ERA

ANALYSIS
RESEARCH
DEVELOPMENT
ENGINEERING

Engineering Research Associates, INC.

ST. PAUL, MINNESOTA ARLINGTON, VA.

H. SNYDER

40

SUMMARY OF CHARACTERISTICS

MAGNETIC DRUM BINARY COMPUTER

Proposed for National Bureau of Standards

U.S. Department of Commerce

Contract CST-10133

30 November 1948

Pub. No. 23

ENGINEERING RESEARCH ASSOCIATES, INC.

1902 West Minnehaha Avenue, St. Paul 4, Minnesota

PREFACE

This report has been prepared for the National Bureau of Standards, U.S. Department of Commerce, as one phase of the work performed under Contract CST-10133. Its purpose is to summarize the characteristics of a digital computer employing magnetic drum data storage. This somewhat condensed summary is intended to serve as the engineering basis for negotiations leading to a possible contract for the construction of a computing machine, in accordance with agreements reached at a conference in St. Paul on 5 November 1948. The properties of the machine are defined in sufficient detail to be useful to anyone wishing to program sample problems for the purpose of evaluating the speed and general utility of the machine.

In compliance with the terms of the contract, the described characteristics are based on techniques which are either in active development or have been subjected to detailed critical analysis.

Preliminary reports outlining three variations on the design of a magnetic drum computer were submitted on 13 August, 30 September, and 19 October 1948. These are listed as References (a), (b), and (c) in Appendix A. The present machine is different from those previously described.

TABLE OF CONTENTS

CHAPTER 1 - FUNCTIONAL DESCRIPTION OF COMPUTER

CHAPTER 2 - COMPUTING CHARACTERISTICS

TABLE OF CONTENTS - Continued

LIST OF FIGURES

LIST OF TABLES

APPENDICES

CHAPTER 1

FUNCTIONAL DESCRIPTION OF THE COMPUTER

1.1 Introduction.

1.1.1 The computer. The device whose properties are the subject of this report is a general purpose electronic binary computer with magnetic drum internal data storage. It is a selective sequence machine which employs a "two-address" system of logic for following a program of instructions.

1.1.2 Parallel transmission and static retention of data. Figure 1 is a block diagram showing the paths over which digital data are transmitted between the principal units of the computer. (This is not a complete block diagram of the computer.) In the transmission of digital signals, the presence of a pulse in a channel represents a 1 and the absence of a pulse a 0. The digits of a multidigit number are transmitted simultaneously over a multichannel bus. All digital information "in process" within the computer (excluding that which is in magnetic storage) is held on registers made up of toggle-circuits, or static flip-flops.

A toggle-circuit is a vacuum tube circuit which has two stable states. It is therefore capable of retaining, or remembering, a single binary digit. In Figure 1, each toggle-circuit register is indicated schematically as a rectangle with two input terminals, labelled 0 and 1, and two output terminals, similarly labelled. The symbol which represents a 30-digit register may be thought of as a composite of 30 individual toggle-circuits, each one of which may be represented by a similar kind of symbol. If the 0 input terminal of a toggle is pulsed, the 0 output terminal becomes positive. If the 1 input terminal is pulsed, the 1 output terminal becomes positive.

The transmission of data between registers is accomplished by means of gating circuits. A gate is a vacuum tube circuit having two input terminals and a single output terminal. A pulse impressed on the first input terminal produces an output pulse only if the second input terminal is positive. The signal voltage on this second input generally originates at the output terminal of a toggle-circuit.

In order to transmit the number contained in one toggle-circuit register to a second register, the destination register must first be cleared to 0's. A bank of gates whose secondary inputs are tied to the 1 outputs of the first register are then actuated by pulsing their primary inputs. The outputs of these gates go to the 1 inputs of the second register. Pulses are transmitted only over those channels in which the corresponding originating toggle-circuit contains a 1. The result is to duplicate the contents of the first register in the second.

It will be apparent that the same result may be obtained by first clearing the second register to 1's, then gating the 0 output of the first register to the 0 input of the second. The <u>complement</u> of the number in the first register may be inserted in a second register either by first clearing to 0's and then transmitting the 0 output to the 1 input, or else by clearing to 1's and transmitting the 1 output to the 0 input. (The complement of a number is described in Section 1.3.)

In Figure 1, a multichannel bus is indicated by a bold line. A bank of 30 gates is represented by a rectangle labelled "30 G", etc. The control pulse lines which actuate these gates are not shown. Whether a given register is cleared to 0's or to 1's is indicated by the position of the control line marked "CL".

It should be emphasized that the act of transmitting a number from one register to another does not remove the number from the source, but merely duplicates it at the destination.

1.1.3 Five sections of the computer. For descriptive purposes, the computer may be divided into five sections: Storage, Arithmetic, Program Control, Input, Output. The functions of the five sections of the computer will be described in the following sections.

1.2 Storage Section.

1.2.1 Organization of the drum. The computer has a storage capacity of 4096 (or 2^{12}) 30-digit binary numbers. Binary digits are stored as magnetic marks on the surface of a continuously rotating cylindrical drum. The marks are arranged in parallel peripheral tracks, with a single magnetic head for reading and writing assigned to each track. The digits of a 30-digit number are entered simultaneously into their respective "cells" on 30 tracks. Each track contains 2048 (or 2^{11}) cells, so that 30 tracks store 2048 30-digit numbers. There are two such groups of tracks on the drum.

Each 30-digit number occupies a "box". The location of each box on the drum is specified by a 12-digit binary number called the "address". The leftmost binary digit specifies one of the two groups of tracks. The other 11 digits specify one of 2048 angular positions of the drum. One group of tracks contains the 2048 boxes whose addresses (expressed in octal, or radix 8, notation) lie between 0000 and 3777, inclusive; the second group, those lying between 4000 and 7777, inclusive. The two boxes at a given angular position have addresses which differ by the octal number 4000.

In addition to the 60 storage tracks, there are 11 tracks for locating purposes and two tracks for timing and control of the storage operations. The 11-track group contains 2048 permanently recorded 11-digit binary numbers, corresponding to the 2048 angular positions of the drum. These numbers appear in consecutive order.

1.2.2 Physical characteristics. The physical characteristics of the magnetic storage drum are listed in Table I. Background material relative to the magnetic storage techniques employed is contained in References (d) and (e). Non-return-to-zero signals, described in these references, are used in the locator tracks. Return-to-zero signals are employed in the storage tracks.

1.2.3 Functional description. The functions of the electrical parts of the storage section are: (1) locating, (2) writing, and (3) reading.

The problem of locating is to find the box on the magnetic drum corresponding to the address which has been set up in the Storage Address Register (SAR, Figure 1). The leftmost stage of SAR selects the proper track group. An 11-fold coincidence circuit compares the settings of the other 11 stages with the outputs of the amplifier circuits which read the 11 locator tracks. When the drum reaches the angular position corresponding to the address contained in SAR, the coincidence circuit generates a pulse. This pulse causes the 30 digits of a number simultaneously to be written into, or read from, the selected one of the two groups of tracks, at the correct angular position.

The writing system records on the drum digital information which has been set up in the Storage Insertion Register and the Storage Blocking Register (SIR and SBR, Figure 1). The number to be written into storage is received and held by SIR. In some operations of the computer, the writing of certain specified digits of the number in SIR must be suppressed. This digit-blocking information is contained in SBR.

Sixty writing circuits drive the 60 magnetic heads associated with the storage tracks. Each writing circuit contains two miniature thyratrons, each of which can discharge a simple network through a winding on the magnetic head. One thyratron is fired to write a 1, the other to write a 0. The thyratrons are triggered by the locator coincidence pulse gated through the proper group-selecting gate. One of the tubes (for suppressed digits, both tubes) is prevented from firing by application of a negative bias to its shield grid. This type of writing circuit is described in References (d) and (e).

The Storage Reading Amplifiers (Figure 1) amplify the signals read from the selected group of storage tracks and cause the number read from a selected box to be transmitted to a specified destination in the computer. There are 30 reading amplifiers, each having dual input stages. One set of inputs is associated with each of the two track groups. Which set is blocked and which is operable is determined by the leftmost digit in SAR.

These Storage Reading Amplifiers have three sets of output gates. The number being read may be directed to the X-Register, to the Program Control Registers (CTS, PAR, EAR, in Figure 1), or to the Print/Punch Register. The destination is selected by impressing the pulse from the locator coincidence circuit on the appropriate set of output gates. In the case of reading data into the Print/Punch Register, a lockout is provided against filling the register until such time as the information previously put into it has been consumed by the typewriter unit.

