT/OUTPUT EQUIPMENT

Data transmission between the input/output devices and the magnetic core storage is controlled by a set of system instructions which are distinct from the computer instructions. The system instruction format, which uses a 48-bit word length, is illustrated in Figure 5. A system instruction is transferred from the core storage to the D-Register, and the next consecutive computer instruction is the Special Instruction which says that an input/output device is involved . The contents of the D-Register are then interpreted by the input/output controllers. If the core storage is involved with instruction, the starting address of the core storage is contained in the address portion of the special computer instruction.

The system instructions are listed in Table I. An appropriate selection of those on-line input/output devices may be made to complete a data processing system.

Since control information is obtained from the Magnetic Core Storage and stored in the D-Register, an Input/Output register is employed to transmit data between the Magnetic Core Storage and the Input/Output control units. The manner in which the Magnetic Core Storage device is connected to the system is illustrated by Figure 6.

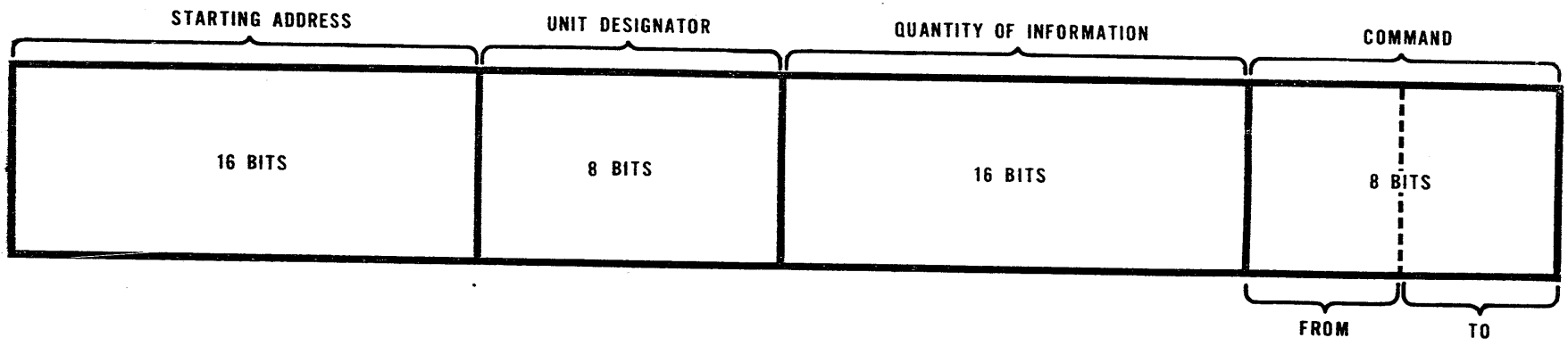
5.1 Magnetic Tape System

The magnetic tape system consists of:

A. Magnetic Tape Units (1 to 256)
B. Magnetic Tape Control Unit (1 to 4 per system)
C. Magnetic Tape Sequencing Unit (one needed when more than one Magnetic Tape Control Unit is used for concurrent operations).

5.1.1 Magnetic Tape Unit

The magnetic tape format is illustrated in Figure 7. Two six-bit groups, each with a parity bit are recorded side by side on 1" mylar tape. The information may be stored on the tape in either 6-bit coded decimal form (alpha numeric data) or in pure binary notation. Storing numerical data in pure binary increases the information density on the tapes; thereby, increasing the effective information transfer rate by a ratio of 3 to 5. The information on the tape is partitioned into fixed blocks where each block has a capacity of 1024 characters. The blocks are separated by the block gap, which is adequate for stopping the tape between blocks without



SYSTEM INSTRUCTION FORMAT

FIG 5

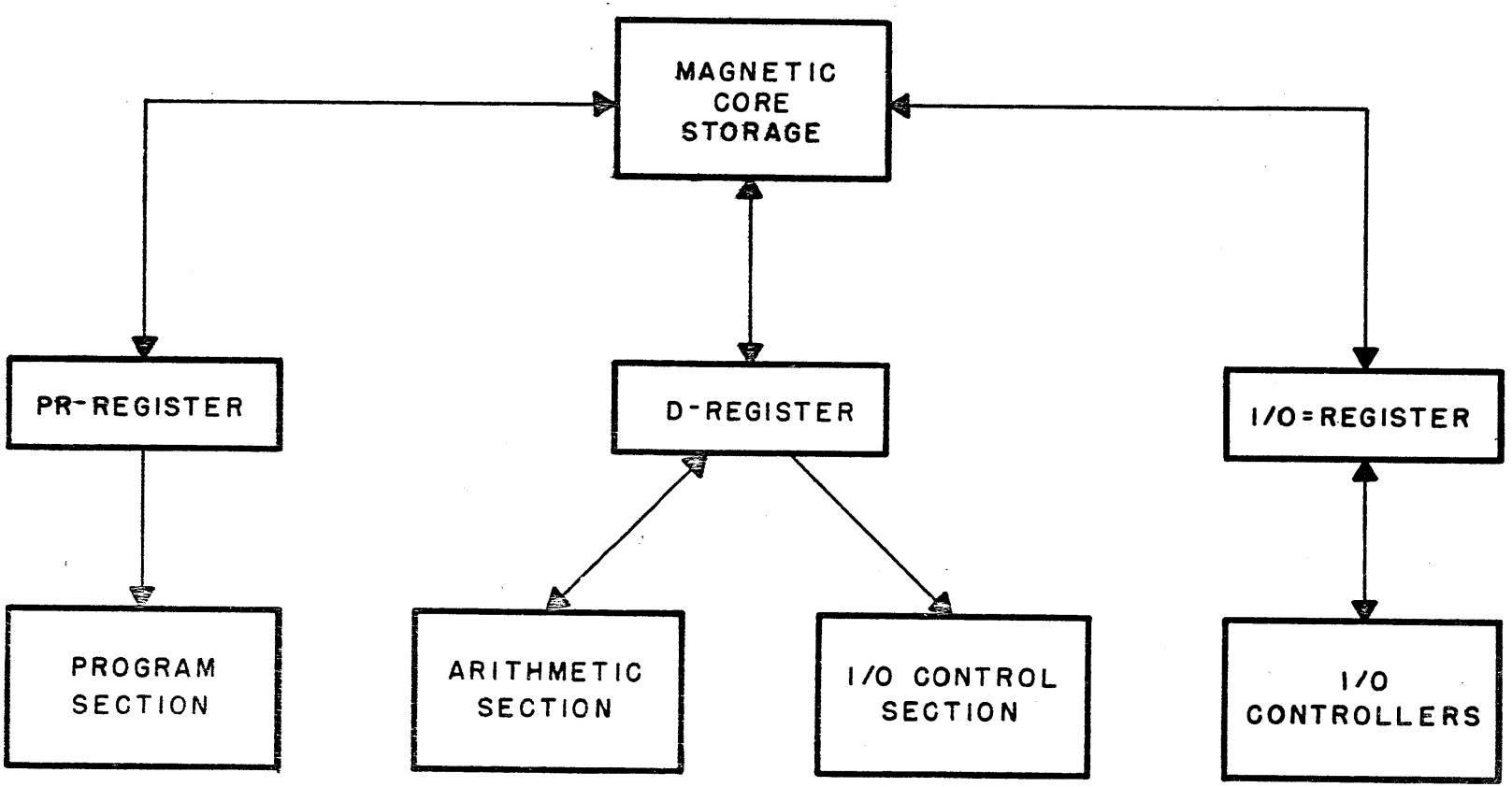
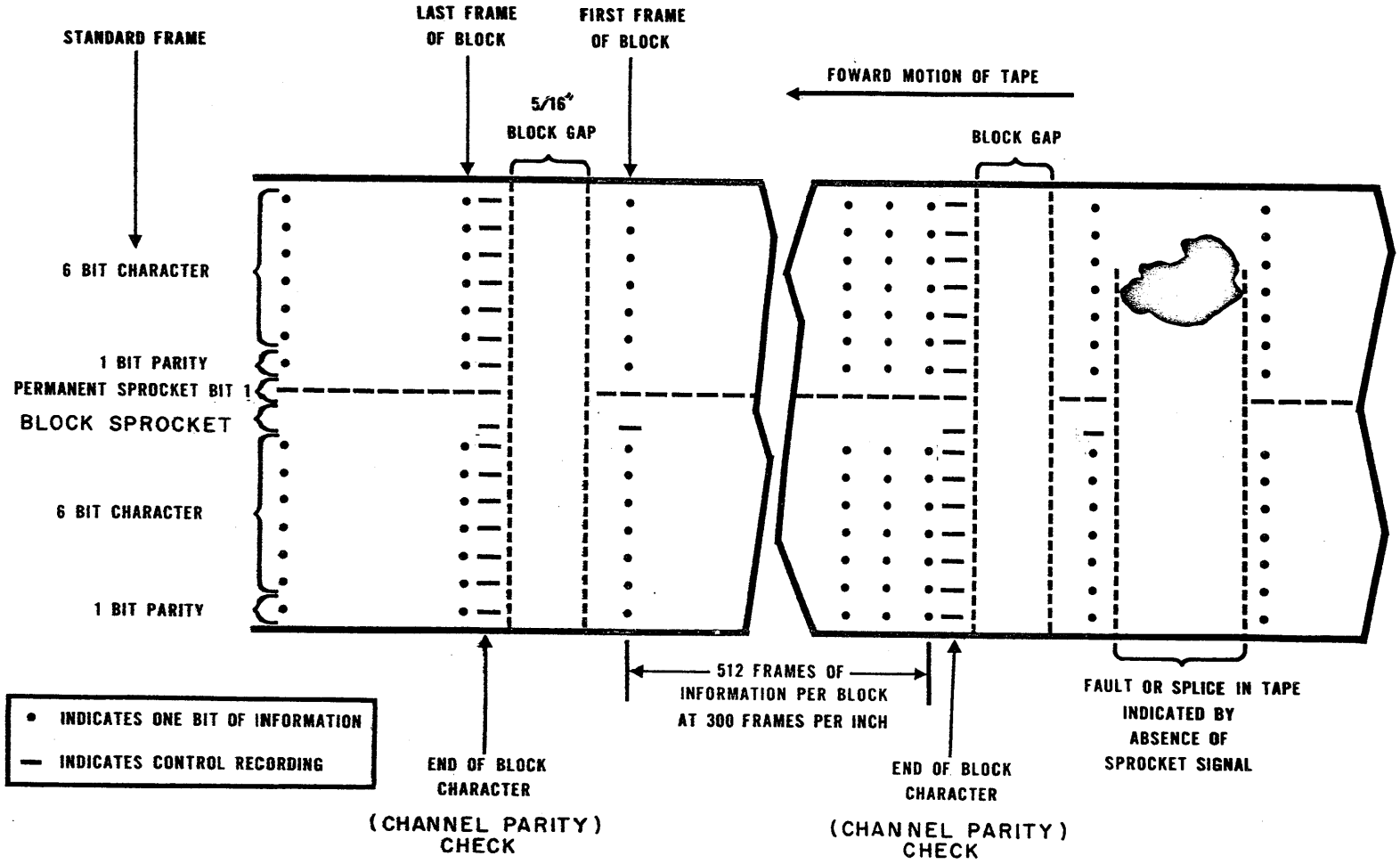


Figure 6. Magnetic Core Storage Connected to the System



MAGNETIC TAPE FORMAT
FIG.7

loss of information. The sprocket channel, which is physically located down the center of the tape, is a permanent recording permitting any faults in the tape to be skipped.

Vacuum controlled servos for the magnetic tape units are used.

The TRANSAC tape unit has the following performance characteristics:

Tape Unit Operating Specifications

1.	Tape Speed	150 inches/sec.
2.	Rewind Speed	250 inches/sec.
3.	Start Time	3 milliseconds
4.	Stop Time	3 milliseconds
5.	Reversing Time	6 milliseconds
6.	Inter-record Gap	5/16 of an inch
7.	Stop-Start Cycles Rate	0 to 120 cps
8.	No. of tracks	16
9.	Tape Width	1 inch
10.	Tape Thickness	1 mil mylar base
11.	Recording Density	300 bits/inch
12.	Type of Recording	NRZ
13.	Reel Size; Type	10 1/2 inch Precision NARTB Reels Machined From Magnesium Alloy
14.	Tape Length	Up to 3600 feet.
15.	Type of Threading	Leader
16.	Reel Change Time	Less than 1/2 minute
17.	End of Tape Sensing	Sensing of Conduct Coating on back of tape
18.	Program Restrictions	None
19.	Write Lock Out	Ring must be inserted in Tape Reel to permit writing on Tape. Tape Reels cannot be stored in container until ring is removed.
20.	Fail Safe Feature	Automatic stopping of tape in case of A. Vacuum loss B. Tape break C. Power failure
21.	Dust Control	Cabinet Pressurized
22.	Reel Door Interlock	Opening door Lock Out Remote Contact.
23.	Local Controls and Displays	A. Start Key B. Ready Light

23.	Local Controls and Displays (cont'd)	C. Select Light
		D. Reset Key
		E. Read Only Light
		F. Load-Rewind Key
		G. Address Selector Key Changes Unit Address
24.	Remote Command	A. Stop
		B. Forward
		C. Reverse
		D. Rewind
		E. Rewind and Lock Out

5.1.2 Magnetic Tape Control Unit

The Magnetic Tape Control Unit is illustrated by the block diagram of Figure 8.

When transmitting data from magnetic tape to magnetic core, the Parity Bit Generator, Checker, checks for odd parity. The Editing Control distributes four double characters to the 48-bit Buffer Storage.

The timing signals which actually control the flow of information originate from the sprocket channel on the tape. During the processing of a block of tape information, a counter is operated which checks whether 1024 characters have been processed.

The Block Sprocket indicates the beginning and end of a block such that abnormal gaps (for tape faults) are not confused as block gaps. The End of Block character contains odd parity check information for the block for 14 channels.

5.1.3 Magnetic Tape Multiple Sequencing

With the tape speed used, words of 48 bits are processed 90 microseconds apart. The magnetic core storage having an access time of 12 microseconds permits several magnetic tape units to be processed concurrently. Also this permits computer operations to be performed concurrently with Magnetic Tape processing. The programming control link which permits this, is a combination of the special skip instruction and jumps.

The Magnetic Tape Control Units are arranged as shown in Figure 9. The programmer selects the appropriate Tape Unit and an available Tape Control Unit is selected by the Multiple Sequencing Control.

TABLE 1 OF SYSTEM INSTRUCTIONS

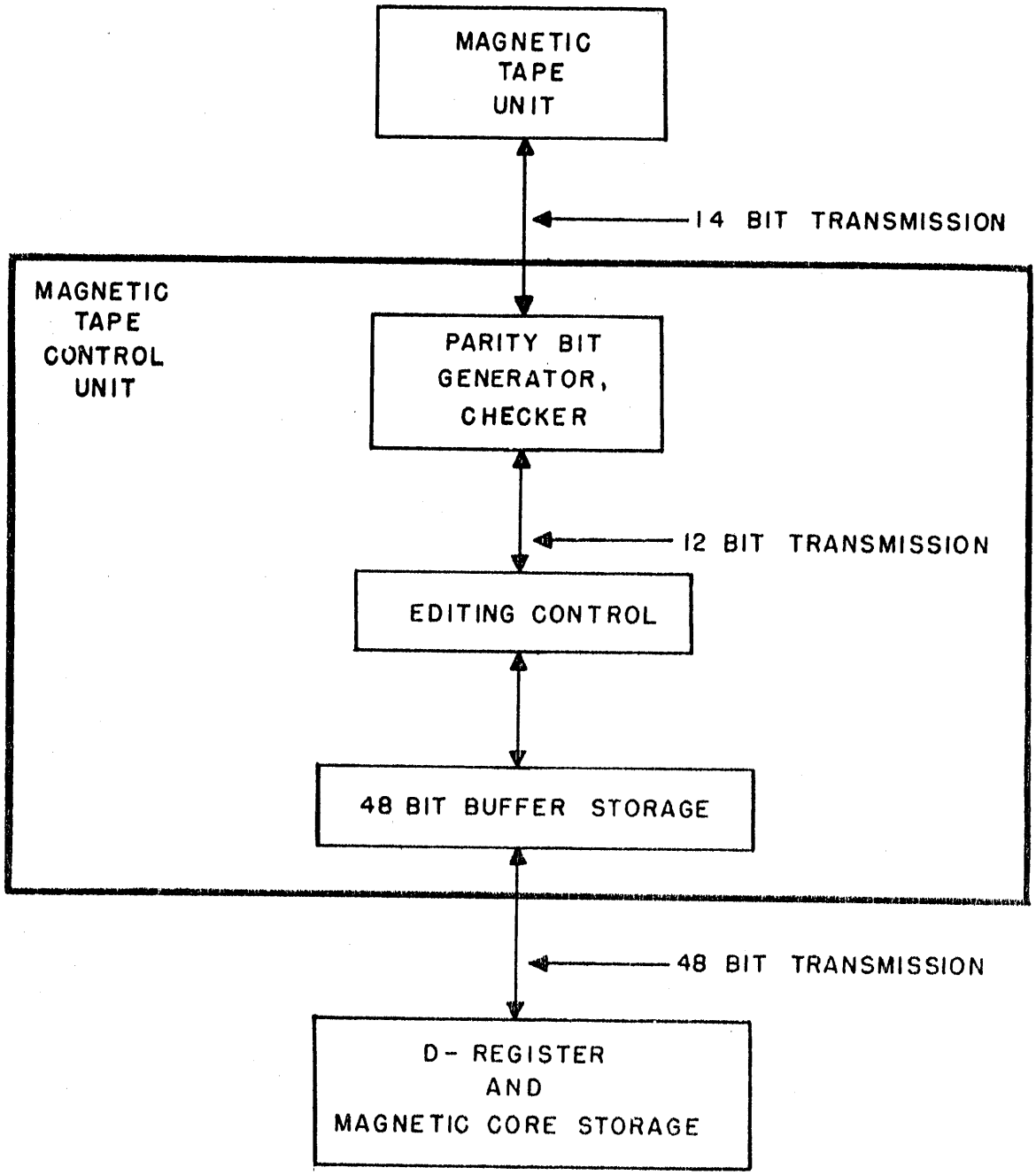
(16 Bits)	(8 Bits)	(16 Bits)	TRANSMIT	
Starting Address	Unit Designator	Amount of Information	(4 Bits) From	(4 Bits) To
On Drum	Which Band on Drum	No. of Words	Core	Drum
On Drum	Which Band on Drum	No. of Words	Drum	Core
On Drum	Which Band on Drum	No. of Cards	Card (Reader)	Drum
On Drum	Which Band on Drum	No. of Cards	Drum	Card (Punch)
On Drum	Which Band on Drum	No. of Words	High Speed Paper Tape (Reader)	Drum
On Drum	Which Band on Drum	No. of Words	Drum	High Speed Paper Tape (Punch)
On Drum	Which Band on Drum	No. of Words	Flexowriter (keyboard or tape reader)	Drum
On Drum	Which Band on Drum	No. of Words	Drum	Flexowriter (printer or tape punch)

TABLE 1 (Cont'd)

(16 Bits)	(8 Bits)	(16 Bits)	TRANSMIT	
Starting Address	Unit Designator	Amount of Information	(4 Bits) From	(4 Bits) To
	Which Tape Transport	No. of Blocks	Magnetic Tape (Forward or Reverse)	Core
	Which Tape Transport	No. of Blocks	Core	Magnetic Tape (Forward)
	Which Tape Transport			Magnetic Tape (Rewind with or without lockout)
	Which Tape Transport	No. of Blocks		Magnetic Tape Space (Forward or Reverse)
Starting Block	Which Bin	No. of Blocks	Core	Magnetic Tape Bin File
Starting Block	Which Bin	No. of Blocks	Magnetic Tape Bin File	Core
		No. of Words	Core	High Speed Paper Tape (Punch)
		No. of Words	High Speed Paper Tape (Reader)	Core

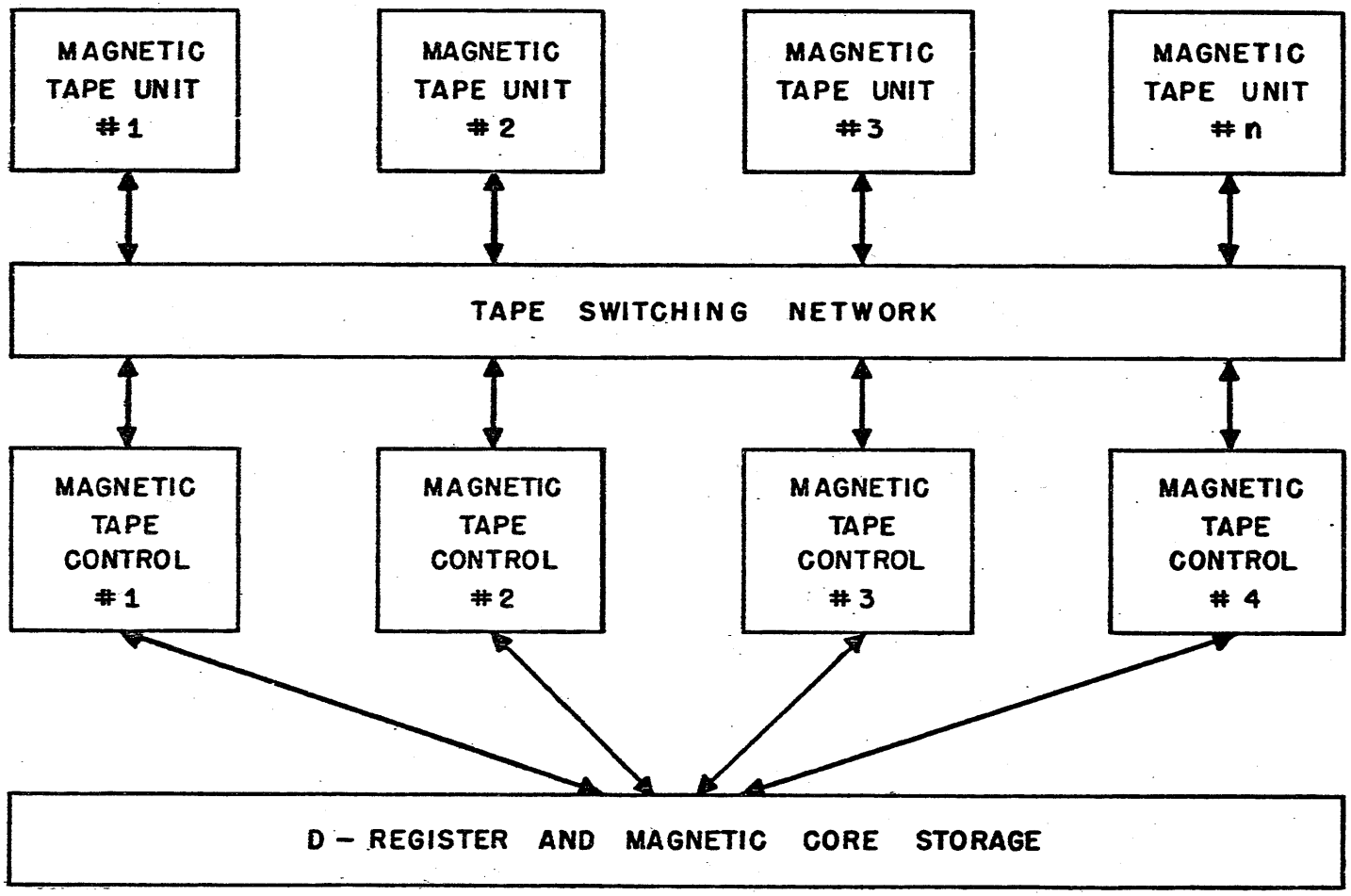
TABLE 1 (Cont'd)

(16 Bits)	(8 Bits)	(16 Bits)	TRANSMIT	
Starting Address	Unit Designator	Amount of Information	(4 Bits) From	(4 Bits) To
		No. of Words	Flexowriter (Keyboard or Tape Reader)	Core
		No. of Words	Core	Flexowriter (Printer or Tape Punch)
		No. of Cards	Card Reader	Core
		No. of Cards	Core	Card Punch



MAGNETIC TAPE CONTROL UNIT

FIG.8



ARRANGEMENT OF MAGNETIC TAPE CONTROL UNITS
FOR MULTIPLE SEQUENCING
FIG.9

When computer operations take place during a Magnetic Tape operation, the demands of Magnetic Tape on the Magnetic Core Storage are given precedence. Sequencing the Magnetic Core is controlled by the Multiple Sequencing Control.