The address search preceding each writing or reading operation is initiated by a signal from the Program Control Section of the computer. Upon completion of the writing or reading operation, an end-signal notifies Program Control and clears SAR, SIR, and SBR.

1.3 Arithmetic Section.

1.3.1 Components. The principal units of the Arithmetic Section of the computer are the X-Register (X), the Q-Register (Q), and the Accumulator (A). All arithmetic operations are performed on numbers contained in these units. The simpler arithmetic operations are directed by the Program Control. Sequential routines (shift, multiply, divide) which are part of the more complex arithmetic processes are directed by the Arithmetic Sequence Control, of which the Arithmetic Shift Counter (ASK) is a component unit. The avenues of communication between these and other units of the computer are shown schematically in Figure 1.

1.3.2 Number representation. Binary numbers are used throughout the computer. Negative numbers are expressed as complements on $(2^{30}-1)$ in the boxes of storage, in X, and in Q. These units are all of 30-digit width. In A, which is 60 digits wide, negative numbers are expressed as complements on $(2^{60}-1)$. This is the familiar "one's-complement" representation. It is a property of this representation that the extreme left digit of a positive number is 0; that of a negative number, 1.

The binary point is located at the right in all registers. That is, all numbers are treated as integers by the machine. Non-integer numbers may be handled by programming scale factors.

To summarize, registers other than A may contain any positive or negative integer whose absolute value does not exceed $(2^{29}-1)$. The Accumulator (A) may contain any integer whose absolute value does not exceed $(2^{59}-1)$.

1.3.3 Accumulator. The Accumulator is the central theater of numerical operations in the computer. In this unit, the sum, difference, product, or remainder are formed, and numbers are manipulated in various ways.

The Accumulator is basically subtractive, with end-around borrow. That is, every number transmitted to A is automatically subtracted into (i.e., subtracted from) the number there, the new number in A being the resulting difference. The subtraction is performed modulo $(2^{60}-1)$. A number may be added into A by transmitting its complement to A.

There are two ways of representing zero in the one's-complement number system: (000...0) and (111...1). An additive accumulator can generate only the kind of zero (111...1) which has the machine properties of a negative number. A subtractive accumulator can generate only the kind of zero (000...0) which has the machine properties of a positive number. Apart from this exclusion of the negative zero, the question of whether the accummulator is subtractive or additive is an internal property of the machine, and need be of no concern to the programmer.

The number in A may be shifted to the left by any number of places from 1 to 59, in response to a single command. This shift is "circular"; i.e., a digit shifting out of the left end of A is not lost, but shifts into the right end. Each single shift is mathematically equivalent to multiplication by 2, modulo $(2^{60}-1)$.

The Accumulator Input Gates shown in Figure 1 perform the conversion of 30-digit numbers from X or Q to the appropriate kind of 60-digit number to be subtracted into the Accumulator. The kinds of numbers which are subtracted into A in the various arithmetic operations are tabulated in Table II, and will be discussed in Section 2.3.

1.3.4 Q-Register. The Q-Register participates in several arithmetic and logical operations. These will be described in Chapter 2. In particular, the quotient is formed in Q in division, and the multiplier held there during multiplication. This register may also be used for rapidly accessible storage of a single number, since it communicates bilaterally with A.

By means of a single command, the number in Q may be shifted circularly to the left from 1 to 29 places.

1.3.5 X-Register. The principal function of X is to receive numbers transmitted from storage to the Arithmetic Section. This register has no adding or shifting property. It is possible to subtract into A the number in X or its complement. Addend, subtrahend, multiplicand, or divisor are held in X during the corresponding arithmetic operations.

1.3.6 Shift Counter. The Arithmetic Shift Counter (ASK in Figure 1) counts the number of places shifted in the several operations involving shifts of Q or A. It is a subtractive counter which may be preset to the number of places it is desired to shift. Each single-shift control pulse subtracts 1 from the counter. The arrival of ASK at zero cuts off the train of control pulses and stops the shifting sequence.

The Arithmetic Shift Counter counts modulo 60, in that it resets to 59 after 0. This property is used in the scale factor shift, to be described in Chapter 2. The output gates to SIR are included solely for the scale factor shift.

1.4 Program Control Section.

1.4.1 Computer instructions. The function of the Program Control Section of the computer is to direct the execution of a program of instructions, contained in storage, for the working of a mathematical problem.

Each instruction is expressed as an aggregate of 30 binary digits and may be stored and transmitted in precisely the same manner as a numerical quantity. The right-hand 12 digits of an instruction represent y, the execution address. This is the address of the box in storage to which reference must be made in order to carry out the instruction. The next 12 digits to the left represent p, the program address. This is the address of the box containing the next instruction in the program. The left-hand 6 digits express the command code. This code specifies which command, of the 39 understood by the machine, is to be executed.

Some of the commands do not require a reference to storage for execution; for these there is no associated execution address, and the right-hand 12 digits are therefore irrelevant. In a few cases, these 12 digits contain specialized information; e.g., the number of places that Q or A is to be shifted. The commands will be discussed in Chapter 2.

1.4.2 Components. Program Control obtains each instruction in the program from storage and translates it into control pulses directed to various parts of the computer in correct sequence for its execution. An instruction transmitted from Storage to Program Control is split into three parts. The execution address is received in the Execution Address Register (EAR in Figure 1), the program address in the Program Address Register (PAR), and the command code in the Command Translator Switch (CTS).

The Command Translator Switch is a toggle-circuit controlled diode matrix, of the type described in Reference (f). A given 6-digit code set up in CTS energizes one of a number of output leads (39 used, 64 possible). This d-c level signal selects the set of basic operations appropriate to the specified command. These operations are sequenced by pulses from the pulse generation and distribution units and associated connecting matrices. (These units are shown in block form in Figure 3.)

1.4.3 End Point Counter. The End Point Counter (EPK in Figure 1) is a 12-digit subtractive counter whose primary purpose is to count the number of times a repetitive routine has been traversed in a program. The counter is operated by two special commands which will be discussed in Chapter 2. This unit also participates in the machine input operations.

1.5 Input Section.

1.5.1 Input medium. The Input Section of the machine transmits numerical data and the program of instructions from the input medium to the storage system. The input medium is 7-hole punched paper tape, with data confined to six levels, or tracks. The seventh level is reserved for codes to control the input operations.

Each 30-digit box in storage is loaded from five consecutive lines on the input tape. These five lines will be referred to as a "frame". The first line of a frame corresponds to the leftmost fifth of a box, the second line to the second fifth, etc.

1.5.2 Tape reader. The tape is scanned by a photoelectric reading device at a nominal speed of 75 feet per minute. This corresponds to 150 lines per second, or 30 frames per second. If it should be desired to load the entire drum, it would take about 2.3 minutes to fill all 4096 boxes.

The magnetic drum rotates continuously at its normal speed during the input operation. The tape feed need not be synchronized with the drum rotation. It is necessary only that the number of drum revolutions per second exceed the number of tape frames scanned per second.

The tape is driven by friction rollers, moving continuously rather than intermittently. The feed holes, normally used in sprocket-driven tape systems, are scanned photoelectrically to provide pulses for timing the input control circuits.

1.5.3 Input control, characteristics of. The information on the tape is of two kinds: (1) data (numerical and program) to be written into storage; and (2) codes for controlling the operation of the input section, but not to be written into storage.

The fifth or last line of each frame on the tape is accompanied by a seventh-level hole. This synchronizing hole is one form of input control code. In addition, there are two basic commands for controlling the tape-to-drum loading operation. Each sequence of data frames is preceded by a frame containing an initial address, plus a tape control code in the seventh level. Following the sequence of data frames there is a frame containing a check address, plus a different tape control code in the seventh level. The two tape control commands are logically equivalent to the following statements:

> INSERT ADDRESS: Write the contents of the next frame of tape into the box specified by this address, and continue to write the contents of succeeding frames of tape into boxes of consecutively ascending address.
>
> CHECK ADDRESS: Stop the loading sequence and compare the address given in this frame with the address (contained in a counter) of the box which was about to be loaded. If they are alike, continue scanning tape for next sequence. If not, stop tape motion and flash alarm. (This command stops the tape motion only if an error is detected.)