When sequencing is used, a single sequencing unit TS-2150 and up to four (4) magnetic tape control units TC-2150 must be included in the system. If it is desired to have four (4) tapes reading or writing concurrently, then four (4) controllers must be used. If three (3) tapes are all that must be run concurrently, then only three (3) tape controllers are required. The total number of tapes that can be addressed by the sequencer is 16, of which up to any four (4) can be reading or writing at any one instant.

5.2 Magnetic Drum Unit

The Magnetic Drum Unit consists of the Drum Storage Cylinder and the Magnetic Drum Control Circuit.

5.2.1 Magnetic Drum Storage Cylinder

The drum has the following characteristics:

1. Diameter - 20 inches
2. Length - 24 inches
3. No. of Channels - 442
4. Recording - Return to Zero
5. Speed - 1740 r.p.m.

The Drum Storage is subdivided along its axis into eight bands, each of which has a capacity of 4096 48-bit words. This gives a total of 32,768/48-bit words. The bits within each word are arranged in parallel, while the words are arranged serially.

With 4096 words scanned in one revolution, the words occur approximately 8 microseconds apart. Since this is not compatible with the speed of the Magnetic Core Storage, consecutive addresses on the drum are interlaced to produce a speed of 16 microseconds between words.

Since one revolution of the Drum requires 34.5 milliseconds, the average time required to locate the starting address is 17.25 milliseconds. Any number of consecutive words may be transmitted between the Magnetic Core and Magnetic Drum by one System Instruction. Hence, once the starting address on the drum is located, transmission takes place at 16 microseconds between words. This is easily accomplished by the Magnetic Core Storage because of its asynchronous timing properties between the Read and Write cycles.

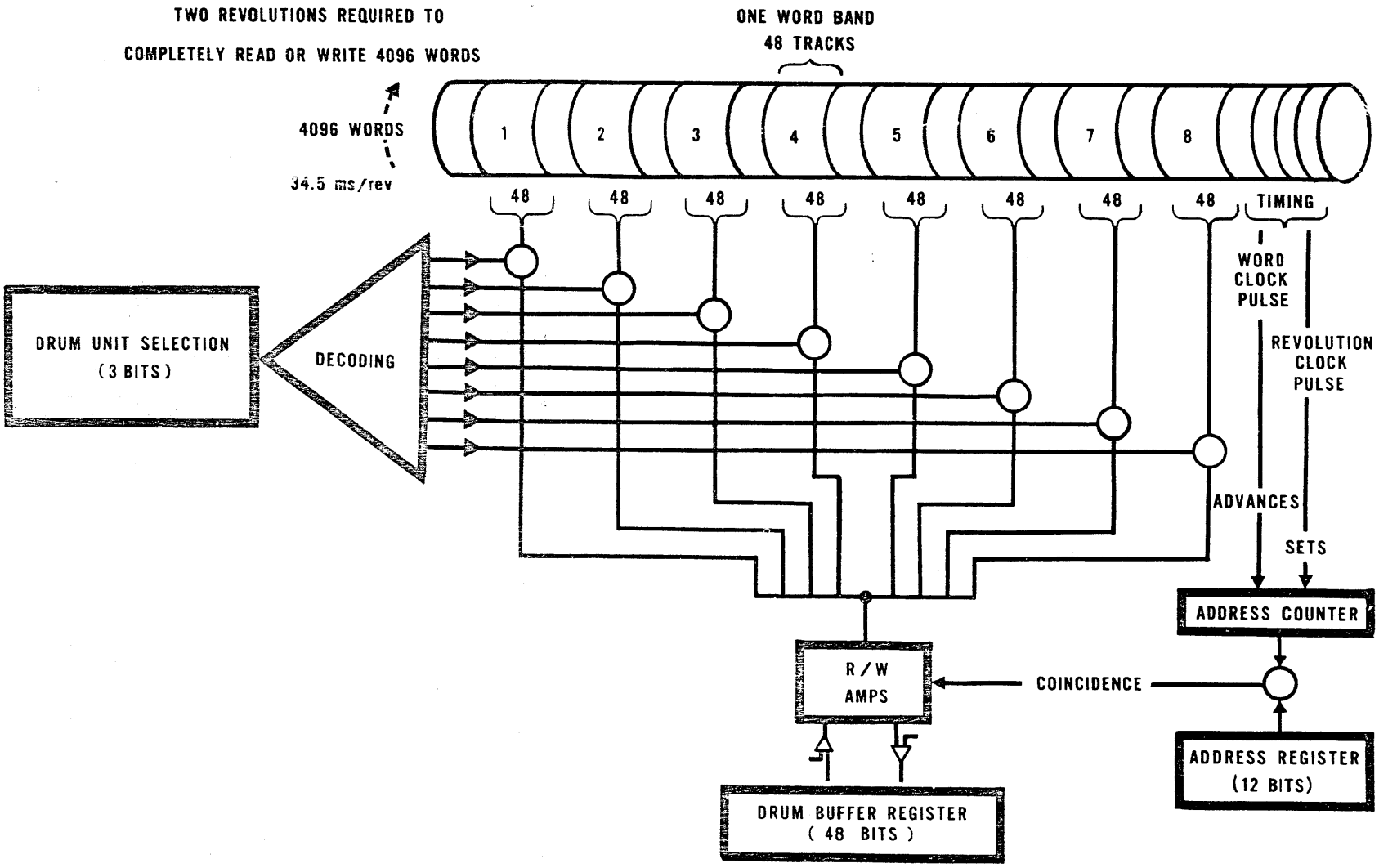
5.2.2 Magnetic Drum Control

The Magnetic Drum and its control are illustrated by the block diagram of Figure 10. For one System Instruction only one band is operated on. Hence, only one set of Read/Write amplifiers is needed. The heads of the drum are connected to the set of Read/Write amplifiers by the switching network as shown in Figure 10. The Read/Write amplifiers are then connected by controlled gates to the 48-bit Drum Buffer.

The words on the Drum are addressed by a binary counter which is clocked by a timing channel of the drum. The counter is checked each revolution by another timing channel.

The starting address is compared to the Drum address counter such that when coincidence occurs, the operation will start.

The Magnetic Drum also communicates with the paper tape and punch card devices, when they are used in the system. Since this does not involve the Magnetic Core Storage, once these System Instructions are set up in the appropriate devices, the computer continues independently. Because of the slower speeds of paper tape and punch card equipment, transmission takes place on a revolution basis with respect to the Drum. Under this condition, the Drum has an asychronous behavior since every revolution of the Drum does not have to be involved in a continuous transmission of several words.



MAGNETIC DRUM UNIT - BLOCK DIAGRAM

FIG. 10

5.3 On-Line Punch Card System

The on-line punch card system can read punch cards into or out of the core memory or onto or off of the Magnetic Drum. When the system is working with the Magnetic Drum the central computer is free to continue with other processing. The on-line card system reads the cards in two modes - binary form in which the card is read or punched in pure binary, first 48 columns only, and the number of cards to be read or punched is controlled by the input/output instruction. The second mode is reading or punching the cards in IBM Card Code. When the mode is selected, columns to be read or punched can be selected by a plug board and the number of cards to be read or punched is controlled by the input/output instruction. The card reader can handle 200 cards per minute and the card punch 100 cards per minute.

5.4 The Paper Tape System

In the On-Line system, the devices which employ a perforated paper tape medium of storage share a control unit as shown in Figure 11. The Paper Tape Control unit is similar to the Magnetic Tape Control unit except that 7-bit instead of 12-bit information is handled at one end. The Editing Control, therefore, converts 8 6-bit characters into a 48-bit word in one direction, and converts a 48-bit word into 8 6-bit characters in the other direction.

The format for paper tape is illustrated in Figure 12, where one frame contains a six-bit character, a parity bit check and a sprocket channel.

5.4.1 High Speed Paper Tape Reader

The High Speed Paper Tape Reader operates at a speed of 200 characters per second. Because of the photoelectric system, the characters are also read while the tape is getting up to speed, and actually can read a character while the tape is stationary. This means that information on the tape does not have to be broken up into blocks. Provisions are made to pass blank tape (leads).

In view of the high speed of the tape, wear of tape is minimized by the use of the photoelectric reading system and a friction drive arrangement for the tape feed.

If the tape is twisted or otherwise prevented from passing through the reader in the correct manner, it will slip in the mechanism without being torn. Tests have shown that a tape can be passed 10,000 times through the reader without appreciable wear, and a spliced tape passes through the machine without difficulty. The tape may be left stationary in the reading position without fear of damage by burning.

5.4.2 High Speed Paper Tape Punch

The High Speed Paper Tape Reader operates at a speed of 3600 characters per minute. This is a synchronous device. The Magnetic Core and Magnetic Drum have an asynchronous behavior at this speed which means that large buffers are not required. When communication is desired between a paper tape device and the Magnetic Drum, the Computer continues to operate independently, unless otherwise decided by the programmer.

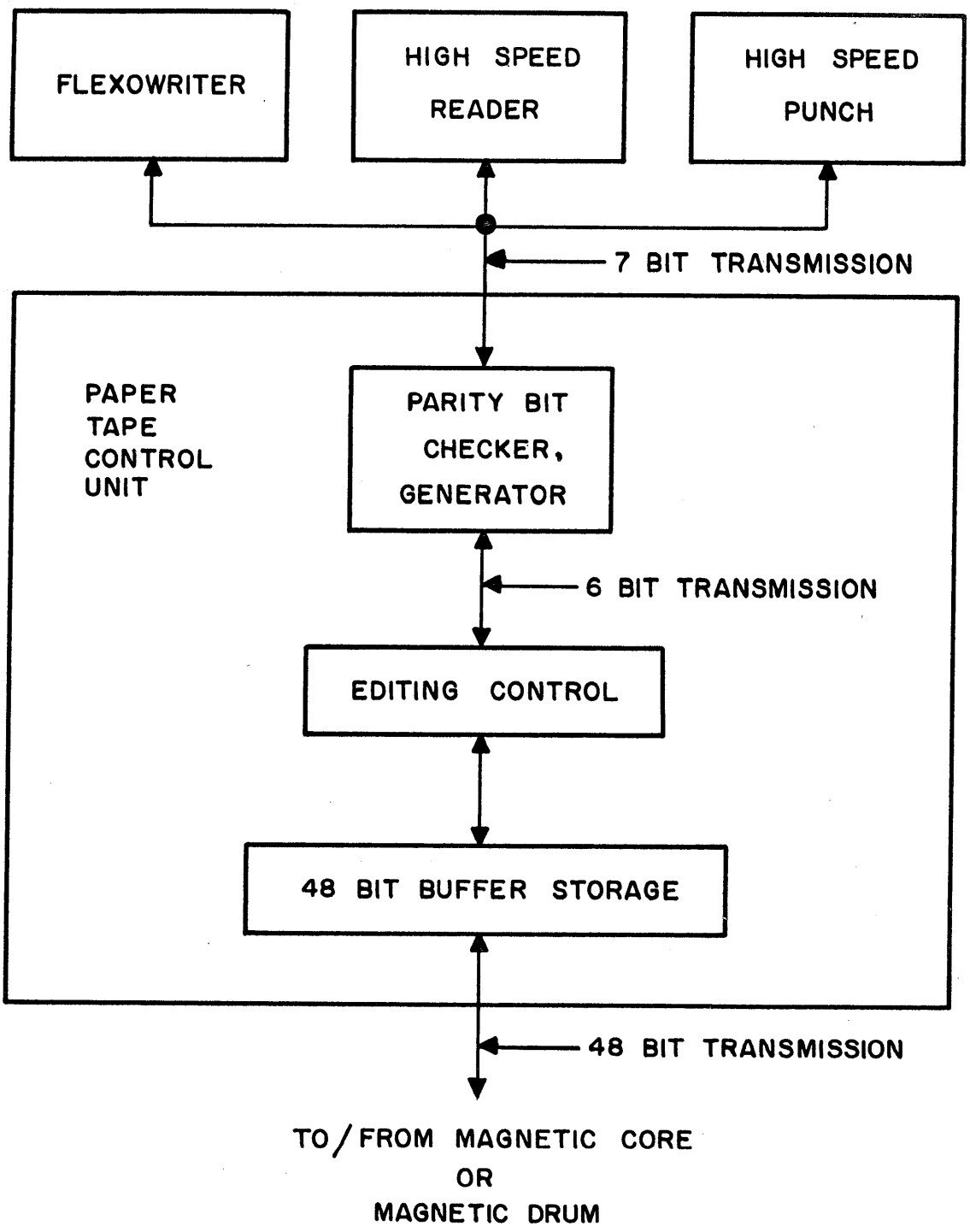


Figure 11 Paper Tape ControlUnit

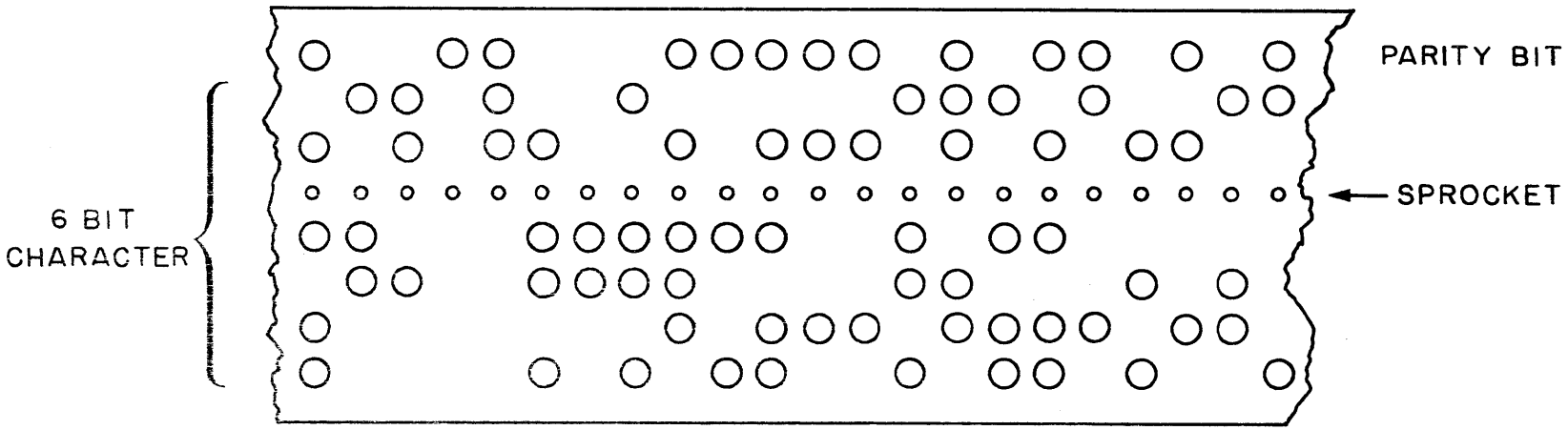


Figure 12. Paper Tape Format

5.4.3 Flexowriter

A standard Flexowriter which consists of a keyboard, printer, Paper Tape Reader and Paper Tape Punch is used. It may be desirable to interrogate the computer during diagnostic operations, by means of the keyboard and at the same time print hard copy or punch a tape to record the activity. On the other hand, during this type of operation information may be desired from the computer which also results in typed hard copy or punched tape.

The Flexowriter printing mechanism can type a maximum of 86 different characters from 43 type bars shiftable to type from either of two characters on each bar. The Tape Reader is set to automatically operate the Flexowriter at a nominal speed of 571 characters per minute.

5.5 Console Typewriter

When an On-Line Paper Tape System is not included in the system an electric typewriter is connected to the console to permit the operator to interrogate the computer during diagnostic operations and to generate a hard copy of such interrogations for the record.

6. OFF-LINE INPUT-OUTPUT EQUIPMENT

The off-line system prepares data from some external source such that it is easily processed by the on-line system. It then takes data from the on-line system and produces it in a form compatible with the application at hand. Since Magnetic Tape is the major storage medium of the on-line system, the off-line system is concerned primarily with writing information on or reading information from Magnetic Tape to some other medium. In general, this is accomplished in a manner as illustrated by Figure 13. With a fixed block length format for magnetic tape, a Block Buffer which has a capacity of 1024 characters is used. The slow device in Figure 13 may be a High Speed Printer, card equipment or some other input/output device depending on the requirements of the particular installation. Hence, the fundamental equipment necessary for most of these applications is the combination of Block Buffer, Magnetic Tape Control Unit, and the Magnetic Tape Unit.

6.1 High Speed Printer

The High Speed Printer system consists of a standard tape unit as described in section 5 above, a Tape Controller, a Block Buffer, Format Editor, Line Buffer and High Speed Printer as described in the block diagram of Figure 14.

A block of magnetic tape information is read into the Block Buffer. Then the Format Editor takes control and distributes the information to the Line Buffer to be printed. The Format Editor consists of a stored program whose instructions have an address and a command. There are 10 bits allocated for the address portion and 2 bits for command; except in the case where the address portion is not used to address the Buffer, in which case 3 bits are allocated to command. The instructions are stored in the Block Buffer along with the Magnetic Tape information. The Format Editor is illustrated by the block diagram of Figure 15. The instructions include:

1. Advance Character Position (column) Counter number of positions indicated 7 bits of address. This command also loads a blank space code in the Line Buffer at each position skipped.