Each sequence of data is preceded by an INSERT ADDRESS code, and is followed by a CHECK ADDRESS code. Several sequences may be spliced together, or punched on one tape, or punched on different tapes. They may be loaded into the machine in any order.

1.5.4 Input control, mechanism of. The appearance of the tape control coding is shown in Figure 2. A bank of 11 phototubes scans the tape as it travels past the reading position. These are identified by the shaded blocks in the diagram. Six of the phototubes read a single 6-digit line of data. One phototube views the feed holes. Four phototubes read the seventh-level control code.

As each line of tape arrives at the horizontal bank of data phototubes, the four seventh-level tubes observe a binary code (e.g., 1101 at the instant shown) which identifies the procedure to be followed on that line. The five lines composing a frame are labelled a,b,c,d, and e. The positioning of the control phototubes is such that when the data phototubes are viewing the e line, the control phototubes are viewing the e', a,b, and e positions in the seventh level. The label e' refers to the e position in the preceding frame. The e' and e control phototubes are energized simultaneously only when an e line is opposite the data phototubes. That is, an e line is identified by a control code of the form 1XX1. In particular, 1101 identifies the e line of an INSERT ADDRESS frame, 1001 the e line of an ENTER DATA frame, and 1011 the e line of a CHECK ADDRESS frame. Control codes not of the form 1XX1 identify a,b,c, and d lines of <u>any</u> kind of frame; these are all given identical treatment. Instructions for the final disposal of a frame of data are not revealed until the e line is read.

Figure 3 is a block diagram showing those parts of the computer which participate in input operations. The digital transmission paths used in input operations are shown also in Figure 1.

The Tape Translator Switch (TTS) is a considerably simplified version of the Command Translator Switch (CTS). One of its five output leads is energized on receipt of the 4-digit tape control code. The outputs operate into the computer's operation selecting matrix, in the same manner as the outputs of CTS. All input operations internal to the computer take place at the same pulse rates as in regular computing.

The Q-Register is used for assembling the five 6-digit lines of a frame of tape, and for providing part of the "cushion" storage of data en route from tape to drum. The rapid shifting property of Q is utilized in the assembly operation.

After the e line of a frame has been received by Q, the number in Q is transmitted to one of two destinations, depending on the kind of frame being read, as identified by TTS. If the frame is of the ENTER DATA kind, the number in Q is transmitted to SIR for writing into storage. If it is either an INSERT ADDRESS or a CHECK ADDRESS frame, the complement of the number is transmitted to EPK (right-hand 12 digits only, others being irrelevant).

The End Point Counter (EPK) is utilized to keep track of the address of the box to be loaded. For machine loading operations, this unit is provided with an "AWC" set of input terminals. A 12-digit number transmitted to EPK via these terminals is "added without carry" to the number in EPK.

In Section 1.4.3, EPK was described as a 12-digit subtractive counter. For input operations, it is required to add 1 to the address in the counter for each successive frame of data. The desired properties are realized by inserting the complement of the initial address in EPK, subtracting 1 from this each time, and transmitting the complement of the number in EPK to SAR. (This is equivalent to relabelling EPK's input and output terminals).

The modus operandi of the input process will now be described (with the lesser details omitted). As each line of tape passes before the phototubes, a sharp pulse is delivered by the Feed Pulse Amplifier. This pulse causes TTS and the right-hand six stages of Q to be filled from the appropriate tape amplifier outputs, and initiates a short, rapid sequence of internal machine operations, as follows:

If the TTS setting is not of the form 1XX1, the line is a,b,c, or d, and the principal operation is a 6-place shift of Q.

If the TTS setting is 1101, the line is the e line of an INSERT ADDRESS frame, and Q is now full. First EPK is cleared to 0's, then the complement output of Q is transmitted to the AWC input of EPK. The number in EPK is now the complement of the initial address.

If the TTS setting is 1001, the line is the e line of an ENTER DATA frame, and Q is full. The complement output of EPK is transmitted to SAR. This sets up the initial address in its normal form. The normal output of Q is transmitted to SIR. The Initiate Write control line to storage is pulsed. The Advance EPK line, which subtracts 1, is pulsed. This sets up the destination address for the next frame.

If the TTS setting is 1011, the line is the e line of a CHECK ADDRESS frame, and Q is full. If the check is good, the numbers in Q and EPK are complements. The complement output of Q is transmitted to the AWC input of EPK, as before. This leaves 0's in EPK. The Advance EPK line is pulsed. If this operation produces an end-borrow, the check is good.

So that orders of magnitude of the time intervals involved may be fully appreciated, it should be noted that the feed pulse repetition period is extremely long (6700 microseconds) relative to the internal clock pulse period (2.5 microseconds).

1.5.5 Tape preparation unit. The input tape is prepared by means of a special keyboard unit which actuates a standard 7-hole tape punch. The keyboard arrangement has not been standardized at this writing. One suggested design has two arrays of 64 keys each. The keys in one group are marked with the octal numbers 00 through 77. Thirty-nine of the keys in the other group are labelled with the 39 command code symbols (OA, OP, TE, etc.). Each key in one group is tied electrically to one key in the other group. Depressing a given key sets up the corresponding 6-digit binary number to be punched as a line of data. After the number is set up, a Punch and Advance bar is struck. There are two such bars. One of these causes the seventh-level hole to be included; the other causes it to be omitted. If a special switch is set, a seventh-level hole will automatically be included opposite every fifth, or e, line. Five lamps indicate which of the five lines of a frame is about to be punched.

1.6 Output Section.

The Output Section of the computer contains means for transmitting information from storage to a punched paper tape and/or a printed page. Two commands, PRINT ONLY and PRINT AND PUNCH, are provided for governing the output operations. These will be described in Chapter 2.

Both of these commands cause the right-hand six binary digits in a specified box in storage to be transmitted to the Print/Punch Register (PPR in Figure 1), as explained in Section 1.2.3. An electric typewriter then prints the character to which this code corresponds, and (if so instructed) a perforator punches the six digits as a line of data on 6- or 7-hole tape. As soon as the mechanical operation is under way, PPR clears and is receptive to further data.

Once a print operation has been initiated, the computer is free to continue with its program. It is not necessary for Program Control to stop and await completion of the printing operation. However, a second printing operation is automatically delayed until completion of the previous one.

CHAPTER 2

COMPUTING CHARACTERISTICS

2.1 Introduction.

The computer has a repertoire of 39 operation commands. These commands are defined in Section 2.2, without elaboration. Comments on the commands are contained in Section 2.3. Factors influencing the speed of computing are treated in Section 2.4.

2.2 Definitions of the Commands.

2.2.1 Nomenclature. In the 39 definitions to follow, parentheses around an address symbol or a register symbol mean "the number contained in" the box or register so designated; e.g., (y) is the number in the box whose address is y, and (A) is the number in the Accumulator.

The numbers contained in A, Q, and X are expressed as aggregates of digits a_i, q_i, and x_i, respectively, where the subscript "i" is the power of 2 associated with that digit. That is, a_0 is the coefficient of 2^0, and is therefore the right-hand digit of (A); a_i is the $(i + 1)$th digit from the right.

A coding designation is included for each command. (These codes are repeated in Table III, together with their mnemonic significance.) Each double capital letter should be regarded as a single symbol. On the tape-preparing keyboard there will be a key for each of the 39 symbols.

This set of commands will be called List E-4, to distinguish it from previously reported lists.

2.2.2 Principal additive commands. The following twelve statements define the "ordinary", "absolute", and "split" types of additive command:

1. OApy - HOLD ADD: Add (y) to (A).

2. OPpy - CLEAR ADD: Clear A and insert (y).

3. OSpy - HOLD SUBTRACT: Subtract (y) from (A).

4. ONpy - CLEAR SUBTRACT: Clear A and insert the negative of (y).

5. AApy - ABSOLUTE HOLD ADD: Add to (A) the absolute value of (y).

6. APpy - ABSOLUTE CLEAR ADD: Clear A and insert the absolute value of (y).

7. ASpy - ABSOLUTE HOLD SUBTRACT: Subtract from (A) the absolute value of (y).

8. ANpy - ABSOLUTE CLEAR SUBTRACT: Clear A and insert the negative of the absolute value of (y).

9. SApy - SPLIT HOLD ADD: Add (y) to (A), except that 0's are to be added into the left-hand 30 places of A.

10. SPpy - SPLIT CLEAR ADD: Clear A, insert (y) into the right-hand 30 places.

11. SSpy - SPLIT HOLD SUBTRACT: Subtract (y) from (A), except that 0's are to be subtracted into the left-hand 30 places of A.