2. Transfer Data (addressed by A) to Line Buffer position indicated by character position counter.

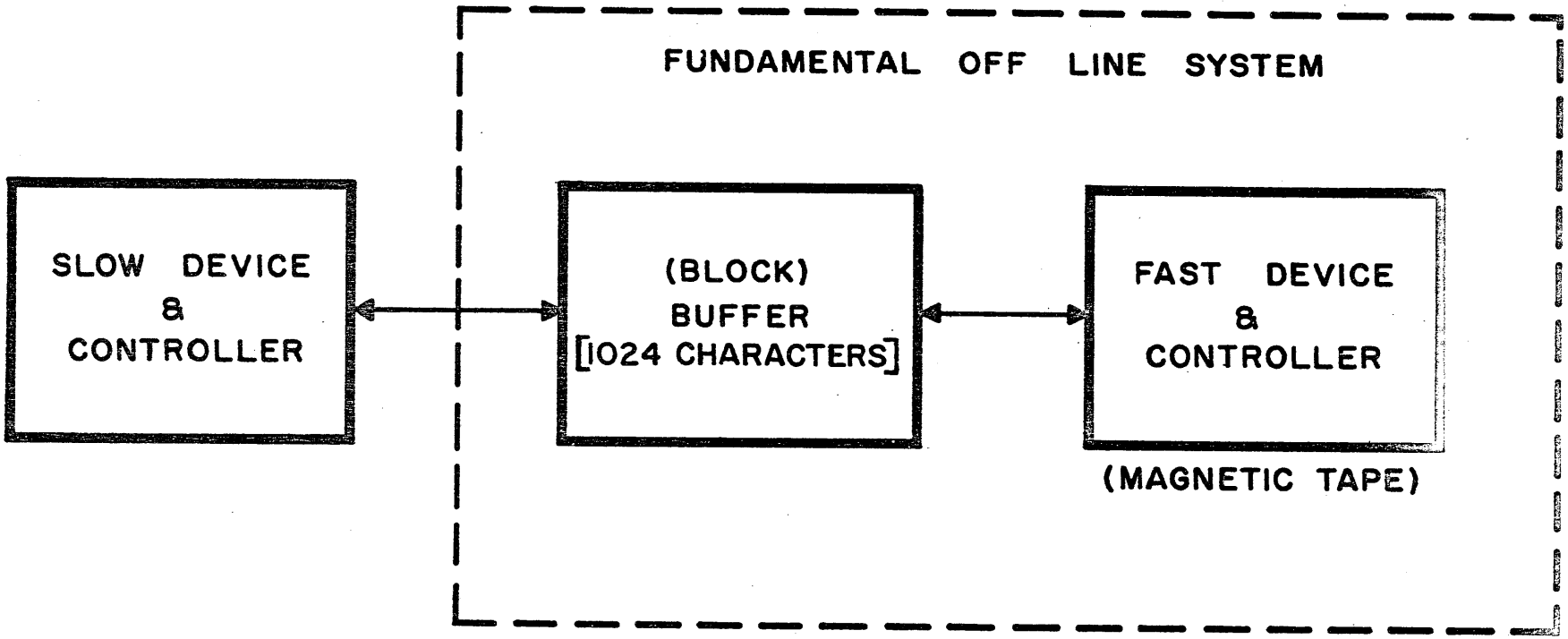


Figure 13. Block Diagram of Generalized Off-Line System

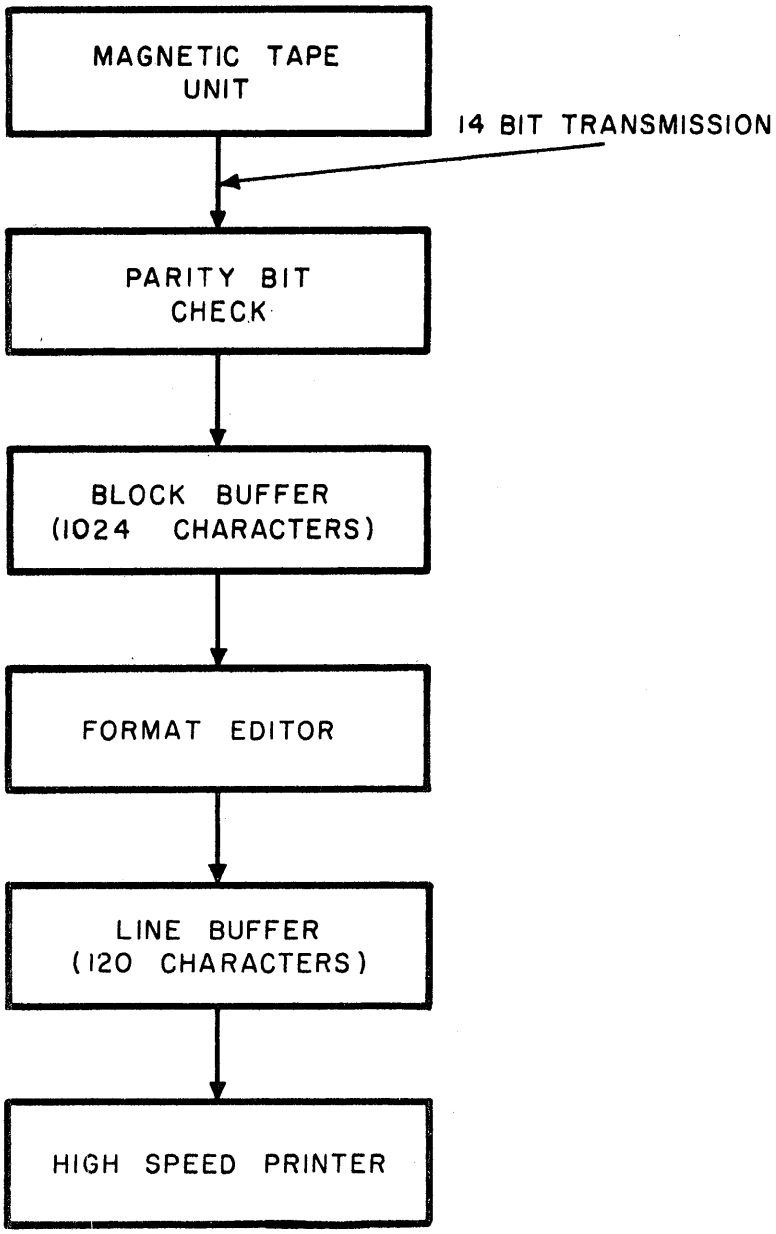


Figure 14 Block Diagram of High-Speed Pointer System

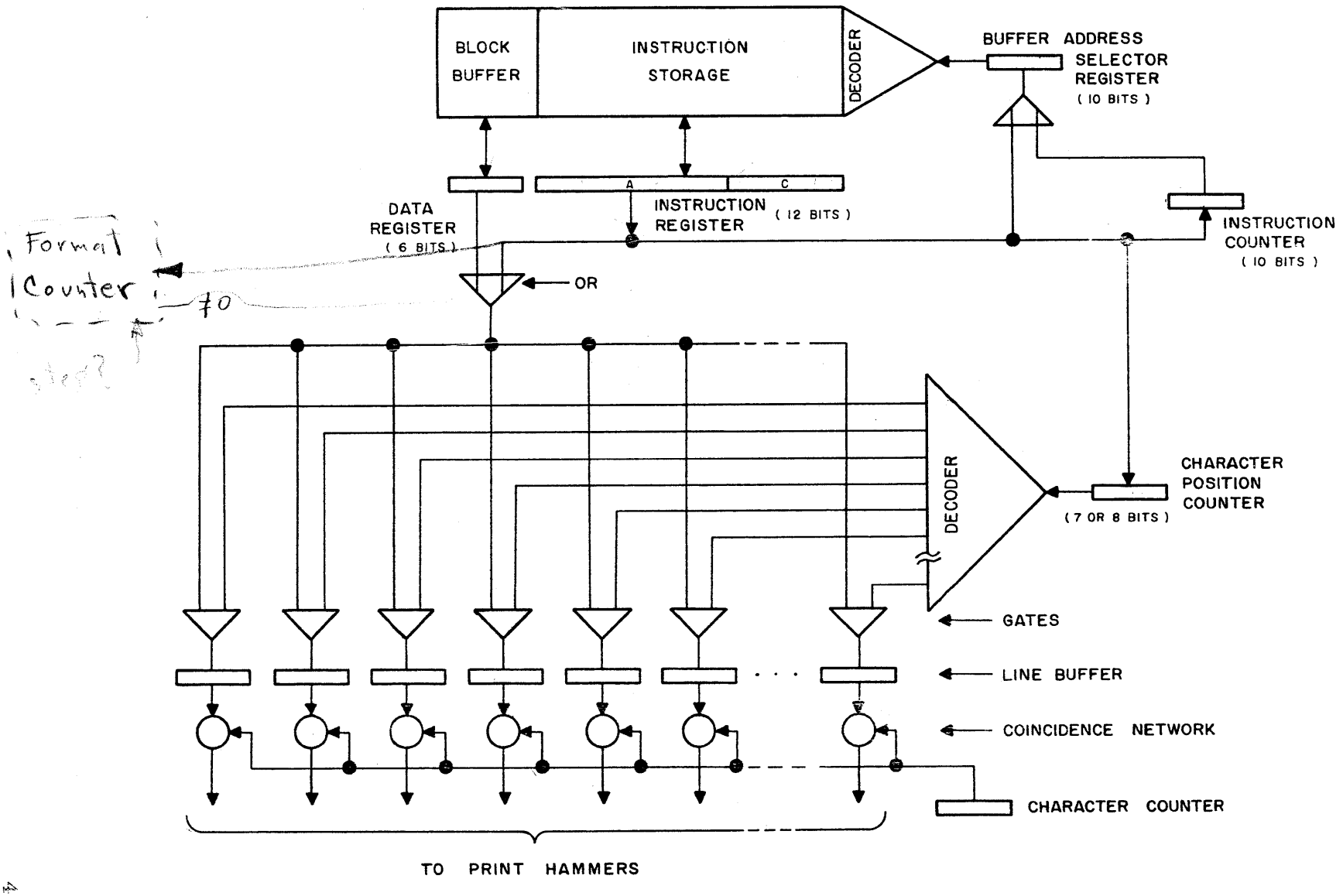


Figure 15 Block Diagram of Format Editor

3. Transfer 6 bits of address to Line Buffer position indicated by character position counter.

4. Set Format Counter to (A). This is a counter which controls the number of times a loop should be performed.

5. Jump to instructions whose address is given by "A" if Format Counter is not equal to zero.

   Instruction 4 and 5 permits placing the same information in several locations on a line or on subsequent lines.

6. Transfer Vertical Format Control to paper loop channel whose address is given by "A".

7. Printing of block is complete. Line counter should equal address "A".

8. Start of new field zero suppression. This command sets the zero suppression circuit so that all zeros are suppressed and in their place a blank space code is loaded into the Line Buffer up until the first significant figure is transferred.

Vertical Format Control is accomplished by a perforated 7 channel tape photo electric sensed. This permited selecting 7 different vertical formats and the loop can be varied in length to handle up to 20 inch forms.

Checks. Parity bit check of information transfers from tape to Block Buffer and Block Buffer, and Block Buffer to Line Buffer.

Firing Circuit Count.

Lines per Block Count.

Line Buffer Load Count.

This type of Format Editor gives the maximum in control. It produces the following features:

(1) Format Control is read into the instruction storage from magnetic tape. It may be contained on the leader of the tape which contains the information to be printed. Hence set up time is minimized. The Format Program can be modified by means of a push-button register on the Printer Control Panel.

(2) Several different types of reports can be made from the same data tape by altering the program. Special symbols and punctuation can be inserted by the Format Editor and, therefore, do not have to be inserted in the data tape by the central computer.

(3) Suppresssion of zeros to the left of significant digits. For example:

$ 25.00
instead of $00025.00,
would be printed.

(4) Arbitrary arrangement of characters on the printed form independent of their occurrence on the magnetic tape.

(5) No preparation necessary in the On-line System for tape to be printed.

The output of the system is printed copy on a weight paper up to and including card stock.

Two models are available, one handles 160 columns and the other 120 columns with 4 horizontal spacing of ten characters per inch. The vertical spacing is six lines per inch. The characters which can be printed are 26 alphabetic, 10 numeric, and 20 symbols. The symbols can be selected to meet specific applications. Listed here is one set of symbols available.

" - Quotation Marks	= - Equal Sign
# - Number Sign	* - Asterisk
$ - Dollar Sign	, - Comma
% - Percent Sign	. - Period
; - Semicolon	? - Question Mark
& - And Sign	@ - At or Per
' - Apostrophe	/ - Diagonal Line or Slash
(- Left Parenthesis	- - Hyphen or Minus Sign
) - Right Parenthesis	¢ - Cent Sign
+ - Plus Sign	: - Colon

Sprocket paper forms from 4 inches to 13-3/4 inches wide can be accommodated. The paper used for single copy is 20 pound weight while special weights for up to five carbon copies can be obtained.

A hand setting control is provided to allow adjustment of the distance between the print roll and hammers to accommodate the number of copies desired.

The paper can be "fast fed" at 25 inches per second when the space between printed lines is sufficient to warrant it. The printing rate is 900 lines per minute.

The "print-on-the-fly" method of printing is used. 160 or 120 hammers are lined up horizontally to be fired at a 2-3/4 inch diameter print roll which is constantly revolving about a horizontal axis at a rate of 900 rpm. The 56 printable characters are spaced around the circumference of the print roll. The impulse hammers, when actuated, strike the paper forcing it against an inked silk ribbon lying across the character face. One major feature of this printer is the "controlled penetration" of the hammer. The hammer travel is controlled between physical stops and never actually strikes the print roll. This feature produces clearer printing and reduces wear on the hammer, print roll, and inked ribbon. Another feature of this printer is the modular construction concept. The hammer-actuator assembly is arranged in groups of four into individual modules, each a complete subassembly. Each module is adjusted so that all four hammers have the same "pulse-to-print" time within ± 5 microseconds.

This factory adjustment eliminates individual hammer timing adjustments in the field. The timing of all hammers as a group is adjustable by shifting the sync-pulse generator with respect to the reading heads. Paper feed tractors shift for horizontal paper adjustment.

The inked silk ribbon used is approximately 15 yards long and is self-reversing in operation. Ribbon life is approximately 100 hours.

6.2 Card to Magnetic Tape - Magnetic Tape to Card Converter System

The Punch Card Converter System employs the same magnetic tape transport and buffer system as the printer system (See Figure 16). The Format Editor works in the same general manner as the printing editor, in that a stored program employing a 10-bit instruction is used. In the case of Punch Card and Magnetic Tape Conversion the program can be loaded from cards.

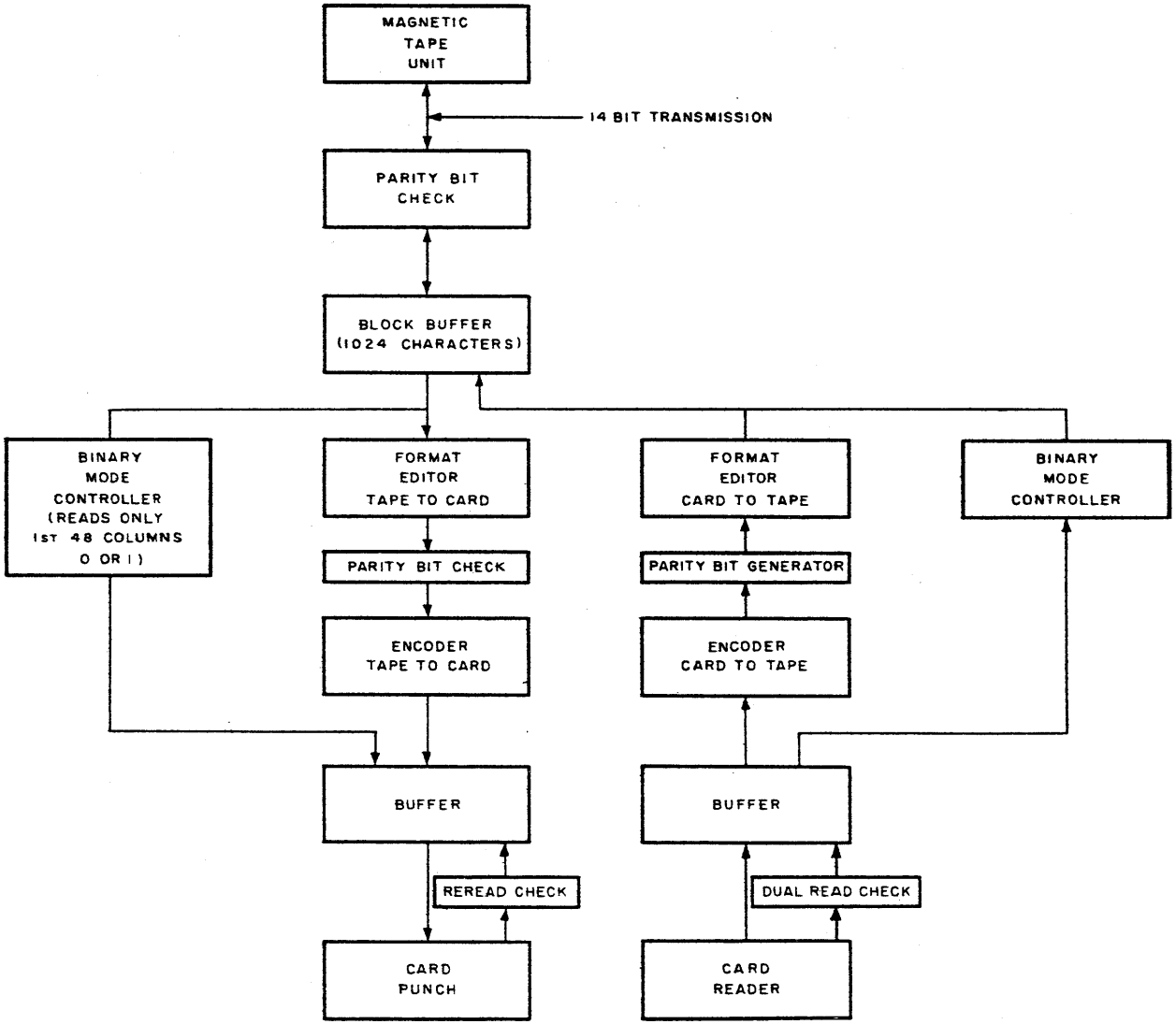


Figure 16 Block Diagram of High Speed Card Conversion System

The instructions for Magnetic Tape to Card Conversion are:

1. Set Character Position equal to address and transfer number punch code.

2. Transfer Data (addressed by "A") to Line Buffer position indicated by Character Position Counter.

3. Transfer 6 bits of address to Line Buffer position indicated by Character Counter.

4. Set Format Counter to "A".

5. Jump to instruction whose address is given by "A", if Format Counter does not equal zero.

6. End of Block Address should equal number of cards punched.

7. Set zero suppression.

The instructions for Punched Card to Magnetic Tape Conversion are:

1. Set Character Position (Column) equal to address.

2. Transfer Data from column indicated by counter to buffer location addressed by "A".

3. Set Card Counter equal to address.

4. Read contents of buffer to tape if card counter equals zero.

5. Set zero field. This command transfers zeros to block buffer if no punch has been read for column being sensed.

The card Read and Punch Conversion equipment can also be set to read or punch the first 48 columns of a card in binary code. When this mode is selected, the converter loads or unloads the Block Buffer from location 000 to 1024 serially. This mode is for converting program cards.

The card reader can read 200 cards per minute and has dual reading heads for checking.

The card punch unit can punch 100 cards per minute and has a read head after the punch head for checking purposes.

Checks. Parity bit check on transfer between buffers and between buffer and tape.

Column Count

Card Count

Sensing brush contact check.

7. PHYSICAL DESCRIPTION

The philosophy in packaging the S-2000 Computer System is expressed as a modular concept. In providing flexible building blocks, various combinations of Computer Units may be grouped to form an S-2000 TRANSAC System tailored to the user's needs.

The modular concept is carried into the construction of the various machine units, all the way down to the few simple basic electronic circuits used to perform the logical functions of the computer.

The frontispiece shows a suggested arrangement for a typical S-2000 System which would permit maximum utilization of installation space and still provide ample room for operation and servicing of the equipment. This arrangement permits flexibility in expansion of peripheral equipment and still keeps the on-line units close to the main computer console. Figure 17 shows the floor space, weight, and power requirements for the various units which are used to make up an S-2000 Computer System.

7.1 Circuit Construction

Figure 18 shows typical module etched circuit cards from which the entire logic of the central computer can be created. These elements numbering 18 different types are 1/16 inch thick x 3/4 inch high x 3-1/16 inches wide. All components are mounted on one side of these modules while the etched circuit is on the opposite side.

Transistor leads are located at the top of the card for access in testing or servicing. Soldered leads which are used to connect the module card to the chassis card are at the bottom. All computer module cards have the same hole pattern.

Figure 19 shows a typical chassis card which has etched circuits on both sides. The connector mounting end is given a special nickel-rhodium plating for low contact resistance and hard wearing surface. The opposite end of the card has mounting provisions for neon indicators. A typical arithmetic register chassis card is shown in Figure 20.

Consistent with Philco's packaging philosophy, to every bi-stable element is associated a visible neon indicator. The neon indicators will be located either on the control panel, if they serve operational functions, or on the end of the chassis cards visible through a transparent door. Individual bi-stable elements on the cards may be changed from one state

Component	Floor Space	Weight	Power		
	ft x ft	lbs.	Standby	Normal	Maximum
Computer	3' x 9.0'	1300	250 watts	115v 1ϕ 250 watts.	250 watts
Core Memory 8192 Words	1.5'x10.0'	1250	250 watts	115v 1ϕ 1650 watts	1700 watts
Paper Tape	3' x 6.5'	800	750	115v 1ϕ 740	750 watts
Magnetic Tape System (a) Control Unit	1.5'x2.5'	700	30	30 watts/tape Unit	30
(b) Each Tape Transport	1.5'x2.0'	500	115v, 60 1ϕ 5 amp.	6.5 amp	16 amp
Magnetic Drum	3.0'x3.5'	1100	220 3ϕ		
High Speed Printer A-Printer B-Printer Driver Unit C-Tape Unit	1.5'x2.0'	600 650 500		1200 watts	
Magnetic Tape File	2.0'x7.5'	1500		220v 3ϕ 30A	
Card Reader & Punch	3.0'x6.0'		400	120v 6A 600 watts	600
Drum, Paper Tape, Card Reader Control	1.5'x2.5'	800	200	300	400

Figure 17 Table of Sizes, Weights and Power Requirements

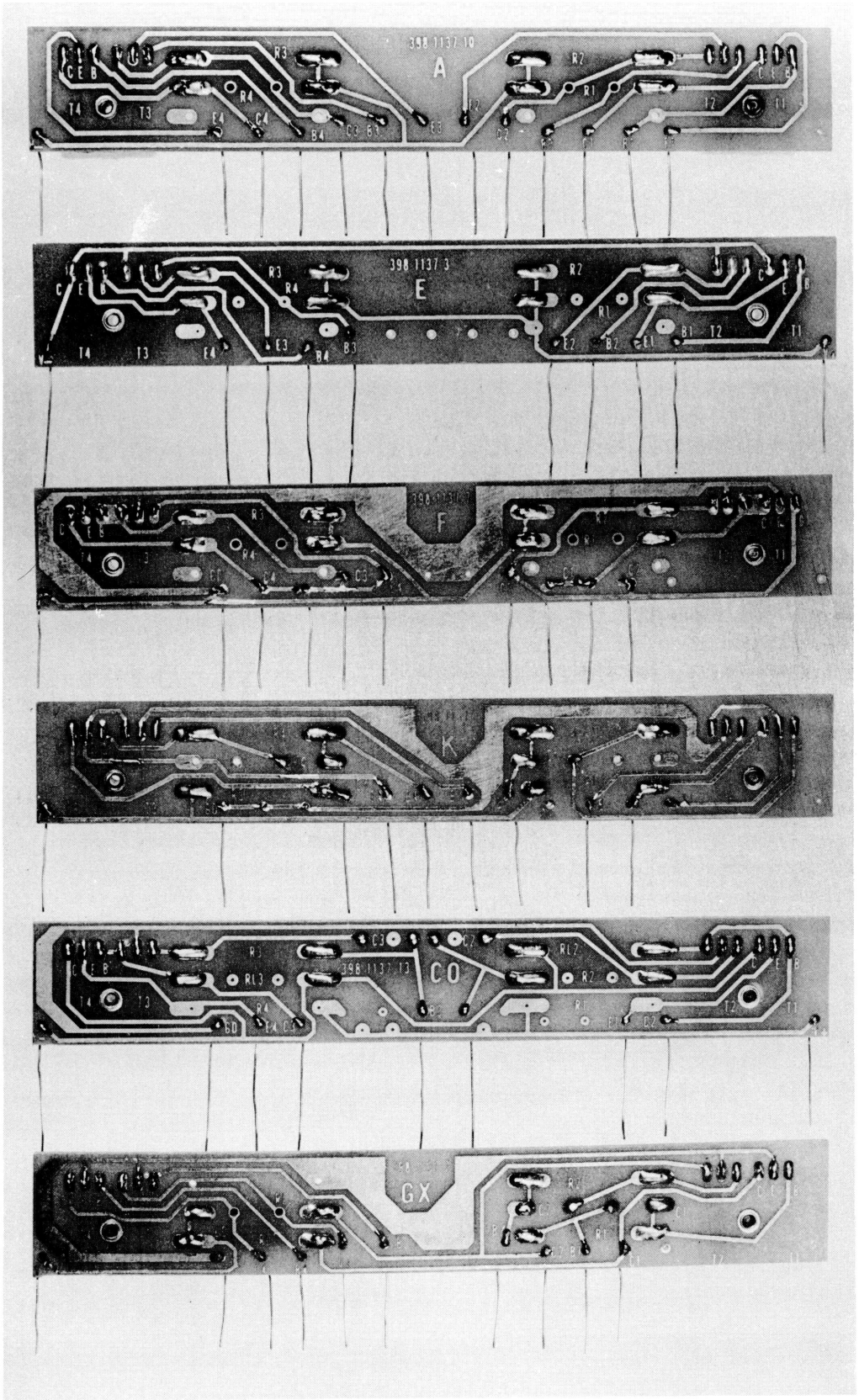


Figure 18 Typical Computer Module Circuit Cards

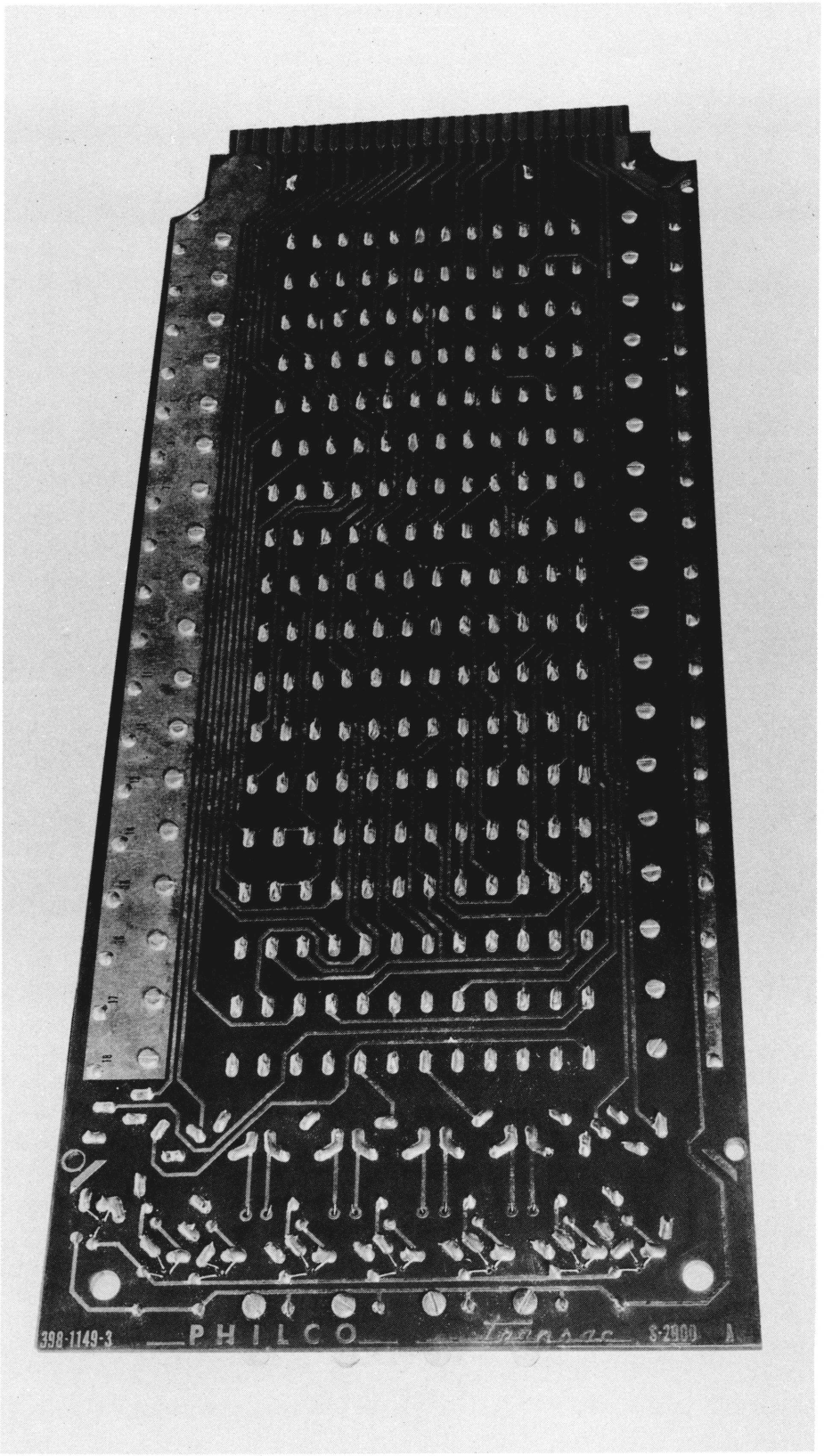


Figure 19 Computer Chassis Card - Rear View

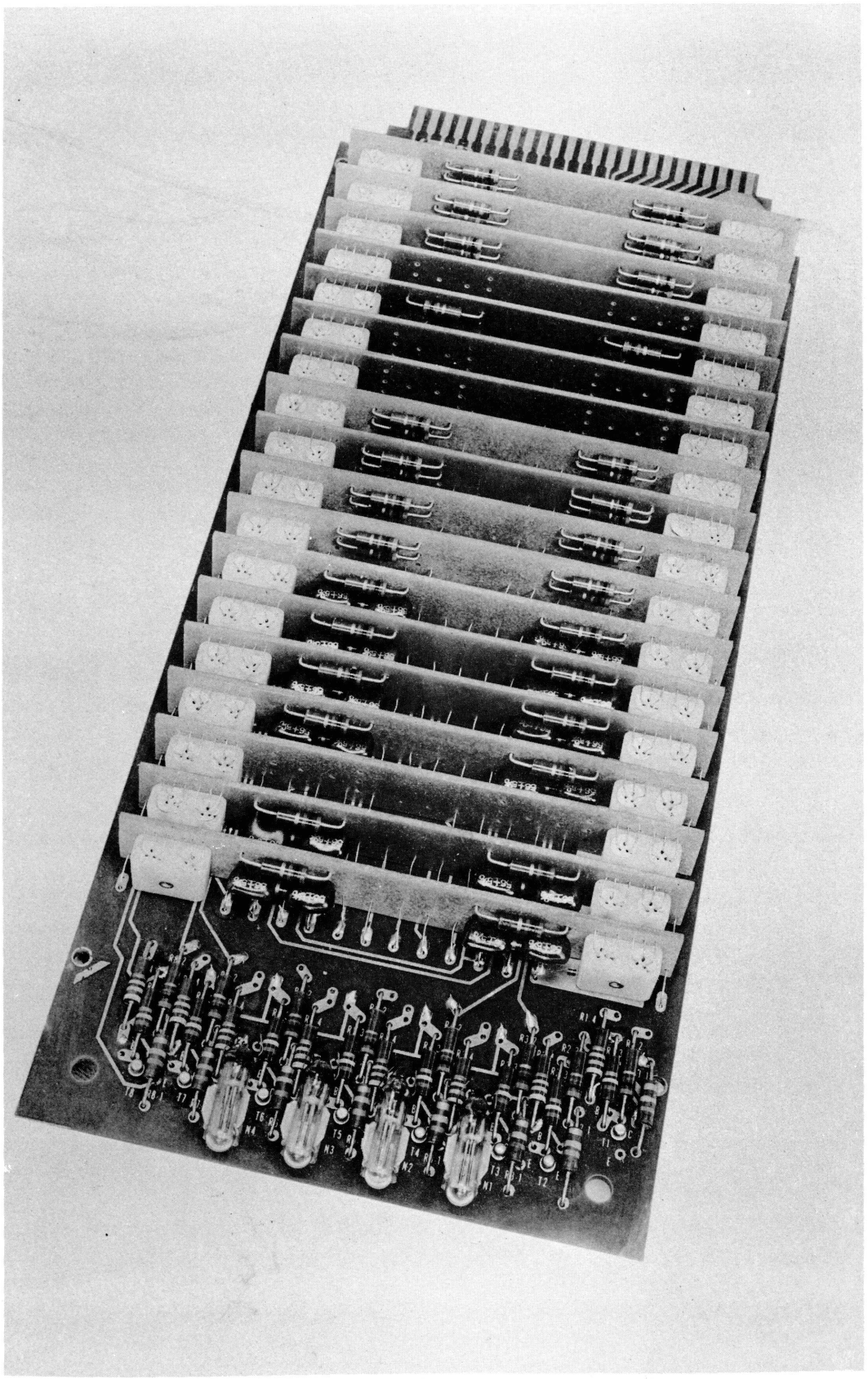


Figure 20 Arithmetic Register Chassis Card

to the other by use of a built-in probe and nickel-rhodium plated test points on each side of the neon-indicators.

The size of each chassis card is identical and each card has the same hole pattern but different etched circuit arrangement to provide a proper interconnecting circuit. A maximum of 18 module cards may be mounted on each chassis card.

7.2 Modular Frame

Figures 21 and 22 show the basic frame which is used to build up many of the various computer system units. This permits flexibility in assembling a system according to customer's requirements and in adding to an existing system. These frames which are 28-3/4 inches wide by 25-3/4 inches high by 15 inches deep can accommodate four rows of chassis cards, 25 per row or a maximum of 100 cards. One photo shows the horizontal level wiring prior to installation of voltage buss bars and vertical wiring, while the other shows the front side of the frame. The etched-circuit cards are installed from this side and are held in place by means of a locking bar.

These modular frames are used to mount the chassis cards, power supplies, and other components in various combinations of each. For instance, the main computer console is built up from four of these frames, and will have 191 chassis cards mounted herein.

Each unit of the computer system is mounted on a separate sub-base providing a toe space at the bottom of each unit. This sub-base is used to distribute the inter-unit power supply cabling and inter-unit signal wiring, plus the air distribution system, which included filters, blowers, and ducting.

Philco's experience in transistorized digital computers has been that, in order to achieve maximum speed and reliability, a minimum amount of cabling is a necessity. Consequently, leads are made as directly as possible and the core memory is placed adjacent to the computer in a manner that, whether it be 4096 words or 32,768 words capacity, the inter-cable distance will be kept relatively constant.

7.3 Computer

The computer unit of the TRANSAC S-2000 consists of four modular frames mounted on one base which is mounted below and behind the main control and display panels. Each such modular frame can accept up to

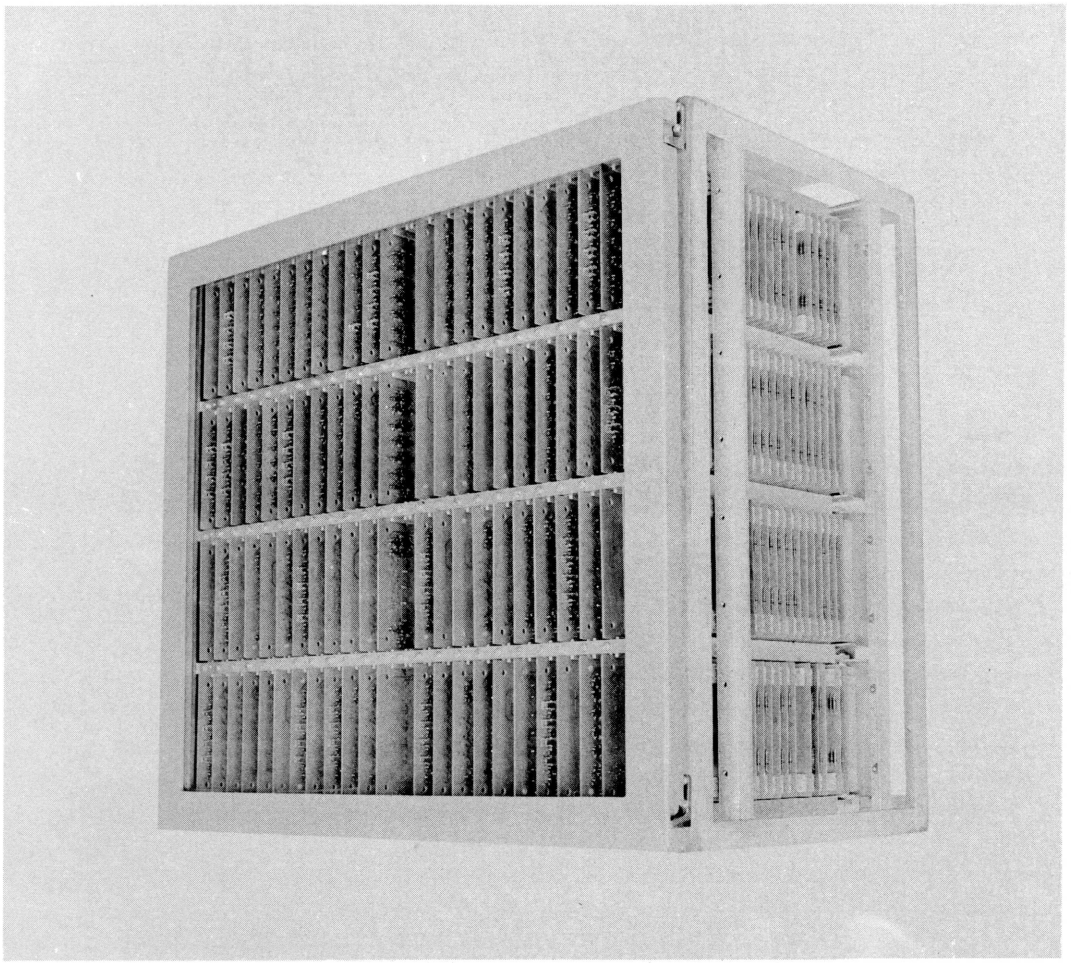


Figure 21 Basic Module Frame - Card End

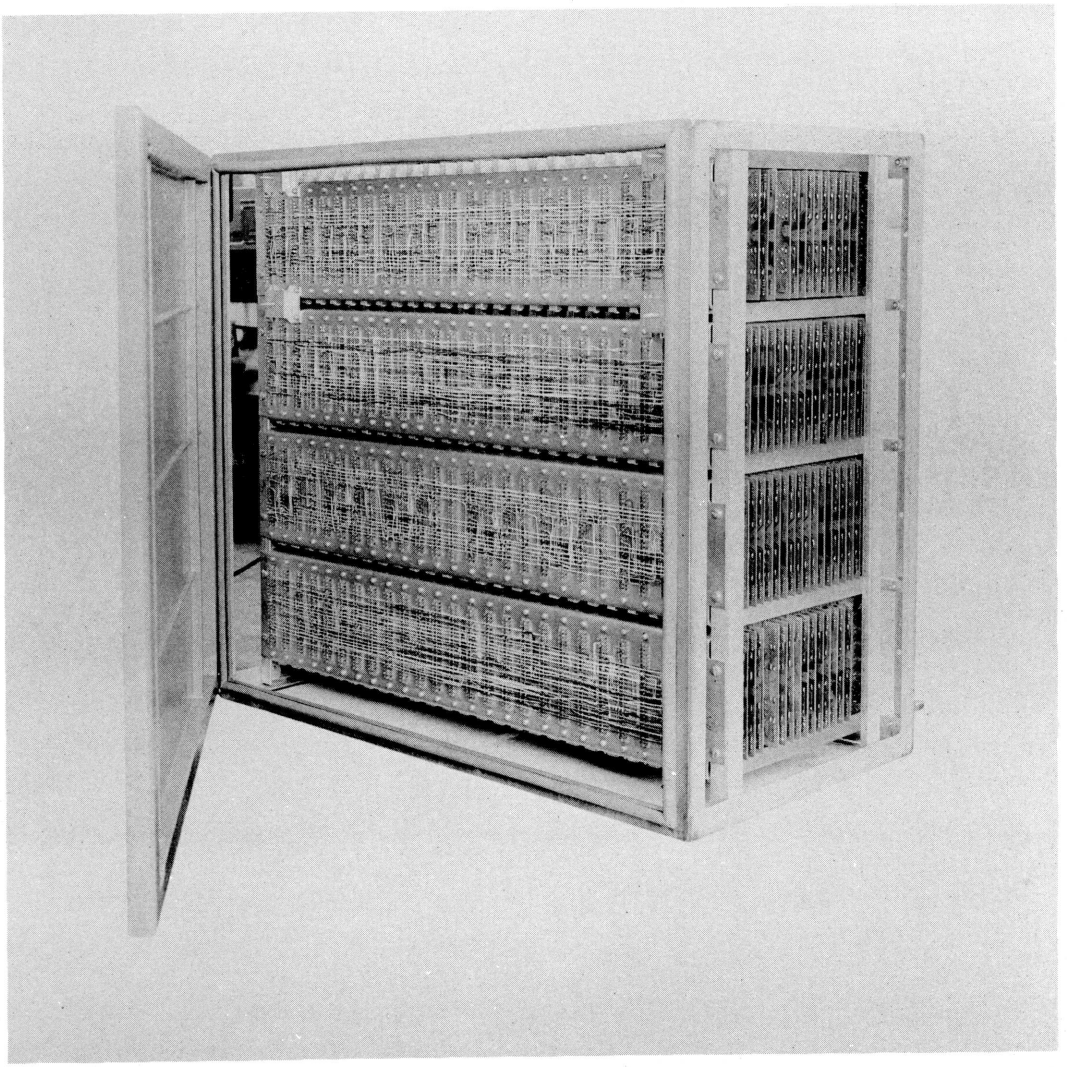


Figure 22 Basic Module Frame - Wiring End

100 plug-in printed circuit cards. Any particular system requirements dictate the actual number of cards packaged in each modular frame. Two of the frames contain all of the cards necessary to the circuitry for arithmetic operations, control operations, index registers, and the floating point system. The third modular frame contains the magnetic core memory control circuits and the decoding and gate circuits for the memory. The four or last right hand section contains the regulated power supplies all the printed circuit cards plug-in from the front of the console and all interwiring is accessible from the rear of the console. Access can also be had to either side by means of hinged doors which are easily removable. Extending on the front of the computer console is a kalistron covered work surface which is hinged at the rear to permit additional access to the plug-in etched circuit cards.

The console control panel, Figure 23, is divided into two main areas, the upper and lower panels. The lower control panel contains the frequently operated controls such as the Data Register and the Program Register. Each of these registers is composed of a keyboard and an register of 48-bit length. Both registers are folded into two sections to centralize controls. The step controls, clear and transfer master push-bottons and the main power controls also appear on the lower panel.

The upper panel contains most of the visual index registers along with some of the less frequently operated manual controls. The right hand end of this panel contains the timing sequence diagram and test controls which are maintenance items and are therefore isolated. The items associated with the Data Register are located on the left half of the upper panel, directly above the keyboard, while the items associated with the Program Register are similarly located on the right half of the upper panel. Associated controls are color coded for operational ease.

The entire lower panel rotates to a near vertical position to allow access and maintenance. The upper control panel is divided into three sections to further facilitate maintenance. Access to these is provided through a door in the rear of the Penthouse.

All wiring is through connectors of the "quick disconnect type. A buss bar system is also provided on the panels for voltage distribution.

7.4 Magnetic Core Storage Unit

Magnetic cores are used for high speed processing of information in or out of the main storage unit or memory. Each core is a .080 O.D. x .050 I.D. ring of ferromagnetic material which is capable of being

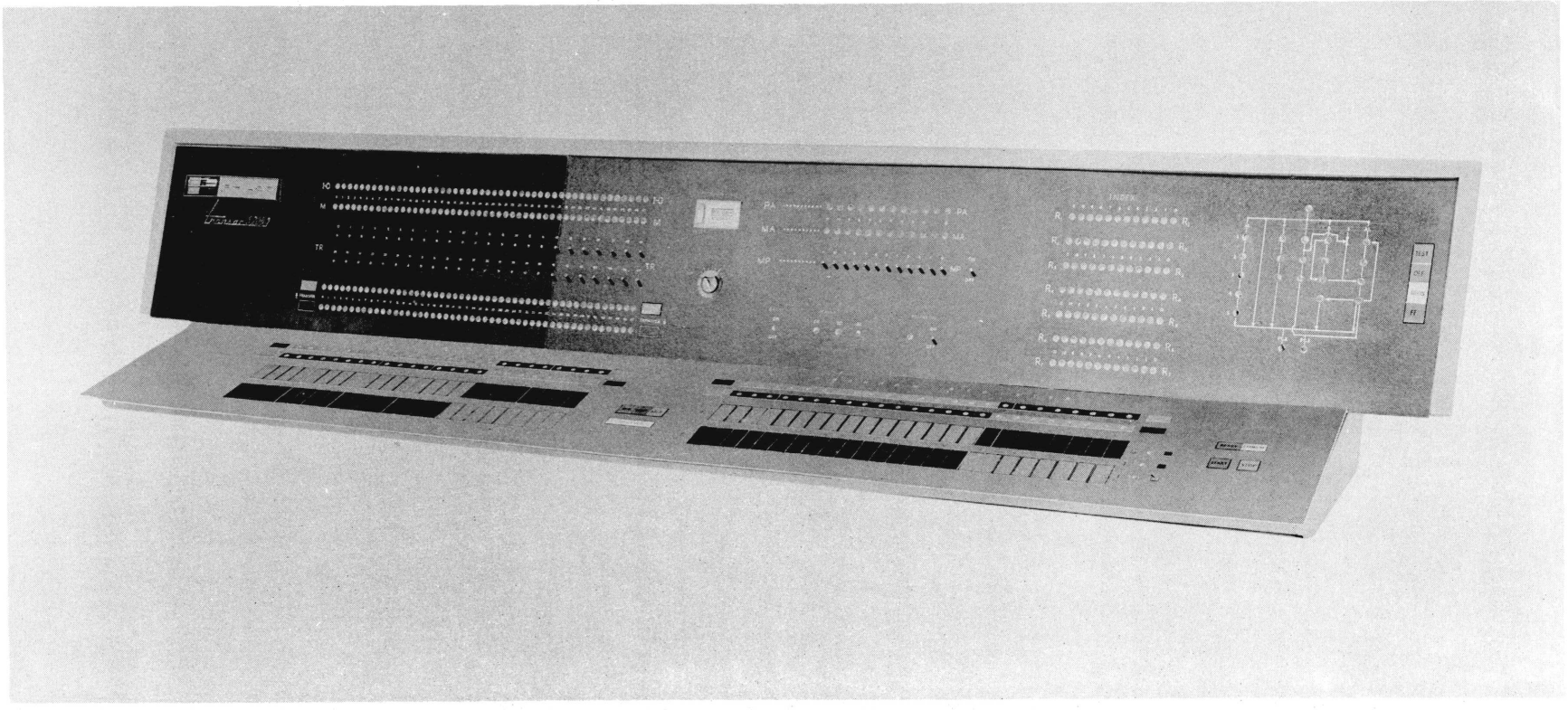


Figure 23 Computer Control Panel

switched to two states of magnetization.

The cores are assembled into a 4096 bit core plane (Matrix) and have an X driver, Y driver, sense and inhibit wire passing through them for selection and interrogation of each core (bit).

Figure 24 shows a frame on which two core mats have been assembled. This dual panel technique has reduced the solder connections between core planes by 50%.

Forty-eight of these planes are mounted into a stack which is controlled to a uniform operating temperature.

The magnetic core enclosure is made up from three modular frames, a base outer skin and doors. This basic unit is assembled to provide either 4096 or 8192 words of memory. It will enclose the magnetic core selection and driver curcuits, power supply and air filtration and circulating system.

Expansion of the core memory system is possible in steps of 4096 words of memory up to 32,768 words. This latter size would be enclosed in four sets of three modular frame cabinets and would be arranged in two level high units having a height of 60 inches.

Figure 25 shows an inhibit drive chassis card and Figure 26 shows a read/write driver output card used in the core memory circuitry.

7.5 Paper Tape Unit

The paper tape unit used with this system is shown in Figure 27. Seven channel punched paper tape is used in this unit. Included in this group is a Flexowriter machine which punches paper tape from a keyboard typing operation or it can print out information read from paper tape fed into the reading head of the machine. A Hi-speed Paper Tape Punch and a Hi-speed Photo-Electric Reader are also mounted in this console. Two slide-away doors are provided to cover these two units when not in use.