12. SNpy - SPLIT CLEAR SUBTRACT: Clear A, insert the negative of (y) into the right-hand 30 places, and insert 1's into the left-hand 30 places.

<u>2.2.3 Miscellaneous Accumulator commands.</u> The following seven statements define additional commands relating to transmission of data between A and storage:

13. NPpy - CLEAR ADD PLUS ONE: Clear A and insert (y) + 1.

14. NMpy - CLEAR ADD MINUS ONE: Clear A and insert (y) - 1.

15. LApy - HOLD LOGICAL MULTIPLY: "Split add" (y) to (A), as in SApy, but suppress the transmission to A of those digits of (y) which are in

places corresponding to 0's in (Q).

16. LPpy - CLEAR LOGICAL MULTIPLY: Clear A and insert (y) into the right-hand 30 places, but suppress the transmission to A of those digits of (y) which are in places corresponding to 0's in (Q).

17. SDpy - SUBSTITUTE DIGITS: Replace each digit of (y) with the corresponding digit of (A), provided the corresponding digit of (Q) is a 1; the remaining digits of (y) are not to be disturbed.

18. SEpy - SUBSTITUTE EXECUTION ADDRESS: Replace the right-hand 12 digits of (y) with the corresponding digits of (A); the remaining 18 digits of (y) are not to be disturbed.

19. AYpy - STORE A: Replace (y) with the right-hand 30 digits of (A).

2.2.4 Q-Register commands. The following five commands are related to transmission of data into and out of Q:

20. YQpy - FILL Q: Clear Q and insert (y).

21. AQp- - TRANSMIT A to Q: Clear Q and insert the right-hand 30 digits of (A).

22. QAp- - HOLD ADD FROM Q: Add (Q) to (A).

23. QPp- - CLEAR ADD FROM Q: Clear A and insert (Q).

24. QYpy - STORE Q: Replace (y) with (Q).

2.2.5 Arithmetic sequence commands. The following six commands are related to arithmetic processes which involve the Arithmetic Sequence Control:

25. ALpk - SHIFT A LEFT: Shift (A) to the left k times, replacing extreme right digit with the one which was at extreme left each time (circular shift).

26. QLpk - SHIFT Q LEFT: Shift (Q) to the left k times, replacing extreme

right digit with the one which was at extreme left each time (circular shift).

27. MApy - HOLD MULTIPLY: Add to (A) the product of (Q) and (y), without roundoff, leaving multiplier intact in Q.

28. MPpy - CLEAR MULTIPLY: Clear A and insert the product of (Q) and (y), without roundoff, leaving multiplier intact in Q.

29. DPpy - DIVIDE: Divide (A) by (y), putting the quotient in Q and leaving a non-negative remainder, R, in A. The quotient and remainder are defined by:

$$N = QD + R \qquad (0 \leqslant R < |D|),$$

where N = dividend (numerator)

and D = divisor (denominator).

30. SFpy - SCALE FACTOR SHIFT: Shift (A) circularly to the left until a_{59} and a_{58} become different; replace the right-hand 12 digits of (y) by a characteristic, k, defined by:

$$k \equiv (30 - S) \bmod 60 \qquad (0 \leqslant k \leqslant 59),$$

where S is the number of places shifted. If (A) consists of all 0's or all 1's, record k as 31.

<u>2.2.6 Test commands.</u> The following five statements define four test, or discrimination, commands (PEpt is simply a preparatory operation for TEpy):

31. PEpt - PRESET END POINT COUNTER: Clear EPK, insert the number t (the total number of times a routine is to be traversed), and subtract 1 from (EPK).

32. TEpy - TEST END POINT: Subtract 1 from (EPK); if an end-borrow (change of sign) results, take (y) as the next instruction; if not, take (p)

X 29008

as the next instruction.

33. TFpy - TEST FULL ACCUMULATOR: If a_{59} and a_{58} (coefficients of 2^{59} and 2^{58} in (A)) are alike, take (p) as the next instruction; if different, take (y) as the next instruction.

34. THpy - TEST HALF OVERFLOW: If a_{30} and a_{29} are alike, take (p) as the next instruction; if different, take (y) as the next instruction.

35. TSpy - TEST SIGN: If (A) is negative, take (y) as the next instruction; if (A) is positive or zero, take (p) as the next instruction.

2.2.7 Print and stop commands. The following four commands are related to output and stop operations:

36. POpy - PRINT ONLY: Transmit to 6-digit Print/Punch Register the right-hand 6 digits of (y), and cause electric typewriter to print the character to which this code corresponds.

37. PPpy - PRINT AND PUNCH: Same as POpy, but in addition cause perforator to punch the 6 digits as a line of data on 6- or 7-hole paper tape.

38. ISp- - INTERMEDIATE STOP: Stop clock pulse generator and flash INTERMEDIATE STOP signal.

39. FS-- - FINAL STOP: Stop clock pulse generator and flash FINAL STOP signal.

2.3 Comments on the Commands.

2.3.1 Numbers transmitted to Accumulator. The numbers subtracted into A in the several basic internal arithmetic operations are recapitulated in Table II. The symbols are those defined in Section 2.2.1. The complement of a digit is represented by a primed symbol; i.e., x_i' means $(1 - x_i)$.

2.3.2 Principal additive commands. The four ordinary additive commands and the four absolute additive commands have obvious purpose. The four split additive commands are provided for manipulating "multiple precision" numbers, where 30-digit portions of multiple length numbers are stored in separate boxes.

2.3.3 Miscellaneous Accumulator commands. The commands CLEAR ADD PLUS ONE and CLEAR ADD MINUS ONE reduce the number of storage references required to change a number (e.g., a computational index or a stored instruction) by one unit.

The LOGICAL MULTIPLY and SUBSTITUTE DIGITS commands provide considerable flexibility. To illustrate one type of application, five 4-digit numbers and one 10-digit number could be stored together in each box, and yet be manipulated as though they were stored in separate boxes. The selective operator in Q may be left set up for a long series of steps.

2.3.4 Q-Register commands. The commands TRANSMIT A TO Q, HOLD ADD FROM Q, and CLEAR ADD FROM Q enable Q to be used for rapidly accessible storage of a 30-digit number.

2.3.5 Arithmetic sequence commands. A natural question relative to the two SHIFT LEFT commands is: What happens if k, the number of places to be shifted, is made greater than 59 or 29 in ALpk or QLpk, respectively? The answer can be deduced from the properties of ASK, as described in Section 1.3.6. The actual number of places shifted is equal to the number represented by the right-hand six digits of k.

A flow diagram of the algorithm followed by the Arithmetic Sequence Control in executing the two MULTIPLY commands is given in Figure 4. (The switch-like symbols in this diagram indicate logical dichotomies, not actual switches.) The steps immediately preceding and following the "basic multiply algorithm" are corrections for a negative multiplier. The step labelled "Part I" is required only for cumulative multiplication.

A flow diagram of the DIVIDE algorithm is given in Figure 5. The "basic divide algorithm", if performed alone, would result in a quotient and a remainder satisfying the following conditions: (1) quotient odd (i.e., $q_{29} \neq q_0$); and (2) absolute value of remainder less than <u>or equal to</u> that of divisor. The initial and final corrective steps produce the positive remainder specified in the definition. In particular, a zero remainder is then represented as zero.

Prior to division, the dividend in A may be shifted to the left to provide the desired number of significant figures in the quotient. A limitation on this preliminary shift is that the quotient must lie within the range of Q; i.e., its absolute value must not exceed $(2^{29} - 1)$. In particular, if the most significant digit of the divisor is x_{28}, then the most significant digit of the dividend should lie no further to the left than a_{55} (this is a sufficient, but not always necessary, condition).

The command SCALE FACTOR SHIFT is a "substitution" command, which makes it possible to insert the characteristic, k, into a stored ALpk or QLpk instruction. The definition of k is such that if the corresponding "mantissa" is read out of storage and then shifted by ALpk, it will have been restored to its original position with respect to the binary point.