The Flexowriter is located to the right of the console at table-top height of 26-1/2 inches. A drawer containing two compartments and located to the left of the knee well accepts processed tape from the tape reader and from the punch. The tape reel for the reader is also mounted in this drawer. The paper supply for the printer is mounted in the lower rear of the console and directly behind the printer. A wide slot in the table top permits the paper to follow the carriage movements. A paper

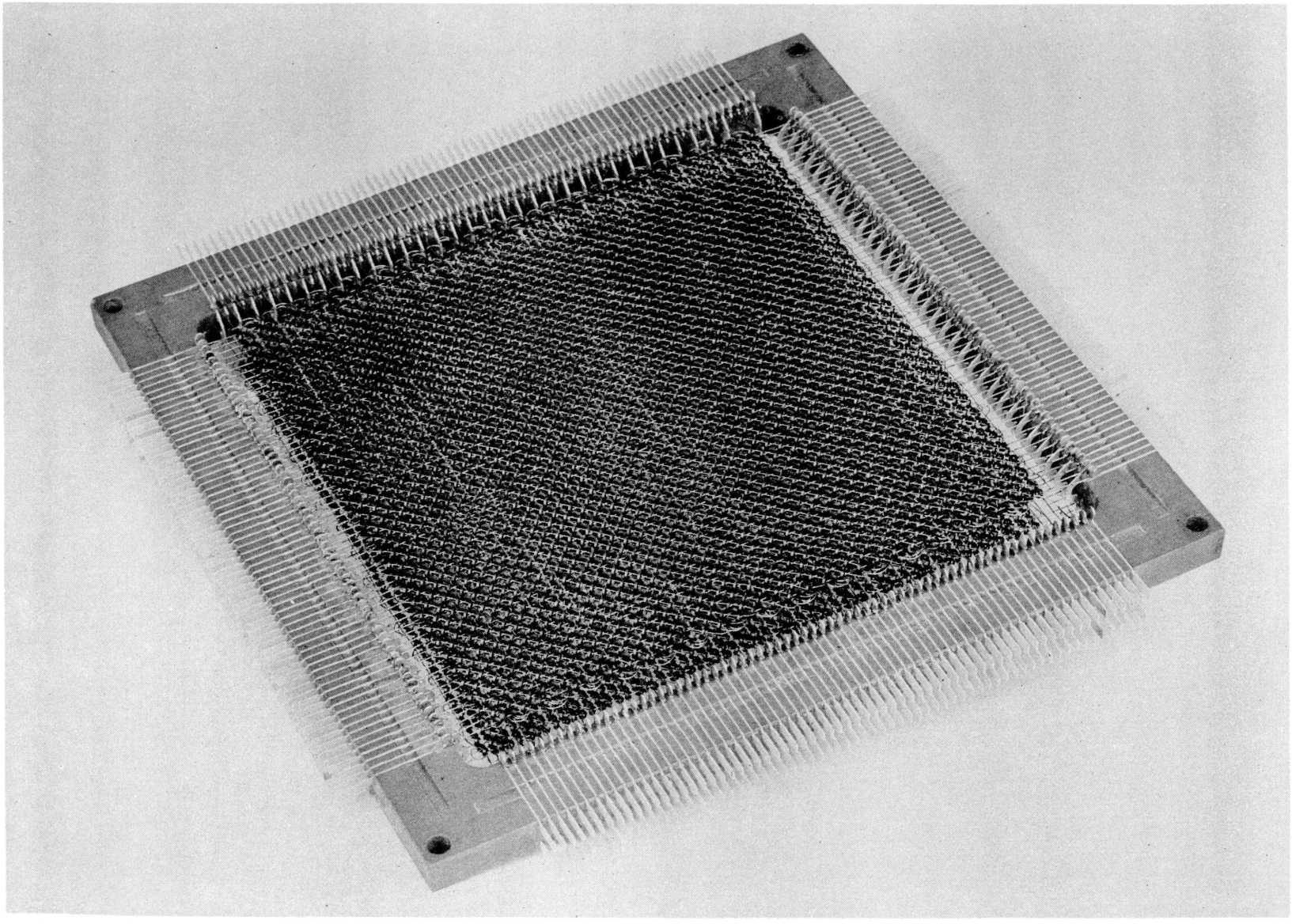


Figure 24 Magnetic Core Plane

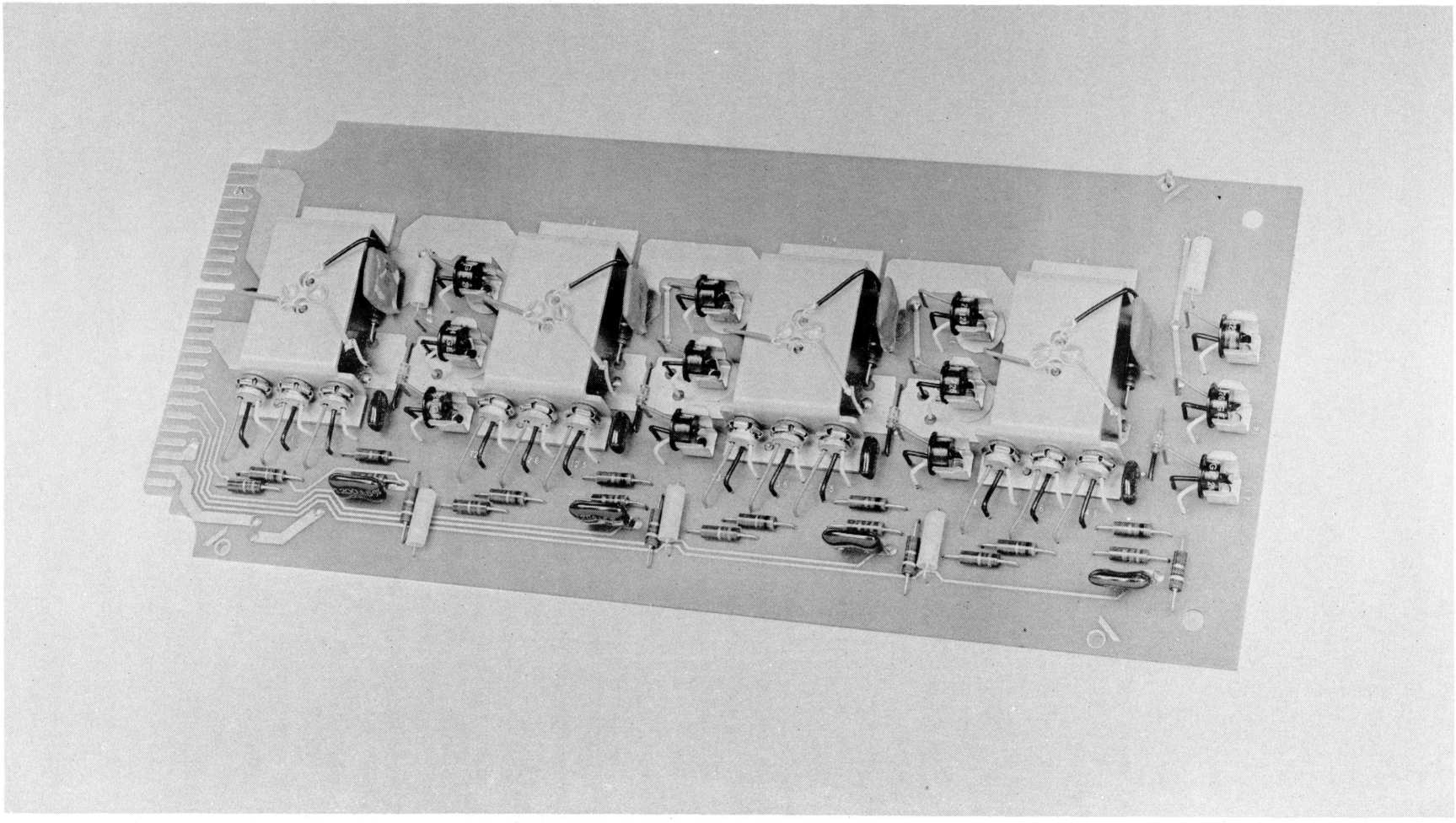


Figure 25 Inhibit Drive Chassis Card

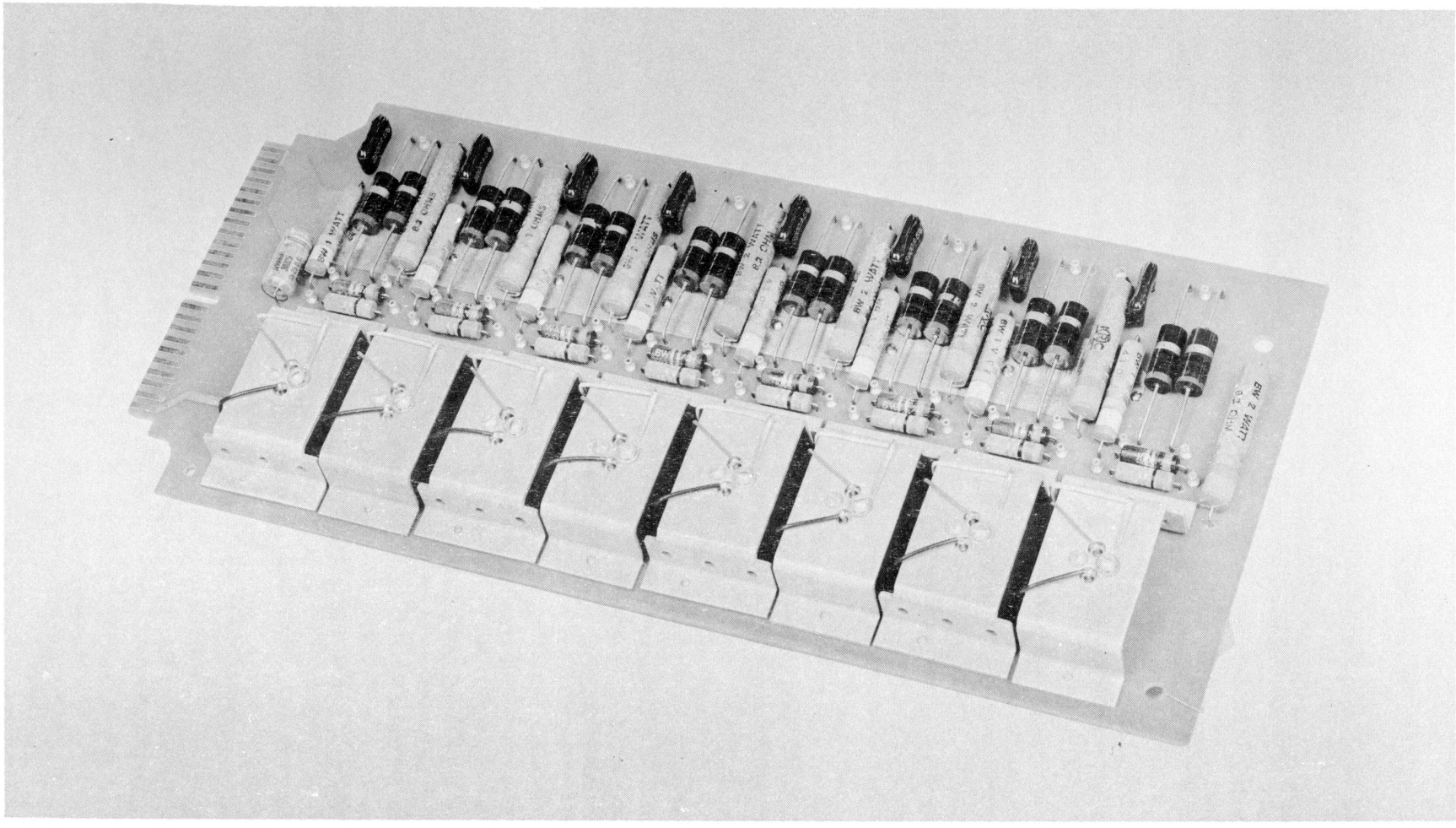


Figure 26 Read/Write Drive Output Card

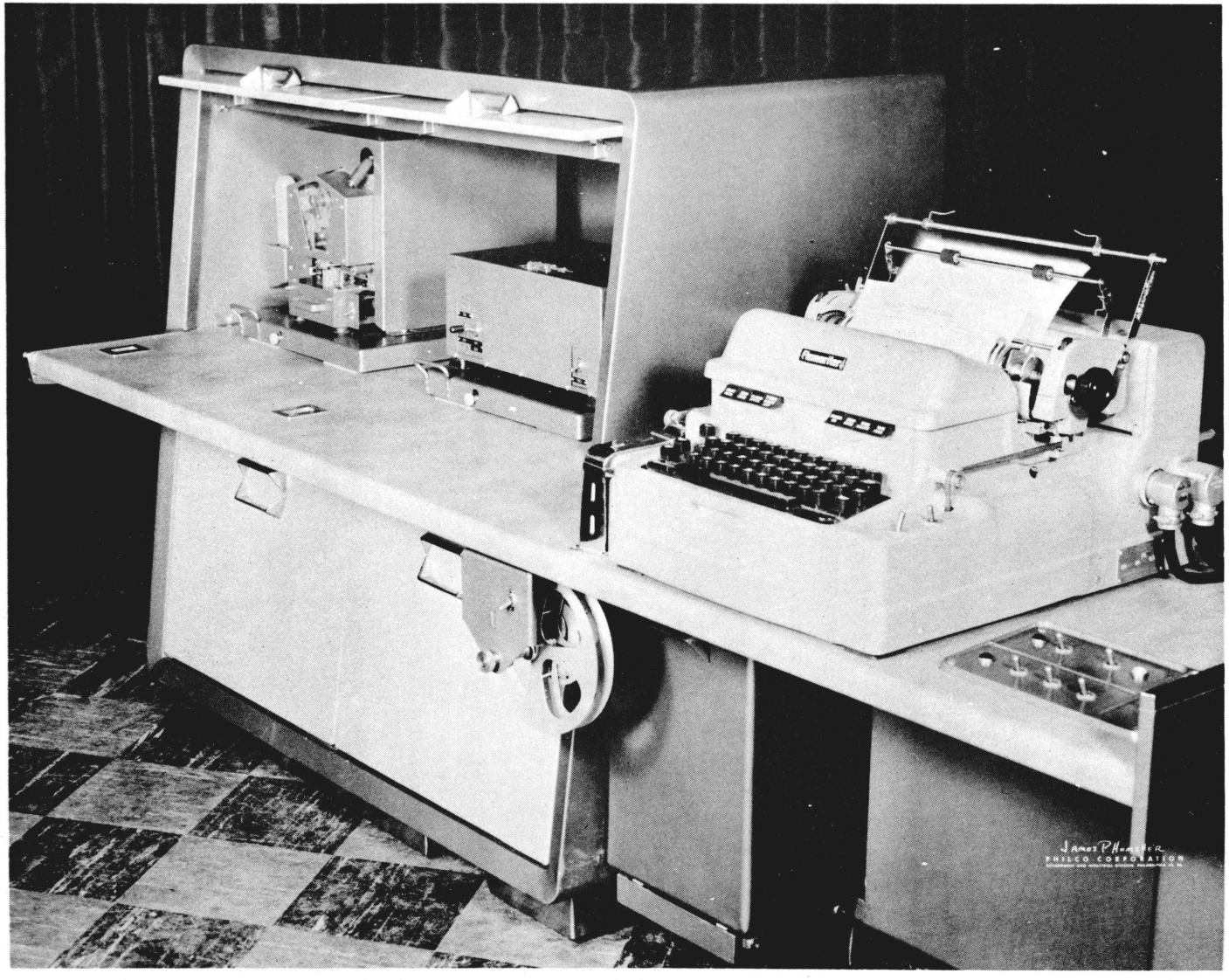


Figure 27 Paper Tape Unit

accumulating shelf attached to the carriage accepts the printed copy as it is delivered.

The high-speed reader and punch are mounted on slides in individual compartments above the table top. The side panels of the console are built with Isofoam in order to reduce the noise made by the high-speed punch. The equipment mounted on slides can lock in three positions thus permitting operation of the equipment either in a closed or semi-open position. The full open position is only used for servicing the equipment.

A rewind reel and drive for the paper tape is also provided at at convenient location.

7.6 Magnetic Tape Units

The magnetic tape units can be arranged as shown in Figure 1 grouped on each side of a Tape Selection and Control Unit, to complete a magnetic tape system.

The control unit is built up from two sections of the standard frame and has glass panel doors on the front side for viewing the neon indicator of the control circuit chassis cards. This unit also contains its own power supply and blower system.

The magnetic tape transports are mounted in individual cabinets which can be butted together to maintain a compact arrangement. Inter-cabling between units is done through the base. A full length door has a partial glass panel for viewing operation of the tape transport. A local control panel is provided at the top of the cabinet and provides visual indication of each particular tape transport's status.

Mounting this panel at this location makes it readily visible even from the computer's operator station. As a safety precaution, a special interlock switch is provided to disconnect the drive circuit of the transport when the front door is opened.

The high performance vacuum sensing type tape transport assembly is rack mounted to the enclosure and can be swung outward on its front panel hinge to permit inspection and servicing of the back panel drives.

The drive electronic chassis is rack mounted below the transport assembly. Under this is mounting provision for the transistorized read/write amplifier chassis. The cooling blowers which are mounted on the electronic drive chassis circulate air entering the cabinet at the bottom

through filters, over the amplifier circuits, up through the tape transport assembly and out of the top of the cabinet.

7.7. Magnetic Drum Storage Unit

The magnetic drum enclosure is a special cabinet which can be lifted from the base to expose the drum assembly. The enclosure contains the drum motor starter, the magnetic head read/write amplifiers and the head selection network.

The drum rotor which is 20 inches in diameter and 24 inches long is precision ball bearing mounted with its axis in a vertical direction, to reduce deflection. The rotor is precision ground after hard magnetic coating. The drum housing is a simple circular shape with radial holes placed on a helix pattern for mounting of individual magnetic recording heads. Both rotor and housing are made from specially treated aluminum alloy forgings for temperature stability.

The drum assembly is attached by three isolation mounts to the sub-base which has casters for moving the unit and lockout provision for final installation.

Drum has storage capacity for approximately 2,000,000 bits.

7.8 High Speed Printer

The high-speed printer system is capable of printing 120 character lines at the rate of 900 lines per minute. Information is read from a Tape Transport Unit in blocks of 1024 characters into the Printer Buffer Memory, where it is edited and available for the Printer's drive and selection circuits.

Printing is done by an on-the-fly type printer which causes a solenoid operated hammer to strike the paper against a revolving print wheel when the proper character is passing a rotation.

This unit is designed for heavy-duty, electronic data processing particularly those applications requiring " high-document-per-hour" printing the business forms. Figure 28 shows the internal chassis construction of the mechanical printer.

A major feature of this printer is the "controlled penetration" of the hammer. The hammer travel is controlled between physical stops and never actually strikes the print roll. This feature produces clearer

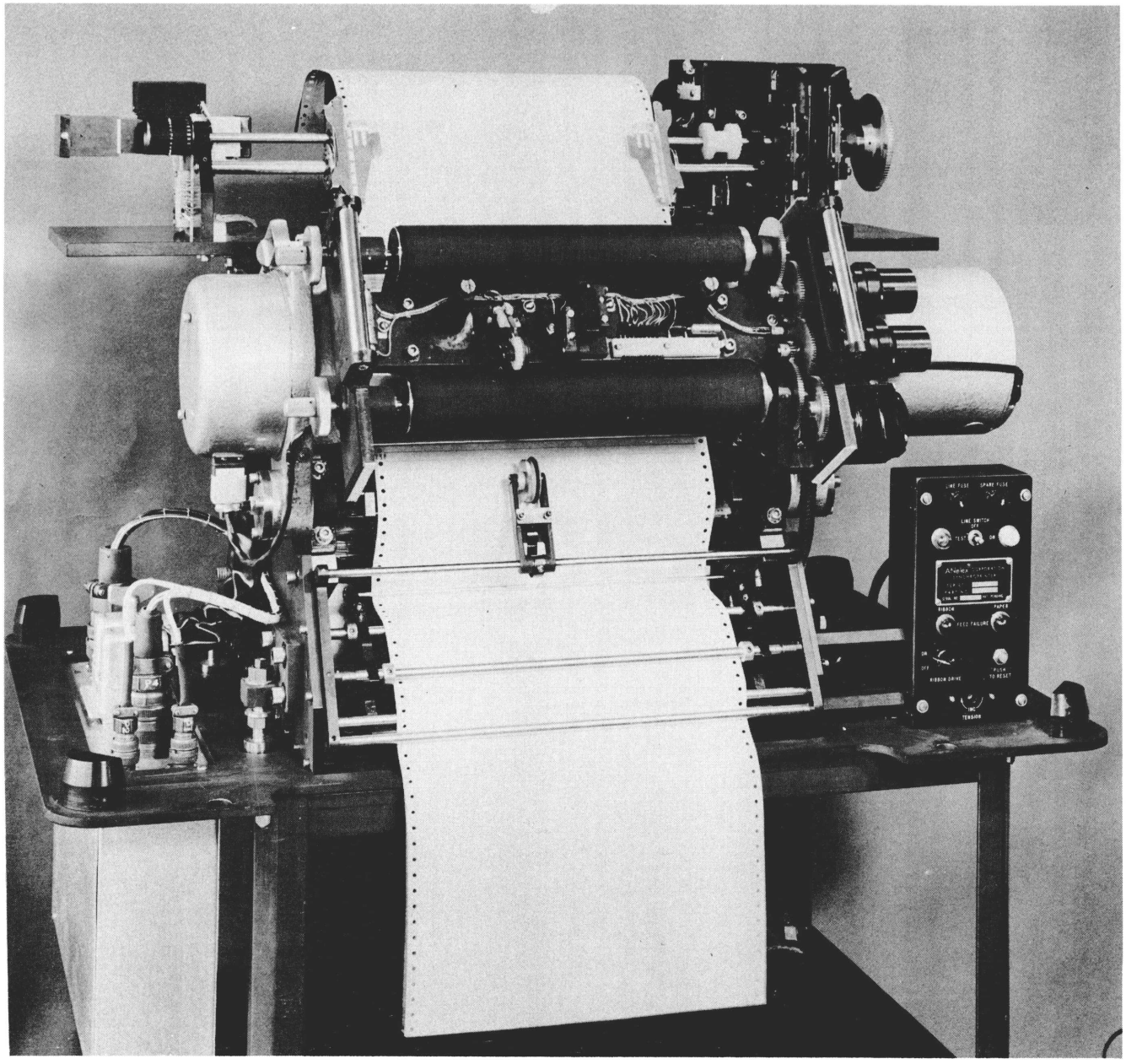


Figure 28 Hi-Speed Printer Chassis

printing and reduces wear on the hammers, print roll, and inked ribbon. A control wheel is used to set this adjustment to cover the range of combinations of copies and paper weights that can be processed in this Printer.

Modular construction is used in the Printer making it easier to maintain and service the equipment. This is exemplified by the hammer-actuator assembly where the hammers and actuators are arranged by groups of four into individual modules, each a completely independent subassembly. Each module is factory adjusted so that all four hammers have the same "pulse-to-print" time within ± 5 microseconds.

The ink silk ribbon used is 15 yards long and is self-reversing. Printing life per ribbon is approximately 100 hours.

The off-line high speed Printer system will be composed of three basic units. First, a Tape Transport Unit, identical to the units used in the magnetic tape storage system. This would include: a tape reel servo, a drive control system, and a 16 channel read/write amplifier section.

The second unit would contain the printer control electronics, power supplies and blowers.

The third unit would house the printer, provide a master control panel station for the system, include the conversion, intermediate buffer storage and program control, and provide racks and guides to accommodate fan fold sprocketed feed paper.

7.9 Punched Card System

The punch card reader and punch units are housed in the same cabinet each with its own feed, drive and sensing system. The feed bins can handle approximately 800 cards. The unit can read 200 cards per minute and punch 100 cards per minute. The Magnetic Tape to Punch Card System consists of three units; the 1st unit contains the card handling and read and punch equipment, the 2nd unit contains the electronic circuits and buffers, the 3rd unit houses the tape transport unit.

APPENDICES

APPENDIX A

TRANSAC S-2000 COMMAND CODE CONSTRUCTION

The command code construction on TRANSAC S-2000 uses the principle that individual bits of the code have meaning.

This will be illustrated in several ways:

1. Chart I divides the instructions into 5 groups as listed on page 10 by bit coding. Branch diagram II shows classes 2 through 5.
2. Chart II shows the bit structure of arithmetic instructions. Branch diagram I corresponds to this.
3. Chart III shows the transfer instructions. Branch diagram III corresponds to this.
4. Chart IV shows the bit structure for the shift instructions. Branch diagram IV illustrates this.
5. Chart V shows the bit structure for the jump instructions. Block diagram V illustrates this.
6. Chart VI shows the bit structure for the special instructions. Branch diagram VI illustrates this.

Following these a complete list of computer instructions is given.

First we need to define some terms used in listing instructions.

Definition of Terms used in explanations of command codes on TRANSAC S-2000

A A register

D D register

Q Q register

() Contents of

V An address in magnetic core storage

VL The left half word at memory location

V. This is considered as an instruction

VR The right half word at memory location

V, Also considered as an instruction

PR The program register which holds two instructions

Io The left half of PR

I_1 The right half of PR

R A bit used to specify index register modification

D_{AL} The left address portion of the word in the D register if it were considered as an instruction word. This is 12 bits if the corresponding R bit is one. It is the length of the complete memory address if the R bit is zero.

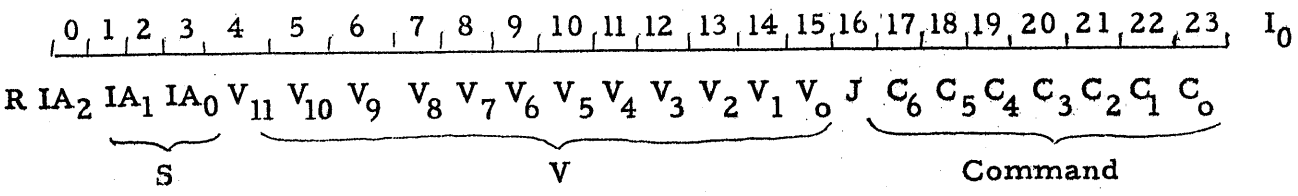
D_{AR} The right address portion of the word in the D register considered as an instruction word.

R_{IA} Index Registers

OVF Overflow flip flop

ICO Inhibit clearing of overflow

Form of an instruction in TRANSAC S-2000



R Determines index register modification

IA Selects the index register or is used for the most significant bits of V when R is zero and the memory is larger than 4096 words.

V Is the address portion of the instruction. The command contains 8 bits J C_6 through C_0.

Multiplication

Unrounded multiplication places the product in A and Q with the more significant half in A. The sign of the product is repeated in the sign bit of Q.

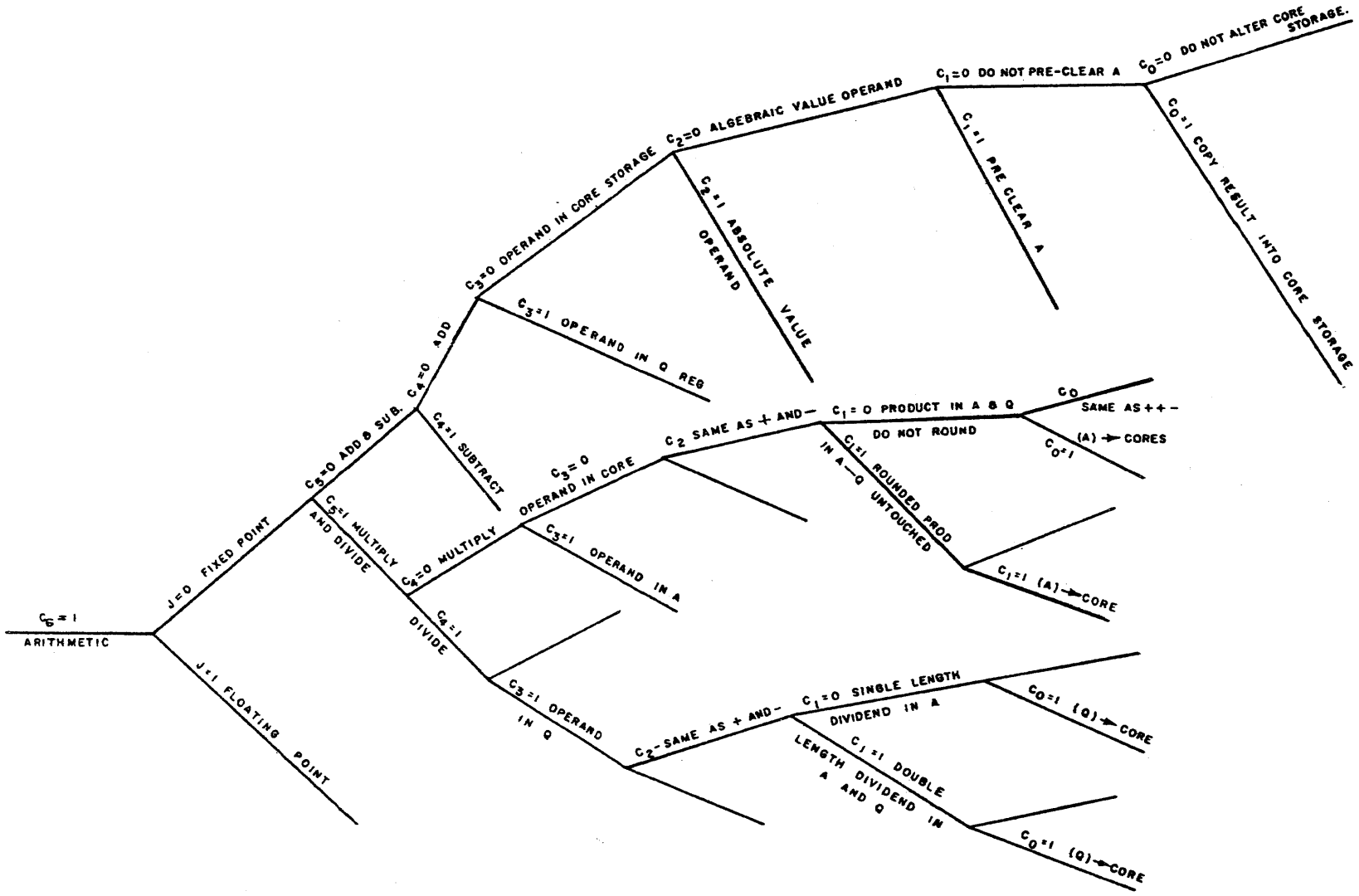
Rounded multiplication places the product in A only (more significant part). It does not destroy Q.

In either case, when the result is copied into memory, it is the contents of the A register which is copied.

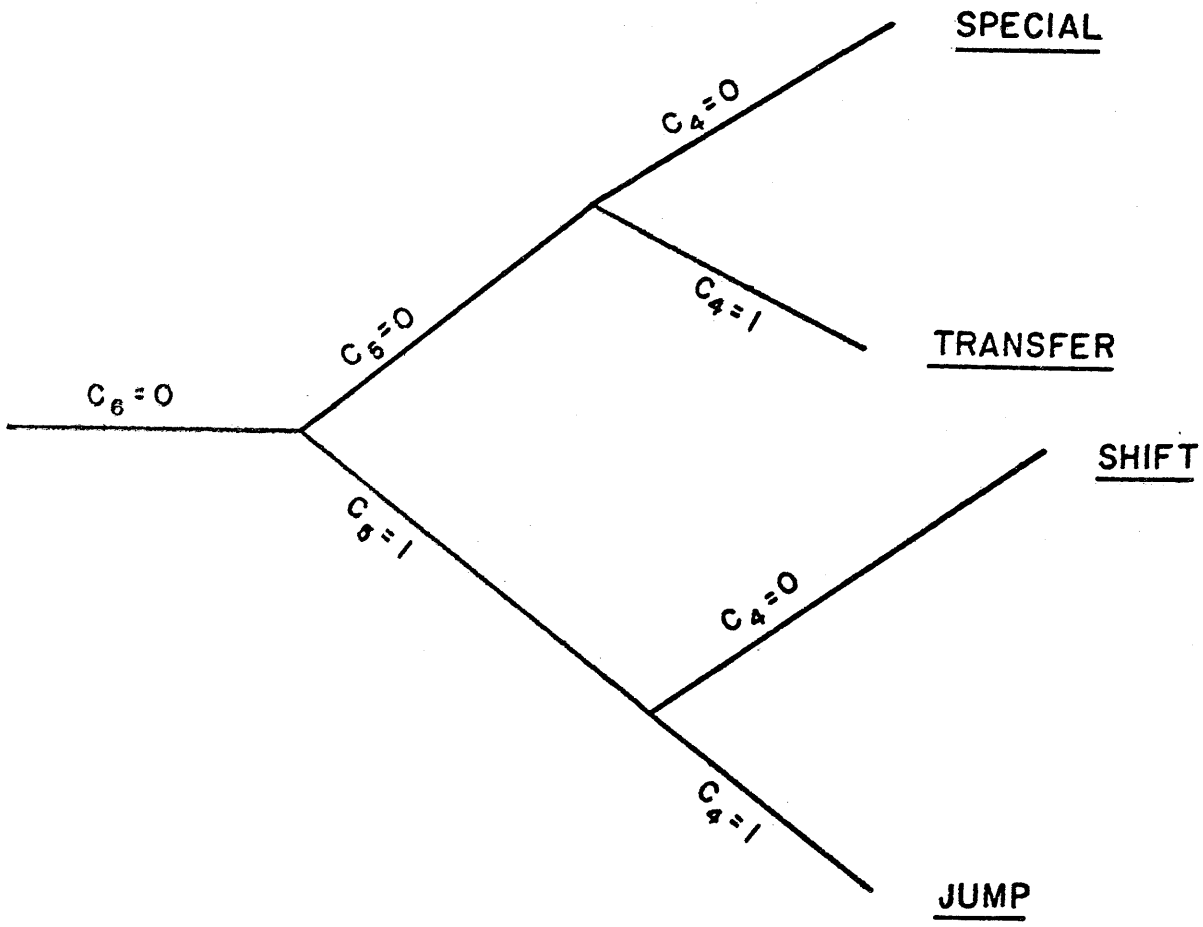
Division

A single or double length dividend may be used.

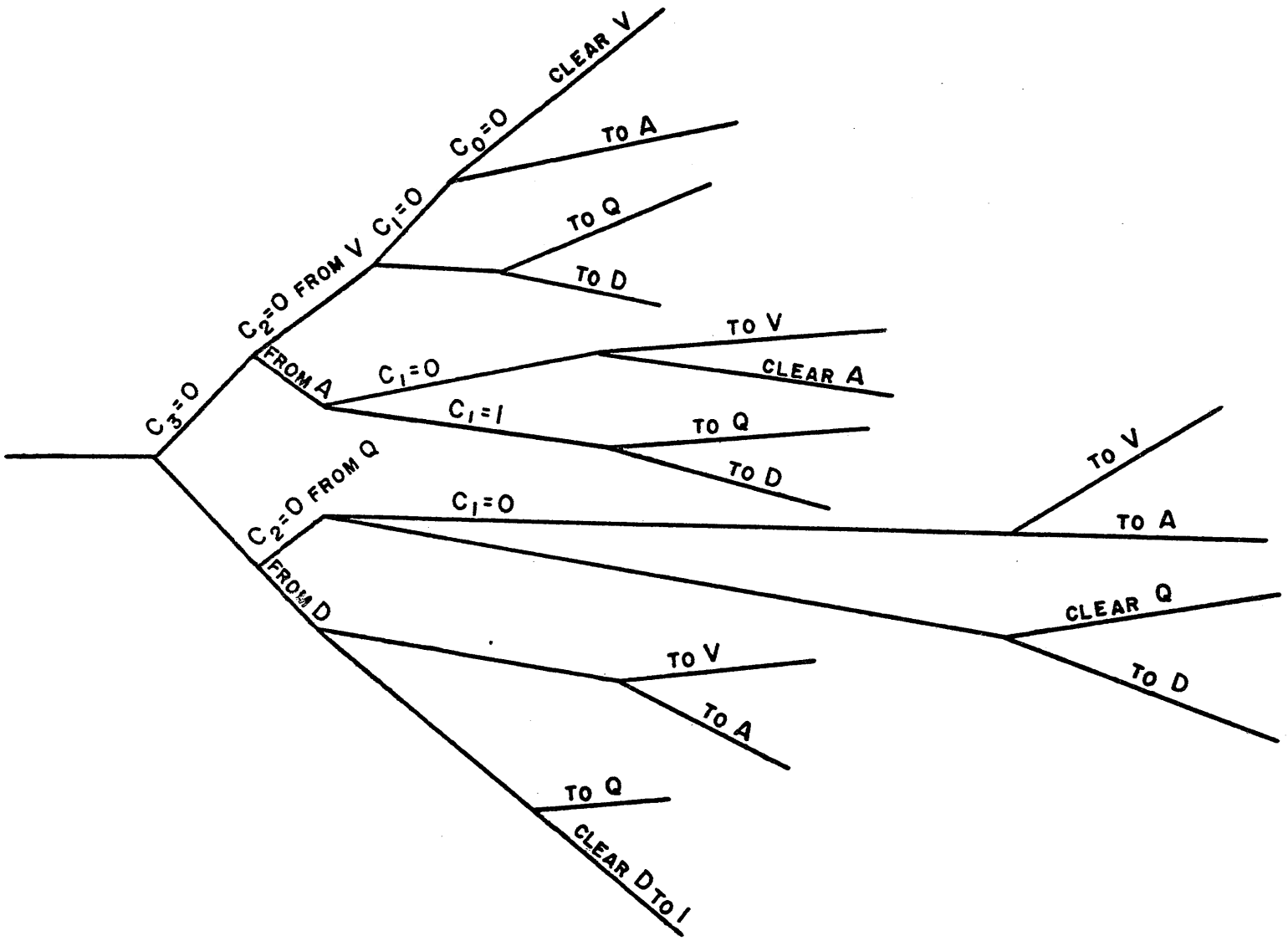
In all cases the quotient is in Q and the remainder in A.