<u>2.3.6 Test commands.</u> The commands PRESET END POINT COUNTER and TEST END POINT are provided for counting the number of times a repetitive sequence of instructions in the program has been traversed. It is initially filled with a number, t, equal to the desired number of traversals, by means of the instruction PEpt. The instruction TEpy, following each traversal, diminishes the count by

one and performs the test. The two commands are defined so that the test goes in the p direction (t - 1) times, and goes in the y direction the t - th time.

The command TEST FULL ACCUMULATOR tests whether the most significant digit of (A) is at a_{58}. The command TEST HALF OVERFLOW may be used for determining whether a single additive operation has produced an overflow into the left half of A.

2.3.7 Print and stop commands. The commands PRINT ONLY and PRINT AND PUNCH are both provided in order that intermediate check results and other control information may be printed by the typewriter without molesting "smooth copy" being accumulated on the punched tape.

Two STOP commands are provided so that the operator may determine whether the machine has stopped to permit inspection of a printed intermediate result, or whether the problem is completely finished. In addition to these programmed stops, there are several alarm stops which are actuated by fault-detector circuits.

2.4 Operating Time Considerations.

2.4.1 Programming for maximum effective computer speed. In order to make the most effective use of the two-address system of commands, the programmer must have knowledge of the minimum allowable time intervals between various kinds of reference to storage. This information is given in Table III for the 39 computer commands.

Let p' be defined as the address of the box containing the present instruction; y, the execution address which is part of the present instruction; and p, the program address in the present instruction (i.e., the address of the box containing the next instruction). Then the meaning of the tabulated quantities, C, is that y should lie at least $C_{p'y}$ cells beyond p'; p should lie at least C_{yp} cells beyond y; and (for commands with no execution address) p should lie at least $C_{p'p}$

cells beyond p'. These numbers are given in decimal notation and are rounded to multiples of 8. Expressed in octal notation, these multiples are 10, 20, 30, etc.

The minimum intervals defined by the C's apply only to the angular portion of each address (see Section 1.2.1). Suppose that p' is 6123 (in octal) and that the instruction is MApy. Then y, the address of the box containing the multiplicand, should be no smaller than 6123 plus 10 (octal), or 6133. But box number 2133, which is situated at the same angular position in the other track group, passes the magnetic heads simultaneously with 6133, and therefore has equal status for timing purposes.

Should a shorter interval be programmed than that specified in the table, the only effect is a loss of operating time. The interval is automatically lengthened by the duration of a complete drum revolution, or 16 milliseconds.

As a further refinement, the programmer may assume that the dead time from 3777 to 0000, and from 7777 to 4000, is equivalent to 40 cell-periods.

2.4.2 Storage lockout delays. One of the factors taken into account in making up Table III is the circuit recovery time which must elapse between the several kinds of magnetic drum storage reference. Automatic lockout delays are built into the storage system controls for this purpose. These delays prevent a writing operation from following a previous writing operation by less than about 2000 microseconds, a reading from following a previous writing by less than about 250 microseconds, and either operation from following a reading by less than about 60 microseconds.

2.4.3 Clock pulse rate. The Clock Pulse Generator, which times the main Program Control and the Arithmetic Sequence Control, operates at a rate of 400,000 pulses per second. This rate determines (and is equal to) the shifting rate of Q and A, in places per second.

The basic addition cycle in the Accumulator is 3 clock pulse periods, or 7.5 microseconds. That is, 7.5 microseconds after transmission of a number to A, a second operation in A, such as shifting, may be ordered. This interval allows sufficient time for propagation of a 60-place borrow (extreme case) in A.

The timing of the MULTIPLY and DIVIDE processes, as directed by the Arithmetic Sequence Control, is based on these shifting and adding rates.

2.4.4 Parallel internal operations. The internal operations ordered by the Program Control occur both in sequence and in parallel. Execution of the instruction MApy will serve to illustrate. First, Part I of the multiply algorithm (Figure 4) is initiated, simultaneously with initiation of the search in storage for the multiplicand. Upon completion of these operations, Part II of the algorithm is initiated, simultaneously with initiation of the search in storage for the next instruction in the program.

CHAPTER 3

MISCELLANEOUS GENERAL CONSIDERATIONS

3.1 Manual Controls.

A number of manual controls are required for starting and for servicing the computer. An initial starting control sets the Program Control registers to read the first instruction from a certain box, say 0000, which is always used to hold the first instruction in the program. The starting control also performs such routine tasks as clearing toggle-circuit registers to their proper initial state.

Controls are also provided for running the Clock Pulse Generator at a reduced rate, and for single-stepping. The distribution of clock pulses may be interrupted and resumed manually.

3.2 Checking and Servicing.

Elaborate means for checking the operation of the computer are not provided. The design of the machine permits the frequent running of brief check problems during the course of a long computation. The non-volatile nature of the data in storage enables the retention of check problems in storage for extended periods. There are, however, certain specialized checks, such as the address check in the machine loading operation (Section 1.5.4).

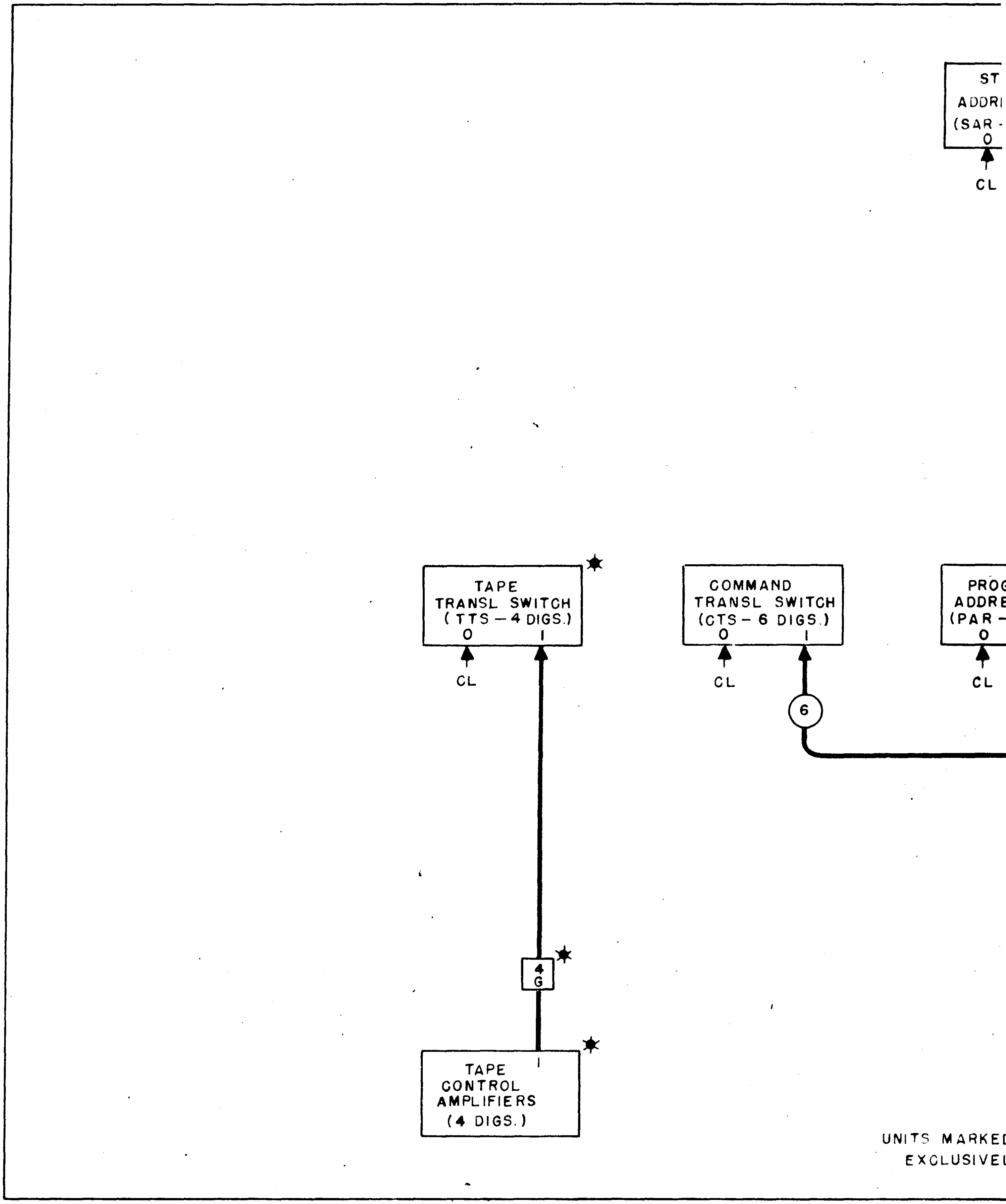
To facilitate servicing, neon lamps are provided for indicating the state of all toggle-circuits in the computer. It is therefore possible to single-pulse the machine and observe the action, for example, of the arithmetic units.