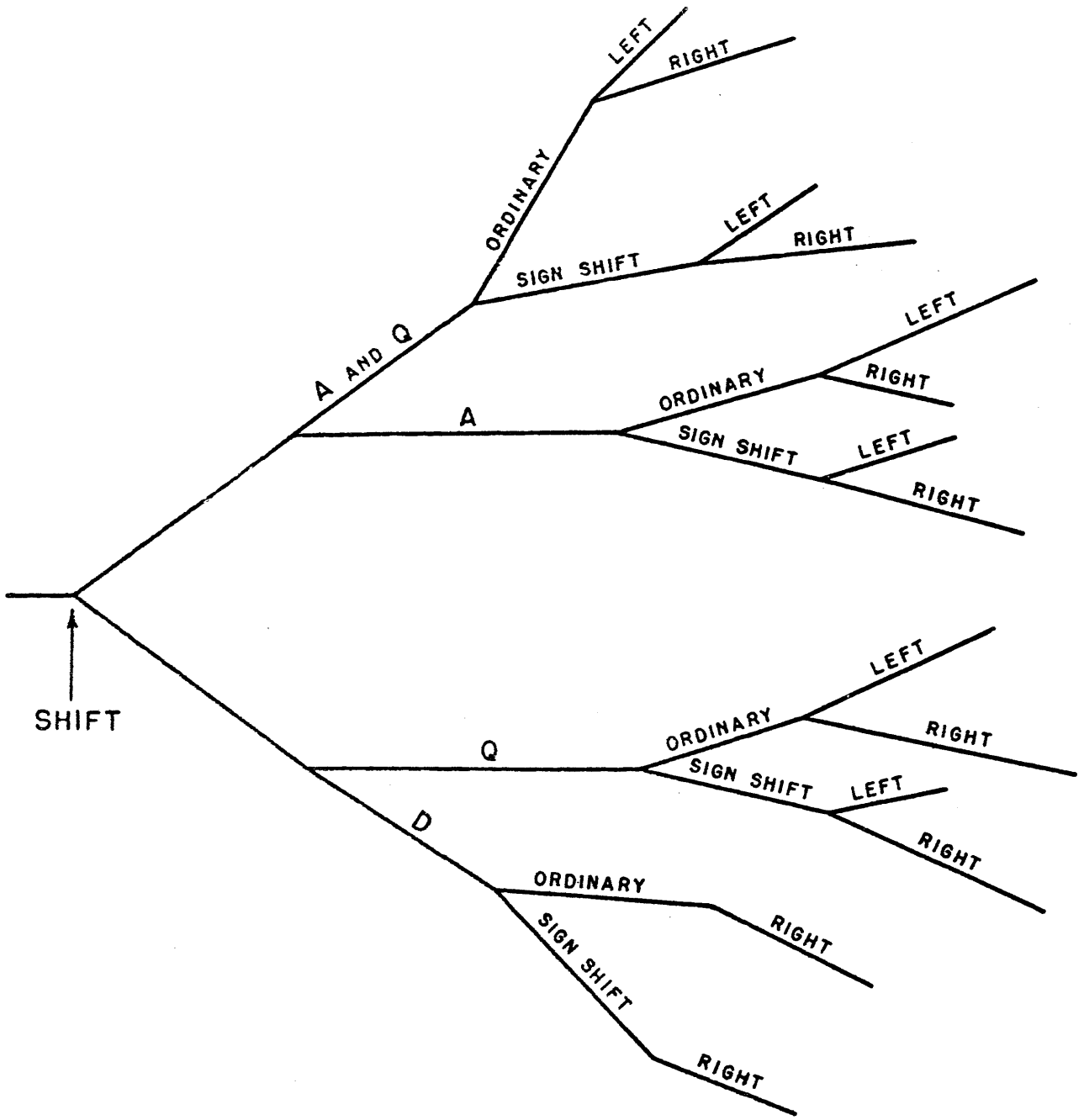
BRANCH DIAGRAM I



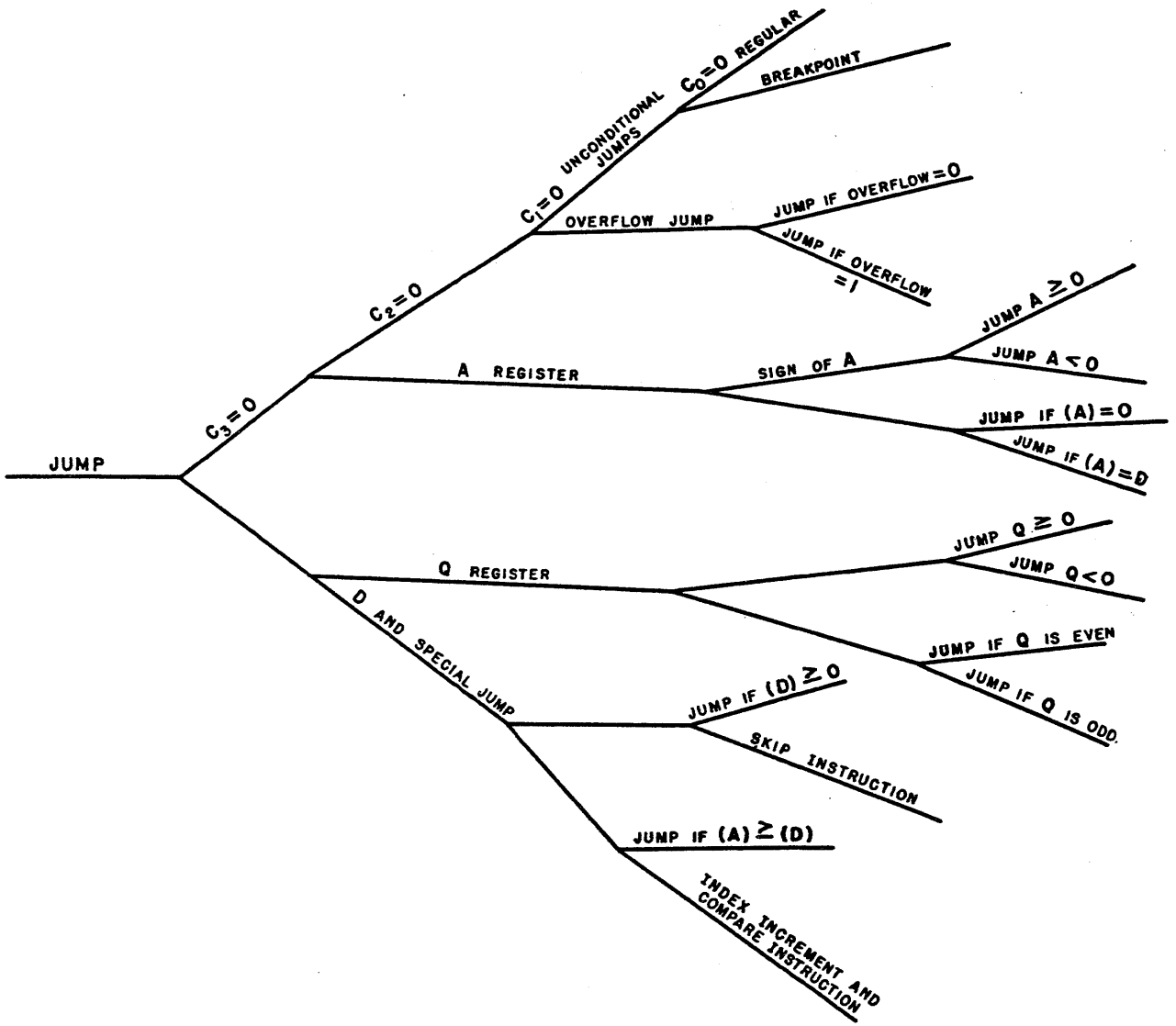
BRANCH DIAGRAM II



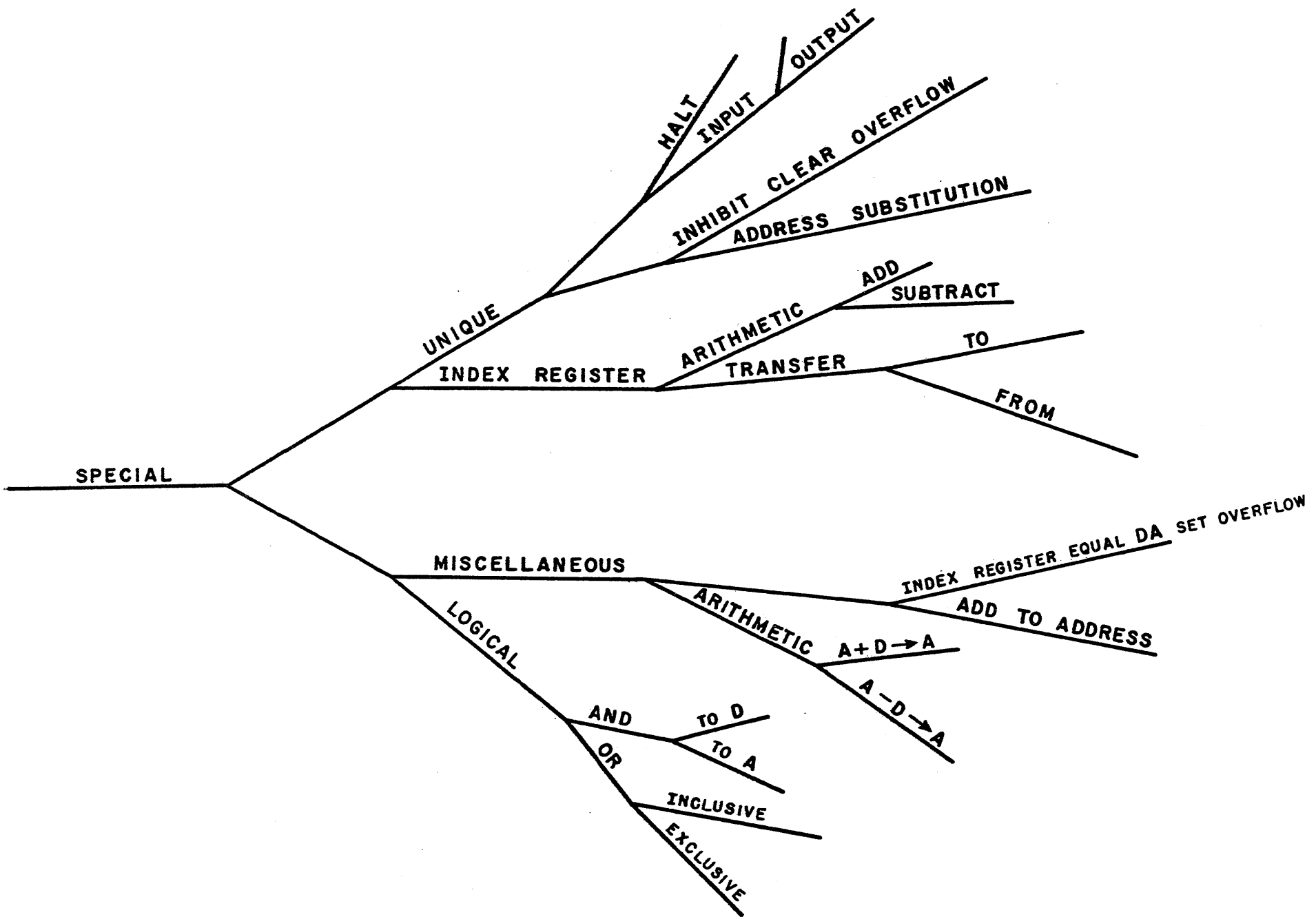
BRANCH DIAGRAM III



BRANCH DIAGRAM IV



BRANCH DIAGRAM V



BRANCH DIAGRAM VI

CHART 1 - COMMAND CONSTRUCTION

J C_6 C_5 C_4 C_3 C_2 C_1 C_o	Meaning
x 1 x x x x x x	Arithmetic
x 0 x x x x x x	Non - Arithmetic
x 0 0 0 x x x x	Special
x 0 0 1 x x x x	Transfers
x 0 1 0 x x x x	Shift
x 0 1 1 x x x x	Jumps

CHART II ARITHMETIC INSTRUCTIONS

J C_6 C_5 C_4 C_3 C_2 C_1 C_o	Meaning
0 1 x x x x x x	Fixed Point Arithmetic
1 1 x x x x x x	Floating Point Arithmetic
x 1 0 0 x x x x	Add
x 1 0 1 x x x x	Subtract
x 1 1 0 x x x x	Multiply
x 1 1 1 x x x x	Divide
x 1 x x 0 x x x	Operand in Memory
x 1 x x 1 x x x	Operand in Arithmetic Register
x 1 x x x 0 x x	Algebraic Value
x 1 x x x 1 x x	Absolute Value
x 1 0 x x x 0 x	Do Not Preclear A [Add + Subtract]

CHART II ARITHMETIC INSTRUCTIONS (cont'd)	
J C_6 C_5 C_4 C_3 C_2 C_1 C_0	Meaning
x 1 0 x x x 1 x	Preclear A
x 1 1 0 x x 0 x	Do Not Round (Mult)
x 1 1 0 x x 1 x	Round and Retain Q (Mult)
x 1 1 1 x x 0 x	Single Length Dividend (divide)
x 1 1 1 x x 1 x	Double Length Dividend
x 1 x x x x x 0	Do Not Alter Core Storage
x 1 x x x x x 1	Copy Result in Core Storage

CHART III TRANSFER INSTRUCTIONS	
J C_6 C_5 C_4 C_3 C_2 C_1 C_0	Meaning
x 0 0 1 0 0 x x	From Memory
x 0 0 1 0 1 x x	From A Register
x 0 0 1 1 0 x x	From Q Register
x 0 0 1 1 1 x x	From D Register
x 0 0 1 x x 0 0	To Memory
x 0 0 1 x x 0 1	To Register A
x 0 0 1 x x 1 0	To Q Register
x 0 0 1 x x 1 1	To D Register
x 0 0 1 0 0 0 0	Clear Memory
x 0 0 1 0 1 0 1	Clear A Register

CHART III TRANSFER INSTRUCTIONS (Cont'd)	
J $C_6C_5C_4C_3C_2C_1C_0$	Meaning
x 0 0 1 1 0 1 0	Clear Q Register
x 0 0 1 1 1 1 1	Clear D Register, to all one's

CHART IV SHIFT INSTRUCTIONS	
J $C_6C_5C_4C_3C_2C_1C_0$	Meaning
x 0 1 0 0 0 x x	Shift A + Q
x 0 1 0 0 1 x x	Shift A
x 0 1 0 1 0 x x	Shift Q
x 0 1 0 1 1 x x	Shift D
x 0 1 0 x x 0 x	Ordinary Shift (Treat All Bits Alike)
x 0 1 0 x x 1 x	Sign Shift (Consider as two's complement number)
x 0 1 0 x x x 0	Shift Left
x 0 1 0 x x x 1	Shift Right

CHART V JUMP INSTRUCTIONS	
J $C_6C_5C_4C_3C_2C_1C_0$	Meaning
x 0 1 1 0 0 0 0	Unconditional Jump
x 0 1 1 0 0 0 1	Unconditional Breakpoint Jump
x 0 1 1 0 0 1 x	Overflow Flipflop Jump
x 0 1 1 0 1 x x	Involves A Register
x 0 1 1 0 1 0 x	Involves Sign of A

CHART V JUMP INSTRUCTIONS (Cont'd)

J	C_6	C_5	C_4	C_3	C_2	C_1	C_0	Meaning
x	0	1	1	0	1	1	0	Jump if (A) = 0
x	0	1	1	0	1	1	1	Jump if (A) = (D)
x	0	1	1	1	0	x	x	Involves Q Reg. (Circular Shift Q in) (direction of sensing)
x	0	1	1	1	0	0	x	Involves Sign of Q
x	0	1	1	1	0	1	x	Involves Least Significant Bit of Q
x	0	1	1	x	x	x	0	Jump if Zero (if bit is involved)
x	0	1	1	x	x	x	1	Jump if one (if bit is involved)
x	0	1	1	1	1	0	0	Jump if D is Positive
x	0	1	1	1	1	0	1	Magnitude Jump
x	0	1	1	1	1	1	0	Skip Instructions
x	0	1	1	1	1	1	1	Increment and Compare Index Register Instructions

CHART VI SPECIAL INSTRUCTIONS

J	C_6	C_5	C_4	C_3	C_2	C_1	C_0	Unique Control
x	0	0	0	0	0	x	x	Unique Control
0	0	0	0	0	0	0	0	HALT
1	0	0	0	0	0	0	0	Breakpoint Halt
x	0	0	0	0	0	0	1	Transfer Control to Input/Output

CHART VI SPECIAL INSTRUCTIONS (Cont'd)

J	C_6	C_5	C_4	C_3	C_2	C_1	C_0		Unique Control
x	0	0	0	0	0	1	0		Inhibit Clear of Overflow Flip Flop
x	0	0	0	0	0	1	1		Address Substitution (PA previous I to address)
x	0	0	0	0	1	x	x		**Index Register Instructions**
x	0	0	0	0	1	0	0		Add B_{as} to R_s
x	0	0	0	0	1	0	1		Subtract D_{aj} from R_s
x	0	0	0	0	1	1	0		Set R_s = to D_{as}
x	0	0	0	0	1	1	1		Transfer R_s to D_{aj}
x	0	0	0	1	0	x	x		**Miscellaneous**
x	0	0	0	1	0	0	0		Set Overflow Equal to One if $R_s = D_{aj}$
x	0	0	0	1	0	0	1		Add 1 to V, right or left address
x	0	0	0	1	0	1	0		Add D to A fixed point
x	0	0	0	1	0	1	0		Add D to A Floating Point
x	0	0	0	1	0	1	1		Subt D from A, Fixed Point
x	0	0	0	1	0	1	1		Subt D from A, Floating Point,
x	0	0	0	1	1	x	x		**Logical Orders**
x	0	0	0	1	1	0	0		"Logical And" Q . V⟶D
x	0	0	0	1	1	0	1		"Logical And" Q . V⟶A
x	0	0	0	1	1	1	0		"Logical Inclusive Or" D ∨ V⟶V
x	0	0	0	1	1	1	1		"Logical Exclusive Or" A∨ V⟶V

Numeric Code						Meaning
J		C_6	X_2	X_1	X_0	
0		1	0	0	0	(V) + (A) ⟶ A
1		1	0	0	0	(V) + (A) ⟶ A [Floating Point]
0		1	0	0	1	(V) + (A) ⟶ V
1		1	0	0	1	(V) + (A) ⟶ V [Floating Point]
0		1	0	0	2	(V) ⟶ A [TVA is Quicker]
1		1	0	0	2	(V) ⟶ A [TVA is Quicker]
0		1	0	1	0	1(V)1 + A ⟶ A
1		1	0	1	0	1(V)1 + A ⟶ A Float
0		1	0	1	1	1(V)1 + A ⟶ V
1		1	0	1	1	1(V)1 + A ⟶ V Float
0		1	0	1	2	1(V)1 ⟶ A
1		1	0	1	2	1(V)1 ⟶ A Float
0		1	0	1	3	1(V)1 ⟶ V
1		1	0	1	3	1(V)1 ⟶ V Float
0		1	0	2	0	(Q) + (A) ⟶ A
1		1	0	2	0	(Q) + (A) ⟶ A Float

Numeric Code						Meaning	
J		C_6	X_2	X_1	X_0		
0		1	0	2	1	(Q) + (A) ⟶ V	
1		1	0	2	1	(Q) + (A) ⟶ V	Float
0		1	0	2	2	(Q) ⟶ (A)	
1		1	0	2	2	(Q) ⟶ A	Float
0		1	0	2	3	(Q) ⟶ V	
1		1	0	2	3	(Q) ⟶ V	Float
0		1	0	3	0	1(Q)1 + (A) ⟶ A	
1		1	0	3	0	1(Q)1 + (A) ⟶ A	Float
0		1	0	3	1	1(Q)1 + (A) ⟶ V	
1		1	0	3	1	1(Q)1 + (A) ⟶ V	Float
0		1	0	3	2	1(Q)1 ⟶ A	
1		1	0	3	2	1(Q)1 ⟶ A	Float
0		1	0	3	3	1(Q)1 ⟶ V	
1		1	0	3	3	1(Q)1 ⟶ V	Float
0		1	1	0	0	(A) - (V) ⟶ A	
1		1	1	0	0	(A) - (V) ⟶ A	Float
0		1	1	0	1	(A) - (V) ⟶ V	
1		1	1	0	1	(A) - (V) ⟶ V	Float
0		1	1	0	2	-(V) ⟶ A	
1		1	1	0	2	-(V) ⟶ A	Float

Numeric Code					Meaning	
J	C_6	X_2	X_1	X_0		
0	1	1	0	3	-(V) ⟶ V	
1	1	1	0	3	-(V) ⟶ V	Float
0	1	1	1	0	(A) - |V| ⟶ A	
1	1	1	1	0	(A) - |V| ⟶ A	Float
0	1	1	1	1	(A) - |V| ⟶ V	
1	1	1	1	1	(A) - |V| ⟶ V	Float
0	1	1	1	2	-|V| ⟶ A	
1	1	1	1	2	-|V| ⟶ A	Float
0	1	1	1	3	-|V| ⟶ V	
1	1	1	1	3	-|V| ⟶ V	Float
0	1	1	2	0	A-(Q) ⟶ A	
1	1	1	2	0	A - (Q) ⟶ A	Float
0	1	1	2	1	A - (Q) ⟶ V	
1	1	1	2	1	A - (Q) ⟶ V	Float
0	1	1	2	2	-(Q) ⟶ A	
1	1	1	2	2	-(Q) ⟶ A	Float
0	1	1	2	3	-(Q) ⟶ V	
1	1	1	2	3	-(Q) ⟶ V	Float
0	1	1	3	0	(A) - 1 (Q) 1 ⟶ A	
1	1	1	3	0	(A) - 1 (Q) 1 ⟶ A	Float

Numeric Code						Meaning
J		C_6	X_2	X_1	X_0	
0		1	1	3	1	(A) - 1 (Q) 1 → V
1		1	1	3	1	(A) - 1 (Q) 1 → V Float
0		1	1	3	2	-1 (Q) 1 → A
1		1	1	3	2	-1 (Q) 1 → A Float
0		1	1	3	3	-1 (Q) 1 → V
1		1	1	3	3	-1 (Q) 1 → V Float
0		1	2	0	0	(Q) x (V) → A and Q
1		1	2	0	0	(Q) x (V) → A and Q Float
0		1	2	0	1	(Q) x (V) → V
1		1	2	0	1	(Q) x (V) → V Float
0		1	2	0	2	{(Q) x (V)} Rounded → A
1		1	2	0	2	{(Q) x (V)} Rounded → A Float
0		1	2	0	3	{(Q) x (V)} Rounded → V
1		1	2	0	3	{(Q) x (V)} Rounded → V Float
0		1	2	1	0	(Q) x |V| → A, Q
1		1	2	1	0	(Q) x |V| → A, Q Float
0		1	2	1	1	(Q) x |V| → V
1		1	2	1	1	(Q) x |V| → V Float
0		1	2	1	2	{(Q) x (V)} Rounded → A
1		1	2	1	2	{(Q) x (V)} Rounded → A Float

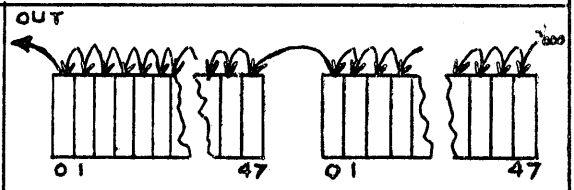
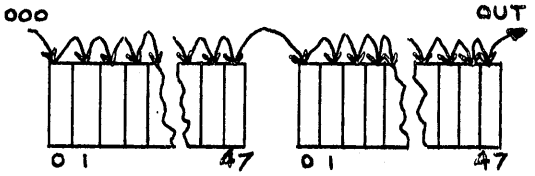
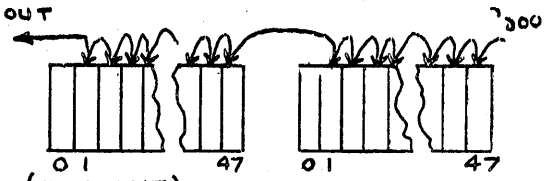
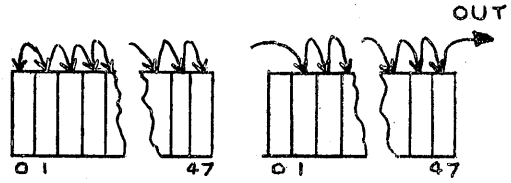
Numeric Code						Meaning
J		C_6	X_2	X_1	X_0	
0		1	2	1	3	{ (Q) x 1 V 1 } Rounded→V
1		1	2	1	3	{ (Q) x 1 V 1 } Rounded→ V Float
0		1	2	2	0	(Q) x (A) → A, Q
1		1	2	2	0	(Q) x (A) → A, Q Float
0		1	2	2	1	(Q) x (A) → V
1		1	2	2	1	(Q) x (A) → V Float
0		1	2	2	2	{ (Q) x (A)} → A Rounded
1		1	2	2	2	{ (Q) x (A)} → A Rounded
0		1	2	2	3	{ (Q) x (A)} RD. → V
1		1	2	2	3	{ (Q) x (A)} RD → V Float
0		1	2	3	0	1 (Q) 1 x (A) → A, Q
1		1	2	3	0	1 (Q) 1 x (A) → A, Q Float
0		1	2	3	1	1 (Q) 1 x (A) → V
1		1	2	3	1	1 (Q) 1 x (A) → V Float
0		1	2	3	2	{1 (Q) 1 x (A)} RD → A
1		1	2	3	2	{1 (Q) 1 x (A)} RD → A Float
0		1	2	3	3	{1 Q 1 x (A)} RD → V
1		1	2	3	3	{1 Q 1 x (A)} RD → V Float
0		1	3	0	0	(A)/(V) → Q
1		1	3	0	0	(A) ÷ (V) → Q Float

Numeric Code							Meaning
J		C_6	X_2	X_1	X_0		
0		1	3	0	1		(A) ÷ (V) → V
1		1	3	0	1		(A) ÷ (V) → V Float
0		1	3	0	2		(A) ÷ (V) → Q (double length) dividend
1		1	3	0	2		(A) ÷ (V) → Q (double length) Dividend Floating
0		1	3	0	3		(A) ÷ (V) → V (double length)
1		1	3	0	3		(A) ÷ (V) → V (double length) Floating
0		1	3	1	0		(A) ÷ 1 (V) 1 → Q
1		1	3	1	0		(A) ÷ 1 (V) 1 → Q Float
0		1	3	1	1		(A) ÷ 1 (V) 1 → V
1		1	3	1	1		(A) ÷ 1 (V) 1 → V Float
0		1	3	1	2		(A) ÷ 1 (V) 1 double length → Q
1		1	3	1	2		A ÷ 1 (V) 1 double length → Q Float
0		1	3	1	3		A ÷ 1 (V) 1 double length → V
1		1	3	1	3		A ÷ 1 (V) 1 double length → V Float
0		1	3	2	0		(A) ÷ (Q) → Q
1		1	3	2	0		(A) ÷ (Q) → Q Float
0		1	3	2	1		(A) ÷ (Q) → V
1		1	3	2	1		(A) ÷ (Q) → V Float

Numeric Code							Meaning
J		C_6	X_2	X_1	X_0		
0		1	3	2	2		(A) ÷ (Q) double length → Q
1		1	3	2	2		(A) ÷ (Q) double length → Q Float
0		1	3	2	3		(A) ÷ (Q) double length → V
1		1	3	2	3		(A) ÷ (Q) double length → V Float
0		1	3	3	0		(A) ÷ 1 (Q) 1 → Q
1		1	3	3	0		(A) ÷ 1 (Q) 1 → Q
0		1	3	3	1		(A) ÷ 1 (Q) 1 → V
1		1	3	3	1		(A) ÷ 1 (Q) 1 → V
0		1	3	3	2		(A) ÷ 1 (Q) 1 double length → Q
1		1	3	3	2		(A) ÷ 1 (Q) 1 double length → Q
0		1	3	3	3		(A) ÷ 1 (Q) 1 double length → V
1		1	3	3	3		(A) ÷ 1 (Q) 1 double length → V
							End of Arithmetic Section
0		0	0	0	0		Halt!
1		0	0	0	0		Breakpoint Halt
0		0	0	0	1		Transfer Control to Input/Output
0		0	0	0	2		Inhibit Clearing of Overflow Flip Flop.
0		0	0	0	3		Address Substitution, Left.
1		0	0	0	3		Address Substitution, Right

J	C_6	X_2	X_1	X_0	
0	0	0	1	0	Index Reg. + Address Portion of D ⟶ Index Register
1	0	0	1	0	
0	0	0	1	1	Index Reg. − Address Portion of D ⟶ Index Register.
1	0	0	1	1	
0	0	0	1	2	Transfer D_{a_j} to Index Reg. Address
1	0	0	1	2	j= L or R
0	0	0	1	3	Transfer Index Contents ⟶ D_{aj} j= L or R
1	0	0	1	3	$(R_i) \rightarrow D_{aj}$) ($D_{aj} \rightarrow R_i$)
0	0	0	2	0	If Index Register = to D_{aj},
1	0	0	2	0	set overflow Flip Flop ⟶ One.
0	0	0	2	1	Add 1 to address portion of V;
1	0	0	2	1	Lor R Instruction Specified by J.
0	0	0	2	2	(A) + (D) ⟶ A
1	0	0	2	2	(A) + (D) ⟶ A Float
0	0	0	2	3	(A) - (D) ⟶ A
1	0	0	2	3	(A) - (D) ⟶ A
0	0	0	3	0	"bit by bit" Logical and of Q + V ⟶ D
0	0	0	3	1	Logical And (Q) x (V) ⟶ A
0	0	0	3	2	"Bit by Bit" Logical Sum (Inclusive or 1+1 = 1+0 = 0+1 = 1

Numeric Code						Meaning
J		C_6	X_2	X_1	X_0	
0		0	0	3	3	"bit by bit" Exclusive or [Add without carry] 1+1 = 0+0 =0 1+0 = 0+1 =1
0		0	1	0	0	'000 ⟶ V
0		0	1	0	1	(V) ⟶ A
0		0	1	0	2	(V) ⟶ Q
0		0	1	0	3	(V) ⟶ D
0		0	1	1	0	(A) ⟶ V
0		0	1	1	1	'000 ⟶ A
0		0	1	1	2	(A) ⟶ Q
0		0	1	1	3	(A) ⟶ D
0		0	1	2	0	(Q) ⟶ V
0		0	1	2	1	(Q) ⟶ A
0		0	1	2	2	'O ⟶ Q
0		0	1	2	3	(Q) ⟶ D
0		0	1	3	0	(D) ⟶ V
0		0	1	3	1	(D) ⟶ A
0		0	1	3	2	(D) ⟶ Q
0		0	1	3	3	Clear D to one's End of Transfer Ins.

Numeric Code				Meaning
J		C_6 X_2 X_1 X_0		
0		0 2 0 0		 A Register Q Register
0		0 2 0 1		 A Register Q Register
0		0 2 0 2		 $(a_1 \rightarrow OUT)$ A Register Q Register Sign Shift Left, A + Q
0		0 2 0 3		 $(a_0 \rightarrow a_0\ a_0 \rightarrow a_1)$ A Register Q Register Sign Shift Right, A + Q
0		0 2 1 0		Shift Left A
0		0 2 1 1		Shift Right A
0		0 2 1 2		Sign Shift Left of Register A.
0		0 2 1 3		Sign Shift Right, Register A

Numeric Code				Meaning
J		C_6 X_2 X_1 X_0		
0		0 2 2 0		Shift Left, Q
0		0 2 2 1		Shift Right, Q
0		0 2 2 2		Sign Shift, Left, Q
0		0 2 2 3		Sign Shift, Right, Q
0		0 2 3 0		Command Fault
0		0 2 3 1		Shift Right, D
0		0 2 3 2		Command Fault
0		0 2 3 3		Sign Shift Right, D
0		0 3 0 0		Unconditional Jump Left
1		0 3 0 0		Unconditional Jump Right
0		0 3 0 1		Unconditional Breakpoint Jump L.
1		0 3 0 1		Unconditional Breakpoint Jump R.
0		0 3 0 2		Jump to Left if no Overflow (Overflow f.f = 0)
1		0 3 0 2		Jump to Right if no Overflow (Overflow f.f = 0)
0		0 3 0 3		Jump to Left if Overflow (ovf = 1)
1		0 3 0 3		Jump to Right if Overflow (ovf = 1)
0		0 3 1 0		Jump to Left if A $\geq$ 0
1		0 3 1 0		Jump to Right if A $\geq$ 0

Numeric Code				Meaning
J		C_6 X_2 X_1X_0		
0		0 3 1 1		Jump to Left if A < 0
1		0 3 1 1		Jump to Right if A < 0
0		0 3 1 2		Jump to Left if A = 0
1		0 3 1 2		Jump to Right if A = 0
0		0 3 1 3		Jump to Left if (A) = (D)
1		0 3 1 3		Jump to Right if (A) = (D)
0		0 3 2 0		Jump to Left if Q ≥ 0
1		0 3 2 0		Jump to Right if Q ≥ 0
0		0 3 2 1		Jump to Left if Q < 0
1		0 3 2 1		Jump to Right if Q < 0
0		0 3 2 2		Jump to Left if Q is even
1		0 3 2 2		Jump to Right if Q is even
0		0 3 2 3		Jump to Left if Q is Odd.
1		0 3 2 3		Jump to Right if Q is Odd.
0		0 3 3 0		Jump to Left if D ≥ 0
1		0 3 3 0		Jump to Right if D ≥ 0
0		0 3 3 1		Skip Next Instruction if: (A) > (Counter) [address portion of instruction] [specified by Index Reg. Bits] Otherwise, Perform next Instruct.

Numeric Code						Meaning
J		C_6	X_2	X_1	X_0	
1		0	3	3	1	Skip Next instruction if the fault specified by the address portion of the instruction in the equipment specified by the index register bits has not occurred. If the fault has occurred, perform the next instruction.
0		0	3	3	2	Jump to Left if (A) $\geq$ D
1		0	3	3	2	Jump to Right if (A) $\geq$ D.
0		0	3	3	3	Add One to specified Index Reg. Jump to the right of D if the left of D is not equal to Index Register Contents.
1		0	3	3	3	Add 1 to specified Index Register. Set Overflow to one if left of D equals index register contents.

APPENDIX B

USE OF INDEX REGISTERS IN TRANSAC S-2000

Index registers in the TRANSAC S-2000 computer have four basic functions. These are:

1) Address modification - In a file maintenance type operation, it is necessary to pass each record through the computer memory making various tests and performing other operations as required. Usually the program to perform this includes many instructions to successively modify address locations to allow this stepping-through procedure. With the use of index registers, the program is written to include the testing and processing only. The address modification, that is, increasing certain addresses by fixed increments, is achieved by presetting an index register and referencing it for each new record being processed. This requires only one additional instruction which is executed in 18 microseconds including all memory accesses.

2) Counting - By using this same index as a counter, we can determine whether a new group of input items is to be read into the computer. Based on memory availability as determined by the programs whenever the index register reaches a certain condition, determined by testing the contents of the index register and a constant in the program, a new read order will be executed, bringing in a new string of input information. The index register will be reset at this point and a new count established. This same feature may be used for computer write orders.

3) Address Placement or Substitution - Using the index registers available, address portions of a constant used as part of a variable connector can be substituted through the index register without further manipulation. Also as a substitution tool, it can be used to prepare shift instructions by inserting through the index register the number of places to be shifted.

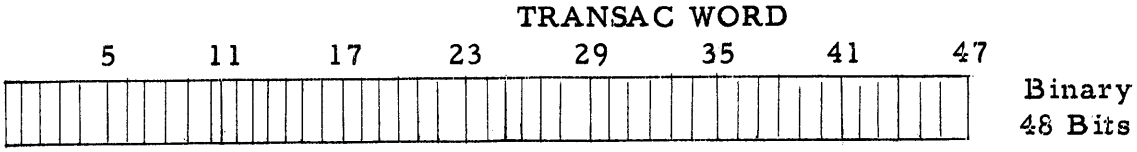
4) Accumulating - The Transac S-2000 index registers perform a valuable function by acting as auxiliary accumulators. Controls required in each computer operation, such as totaling the number of items processed, or the total value of a file of items can be accomplished by accumulating the amounts in an index register. This accumulation is accomplished without alteration of the contents of the basic A (accumulator) register.

All of the above functions result in a considerable saving of valuable memory space by decreasing the total number of instructions required. Also the computer operation will be speeded up since fewer instruction execution times are required.

APPENDIX C

TRANSAC S-2000 WORD CONSTRUCTION AND USAGE

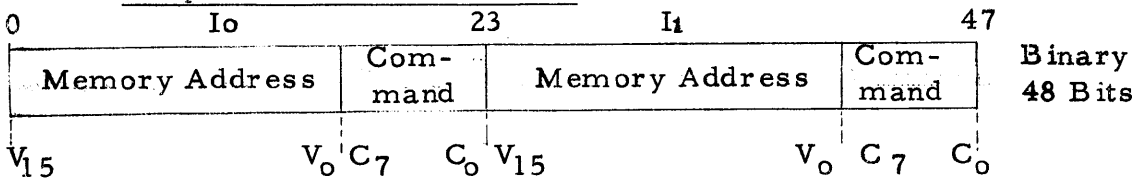
The TRANSAC S-2000 word is 48 binary bits in length, each bit having the value of one or zero. The bits are numbered from left to right. The left most bit is the zero bit, the rightmost, or least significant bit, is the 47th bit. Each number in the memory can be considered as a data word since it only functions as an instruction word if it is present in the instruction register



TRANSAC Word has 3 possible representations in Memory

I. TRANSAC S-2000 is a single address machine, with two 24-bit instructions per word. Each instruction is composed of an 8-bit command and a sixteen bit memory address. Thus, an instruction uses no more than one operand. Each instruction execution follows sequentially except when interrupted by a jump instruction. The left hand instruction in each Instruction word (2 instructions) is executed first. The right hand instruction then is executed. This is followed by the left hand instruction in the next instruction word, etc.

Ia. Simplified Instruction Word



Command = 8 Binary Bits per Instruction

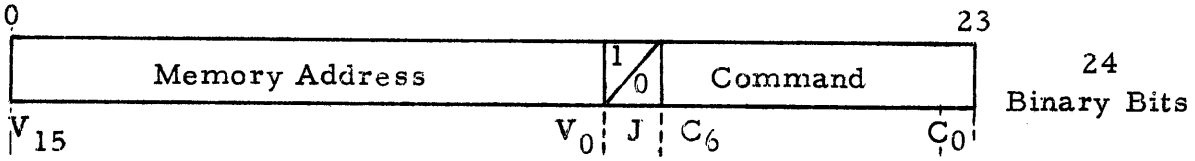
Memory Address = 16 Binary Bits per instruction

The command portion of the TRANSAC S-2000 instruction may be further broken down as follows:

1. The first bit of the command is called the J Bit. It is used with jump instruction to indicate which half of the word being jumped to is to be executed; and with floating point instructions to indicate whether floating point is to be used.

2. The remaining seven bits indicate the command itself.

Ib. One Instruction



$C_6 - C_o$ = Command (7 Bits)

J = If Jump Instruction, tells which half of instruction word being jumped to is to be executed.
= If arithmetic tells whether floating point is involved (1 Bit)

$V_{15} - V_o$ = Memory Address (16 Bits)

The address portion of the TRANSAC S-2000 instruction indicates the following information:

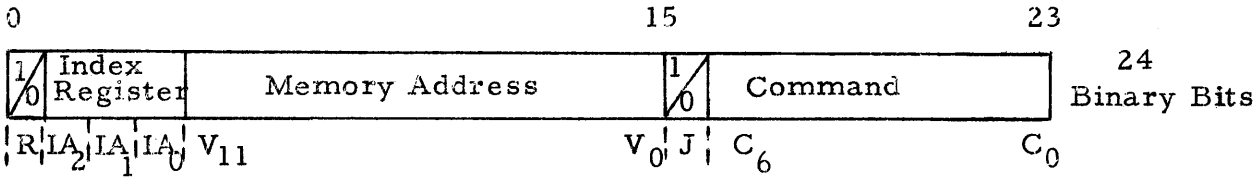
1. If index registers are not present in the system all sixteen bits of the address portion indicate the memory address.

2. If index registers are present in the system, the sixteen bits are broken down as follows:

   a. The first bit of the instruction tells whether an index register is involved for this particular instruction. If an index register is not involved, the remaining 15 bits indicate the memory address or operand being referred to in the instruction. If an index register is involved, the second bit, or the second and third bits, or the second, third, and fourth bits, etc., depending upon the number of index registers in the system, indicate the

address or the number of the index register being used. The remaining bits indicate the memory address. When an index register is involved, the memory location selected is the address indicated in the instruction, modified by the contents of the index register involved.

Ic. One Instruction



I = Index Register Involved (1 Bit)

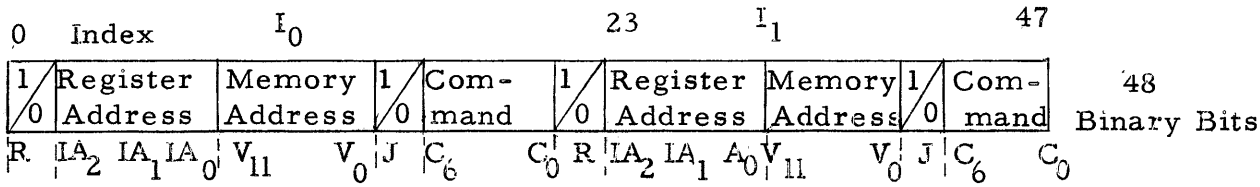
I_2-I_o = Index Register Address (3 Bits) 8 Index Registers

V_{11} -V_o = Memory Address: 0000-4095

J = Jump or Floating Point

C_6 -C_o = Command

Id. Instruction Word



R = Index Register Involved?

A_2-A_1 = Address or Number of Index Register (8 Index Registers)

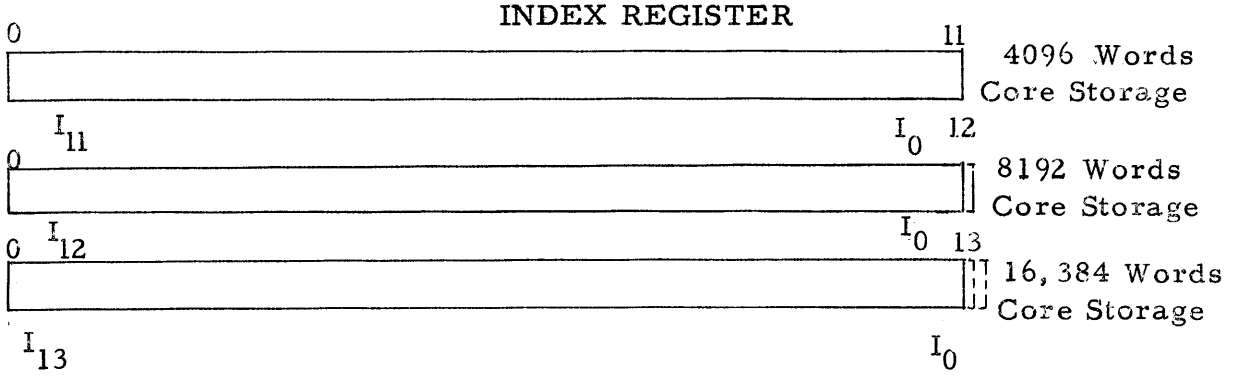
$V_{11}V_o$ = Memory Address: 0000-4095

J = If jump instruction, tells which half of instruction word being jumped to is to be executed.

= If arithmetic, whether floating point is involved.

C_6 -C_o = Command

INDEX REGISTER



II Although the representation of a word in the arithmetic operations must be in binary form, the representation of information in other areas of the system, such as, memory or input/output, may be either in binary or in coded decimal form or in any combination of the two. A data word coming from magnetic tape will eventually be worked on in the computer. If the program has been set up to recognize the coding of the word, it is of no consequence if the representation is binary or coded decimal. It is always the parameters of the program that determine the makeup of the word. This is no more of a restriction than indicating through the program record sizes or field sizes and arrangements. Consistency is the only requirement. Coded decimal representation in TRANSAC S-2000 is six binary bits per character, whether they are alphabetic or numeric characters. Thus, each TRANSAC S-2000 word has the capacity of eight coded decimal characters (8 characters · 6 bits each equals 48 binary bits). The parallel nature of the system offers a unique advantage in being able to compress, on magnetic tape and in the memory, the amount of storage required to represent numeric fields of information. TRANSAC S-2000 represents numeric fields as one whole number rather than a series of individual digits.

Considering the number "1234567", a serial type computer recognizes this number as 1 and 2 and 3 and four and five and six and seven. TRANSAC S-2000 recognizes it as one entry, 1234567. The binary representation of this number is thus:

100101101011010000111 = 21 binary bits

The same number on a serial type computer would read, in coded decimal:

000001 and 000010 and 000011 and 00100 and
000101 and 000110 and 00111 = 42 binary bits.

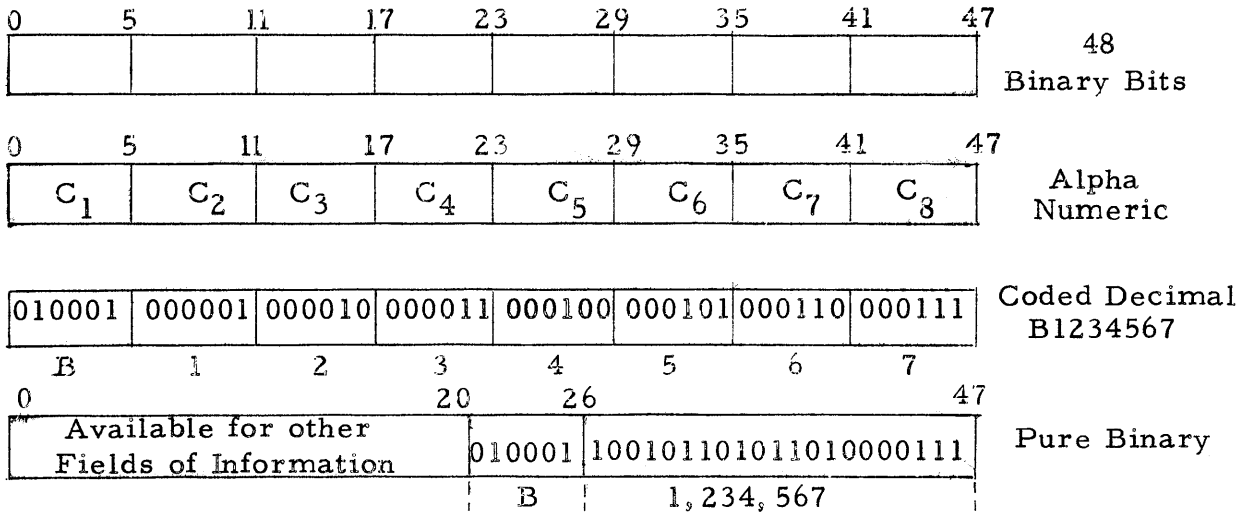
Thus, the gross saving in storage space on magnetic tape and in core storage is 21 bits which would be available to represent other fields of information. Alphabetic characters, in both types of computers, would be represented by six binary bits. Parity bits have not been considered here since they add nothing to or subtract nothing from the net total storage requirements on tape or in the memory for either type of computer.

To offset the savings, on a binary machine, of tape and memory space, is the need to convert input information from punched cards or paper tape from decimal notation to binary when it is to be used in an arithmetic operation. This, may be only one or two fields of information for each

input transaction. In our case this does include alphabetic information which always remains in coded decimal form. The net saving in computer time will be the amount of time saved by compressing information on tape and in memory less the conversion time for information entering the computer in coded decimal form or leaving the computer in the same form for printed or punched card or paper tape output.

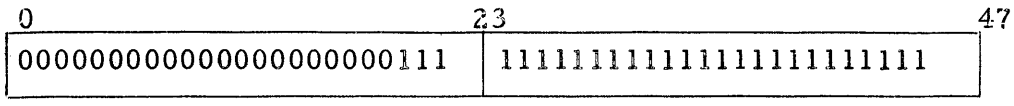
In low activity file maintenance operations where the amount of input and output conversion is relatively small compared to the total storage saved on long tape files, the net saving is quite obvious. In other types of activity the optimum method selected will depend upon the relationship of these two factors to each other.

IIa. Data Word

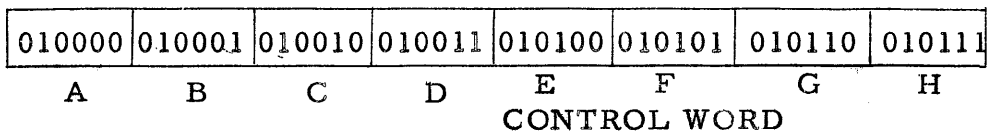


III As in data words, program constants can be represented in any form the program requires, remembering only that arithmetic operations require a pure binary representation.

III Program Constant



EXTRACT PATTERN

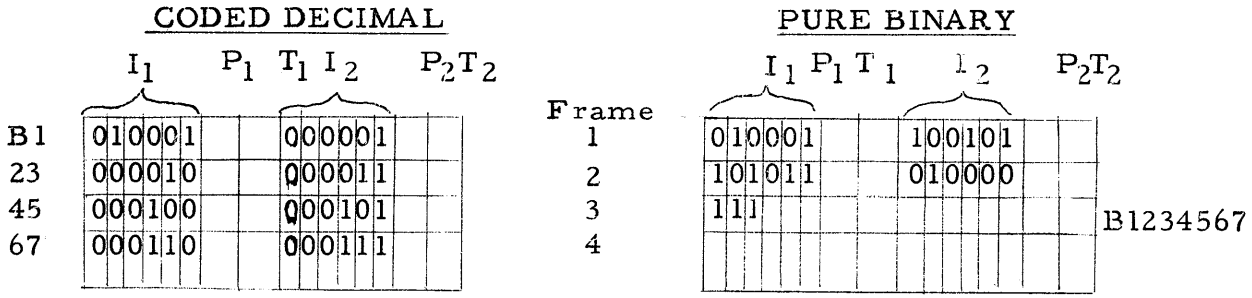


CONTROL WORD

MAGNETIC TAPE

ADVANTAGES OF PURE BINARY PACKING

Binary representation of information on Magnetic Tape allows a considerable saving in storage and a reduction in computer memory time for most computer applications. The use of a one inch wide tape in the TRANSAC S-2000 system increases the effective density of information stored on tape. Using binary representation rather than coded decimal increases the density even more. The character configuration "B1234567" represented on Magnetic Tape in coded decimal form requires four frames or 8 six-bit characters each with a parity bit. Using binary representation, the alphabetic character still requires 6 information channels but the numeric portion can be contained in slightly more than three character positions. This, of course, is only one field of information. A Tape file with many fields of information and thousands or millions of records is substantially reduced by packing information in pure binary form.



I_1 = First 6 Information Channels

P_1 = Parity Bit for first 6 Information Channels

T_1 = Main Timing or Sprocket Channel

I_2 = Second 6 Information Channels

P_2 = Parity Bit for Second 6 Information Channels

T_2 = Secondary Timing or Sprocket Channel

APPENDIX D

TRANSAC S-2000

PRODUCT LINE LIST AND UNIT DESIGNATION CODE

The unit designation code will consist of a group of two letters followed by a four digit number for all on-line units. The off-line conversion units will be designated by a code consisting of a group of three (3) letters followed by a four (4) digit number. The letters will designate the function of the unit such as, MF for Main Frame, IR for Index Registers, MC for Magnetic Core, etc. The most significant digit will be a 2 to identify the unit with the S-2000. The next 3 or 4 digits will give important parameters of the unit such as, the number of blocks of 4096 words in the case of the core memory, the number of index register, the number of columns in the case of punch card equipment, the number of lines per minute and columns in the case of the high speed printer.

On-Line Equipment List

MF-2000	Main Frame including arithmetic and control unit, operator console and power distribution.
MC-2001 to 16	Magnetic Core Memory unit. The last two digits indicate the number of blocks of 4096 words in the core memory.
IR-2004, 8,12,16	Index Registers Unit from 4 to 16 registers in steps of 4.
FP-2000	Floating Point Arithmetic Unit.
TT-2150	Magnetic Tape Transport Unit.
TC-2150	Magnetic Tape Control Unit for TT-2150
TS-2150	Magnetic Tape Sequencer for TT-2150 and TC-2150.

On-Line Equipment List

MD-2032	Magnetic Drum. Last two digits are two most significant digits in capacity of the drum expressed in number of words.
DC-2032	Magnetic Drum Controller for MD-2032.
PT-2007	On-line Paper Tape System - last digit indicates 7 level tape. Including: 1. Flexowriter Model FR 2. Ferranti Photo-Electric Reader 200 Characters/Second 3. Western Electric Hi-Speed Punch 60 Characters/Second 4. Console 5. Paper Tape Controller
PC-2208	On-line Punch Card System.
CT-2000	Console Typewriter including controller.

Off-Line Equipment List

CPC-2208	Off-Line Punch Card ⇔ Magnetic tape System. The 2nd, 3rd digits designate the read and punch unit by speed performance, and the last digit indicates 8 for 80 column cards and 9 for 90 column cards.
CPS-2912	Off-Line High Speed Printing System Including: 1. HP-2912 Printer rated at 900 lines/minute, 120 Column, 56 Characters 2. Controllers TC-2151 PC-2912 3. Buffer--1024 Characters OB-2000 4. Tape Unit--TT-2150

The 9 refers to the printing speed of 900 lines/minute; the 12 refers to the number of columns -- 120. The printing system can also be provided with speed of 600 lines per minute and 1200 lines per minute in the cases the code would read CPS-2612 and 2112 respectively. The unit is also available in 24, 120, or 160 column widths. In which case, the last two digits would be 24, 12 and 16 respectively. Variable speed drive can also be made available on the 120 and 160 column units. In which case, the code is 2012 or 2016.

PHILCO CORPORATION
GOVERNMENT & INDUSTRIAL DIVISION
PHILADELPHIA 44, PENNSYLVANIA