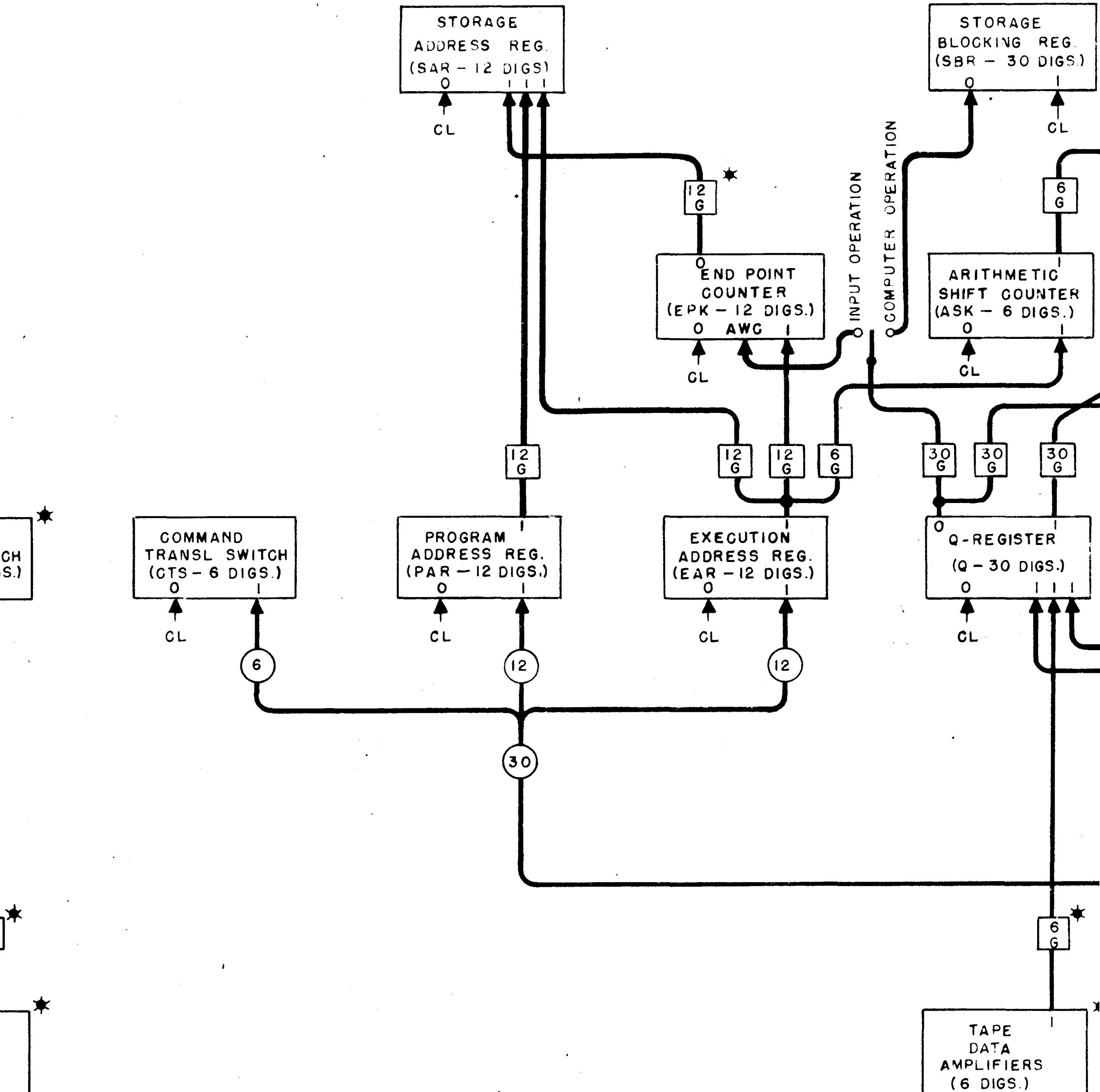
All electronic units are divided into replaceable plug-in chassis of convenient size.

3.3 Number of Tubes.

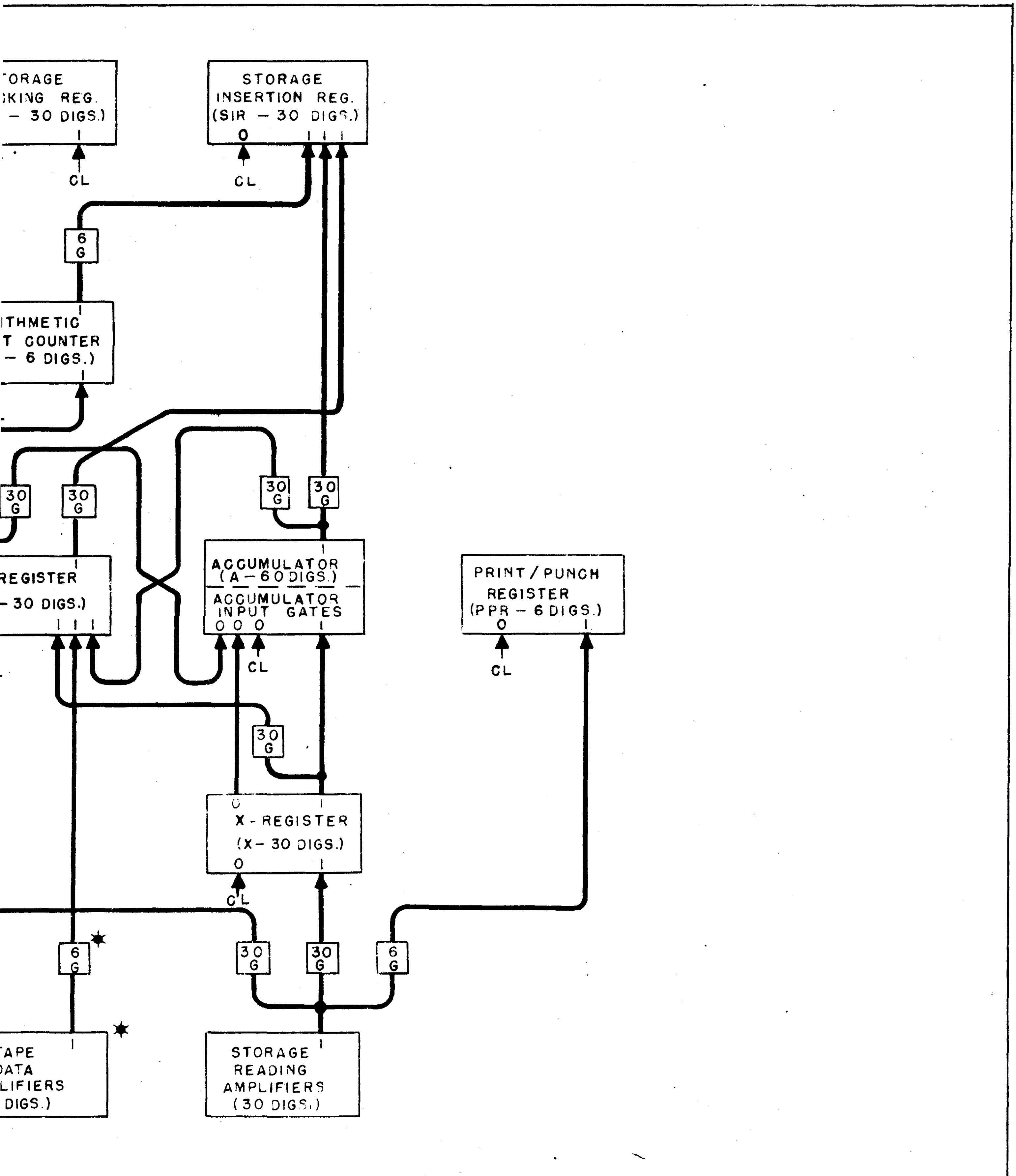
The number of tube envelopes in the entire computer is estimated at 2500.

Figure 1. DIGITAL TRANSMISSION PATHS


ST
ADDR
(SAR -
0
CL
TAPE
TRANSL SWITCH
(TTS — 4 DIGS.)
0
1
CL
COMMAND
TRANSL SWITCH
(CTS – 6 DIGS.)
0
1
CL
6
PROG
ADDRE
(PAR –
0
CL
4
G
1
TAPE
CONTROL
AMPLIFIERS
(4 DIGS.)
UNITS MARKED
EXCLUSIVEL



UNITS MARKED (✱) ARE USED EXCLUSIVELY FOR INPUT


ORAGE
KING REG.
– 30 DIGS.)
CL
STORAGE
INSERTION REG.
(SIR – 30 DIGS.)
CL
6 G
ITHMETIC
T COUNTER
– 6 DIGS.)
30 G
30 G
30 G
30 G
REGISTER
– 30 DIGS.)
ACCUMULATOR
(A – 60 DIGS.)
ACCUMULATOR
INPUT GATES
CL
PRINT / PUNCH
REGISTER
(PPR – 6 DIGS.)
CL
30 G
X - REGISTER
(X – 30 DIGS.)
CL
6 G
30 G
30 G
6 G
TAPE
DATA
LIFIERS
DIGS.)
STORAGE
READING
AMPLIFIERS
(30 DIGS.)

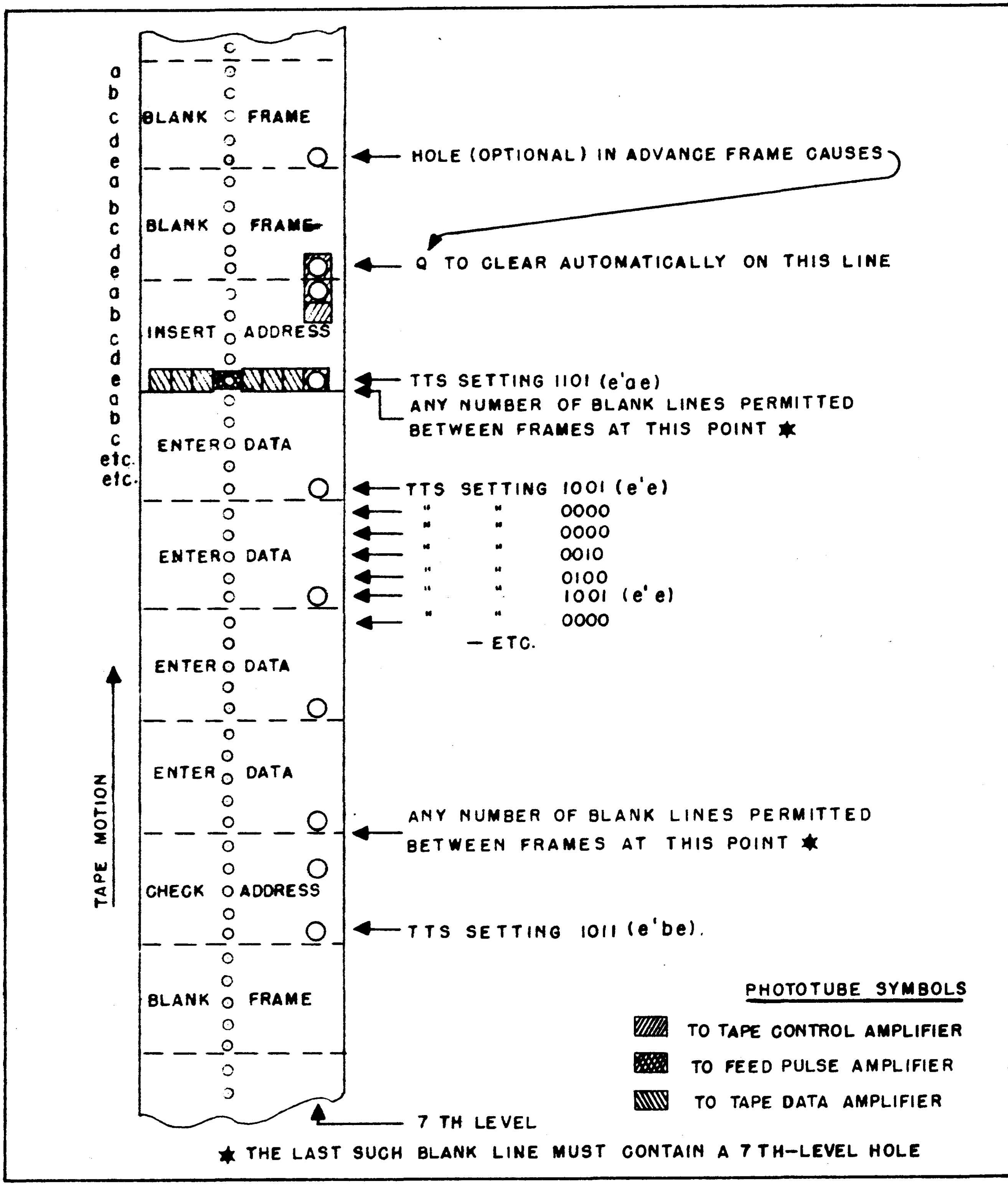
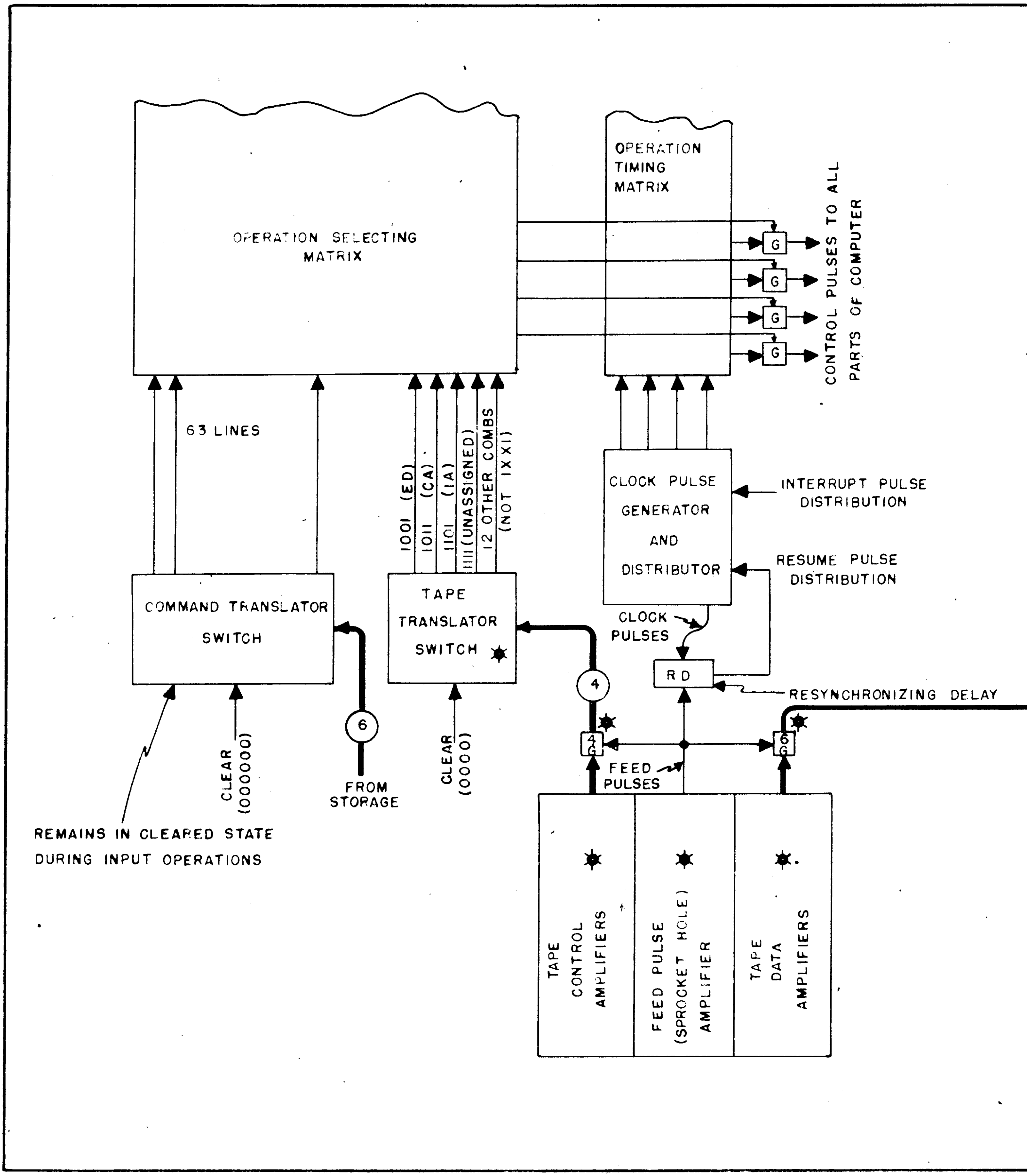


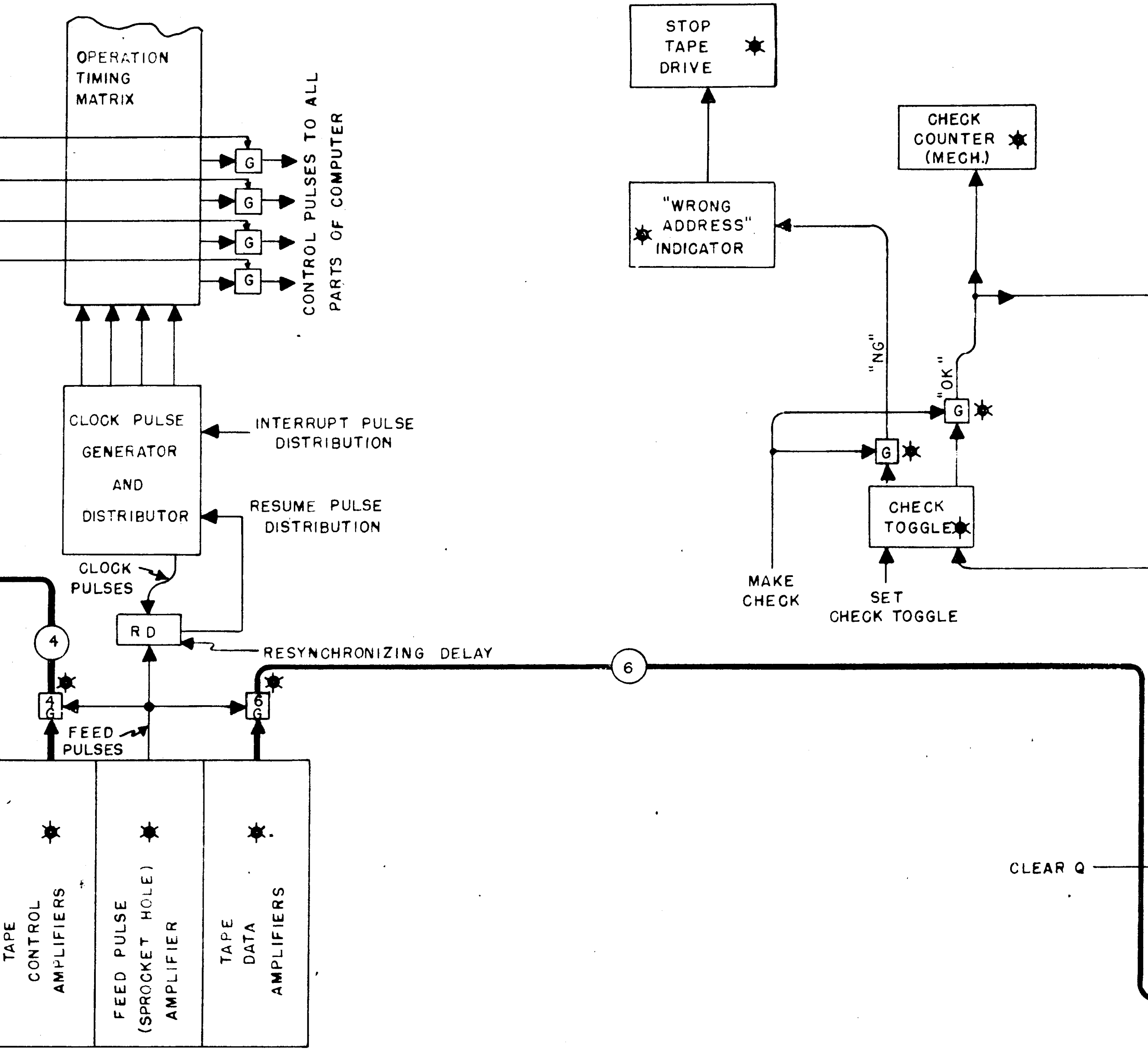
Figure 2. TAPE CONTROL CODING

A-28440

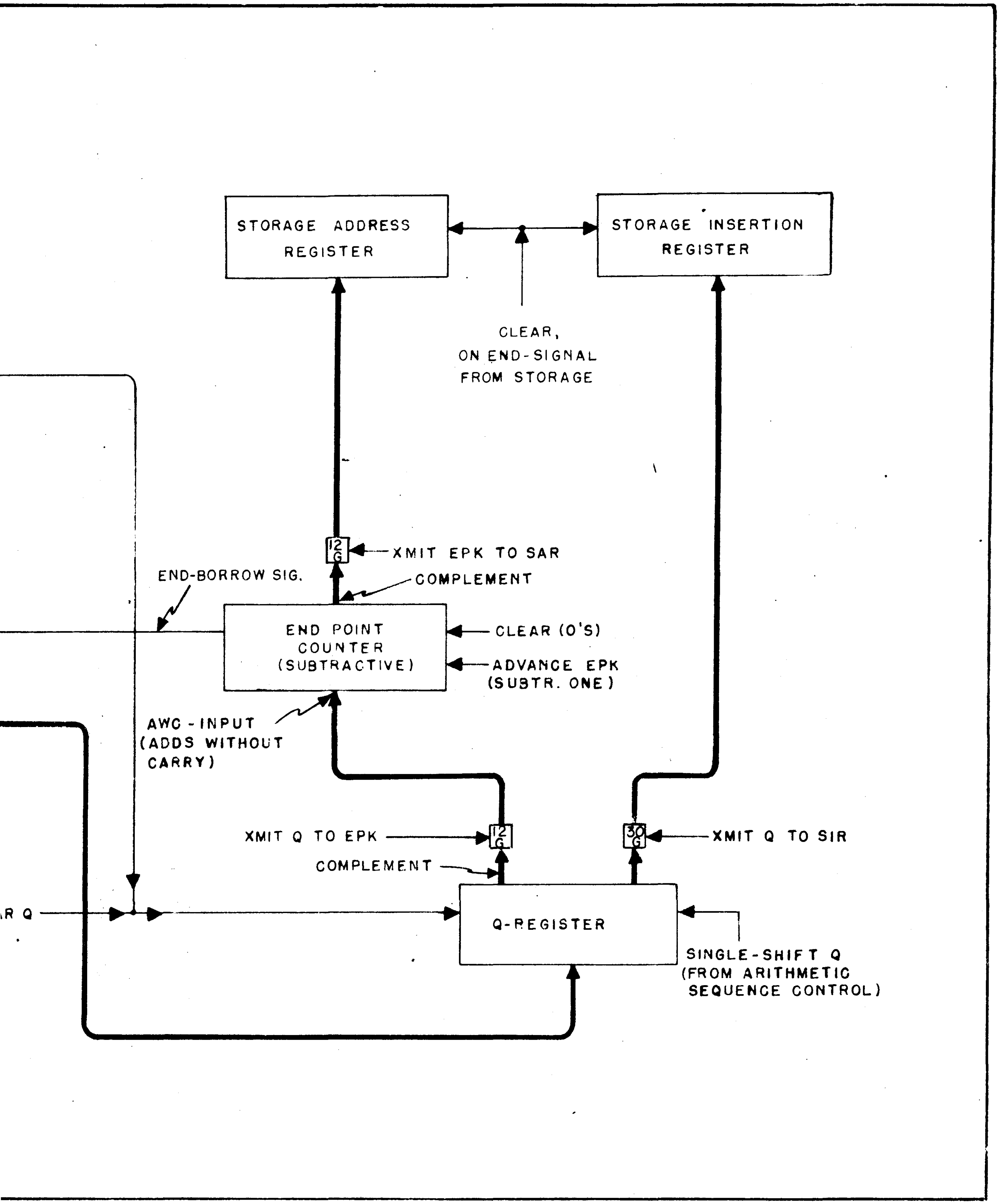
Figure 3. BLOCK DIAGRAM OF INPUT OPERATIONS

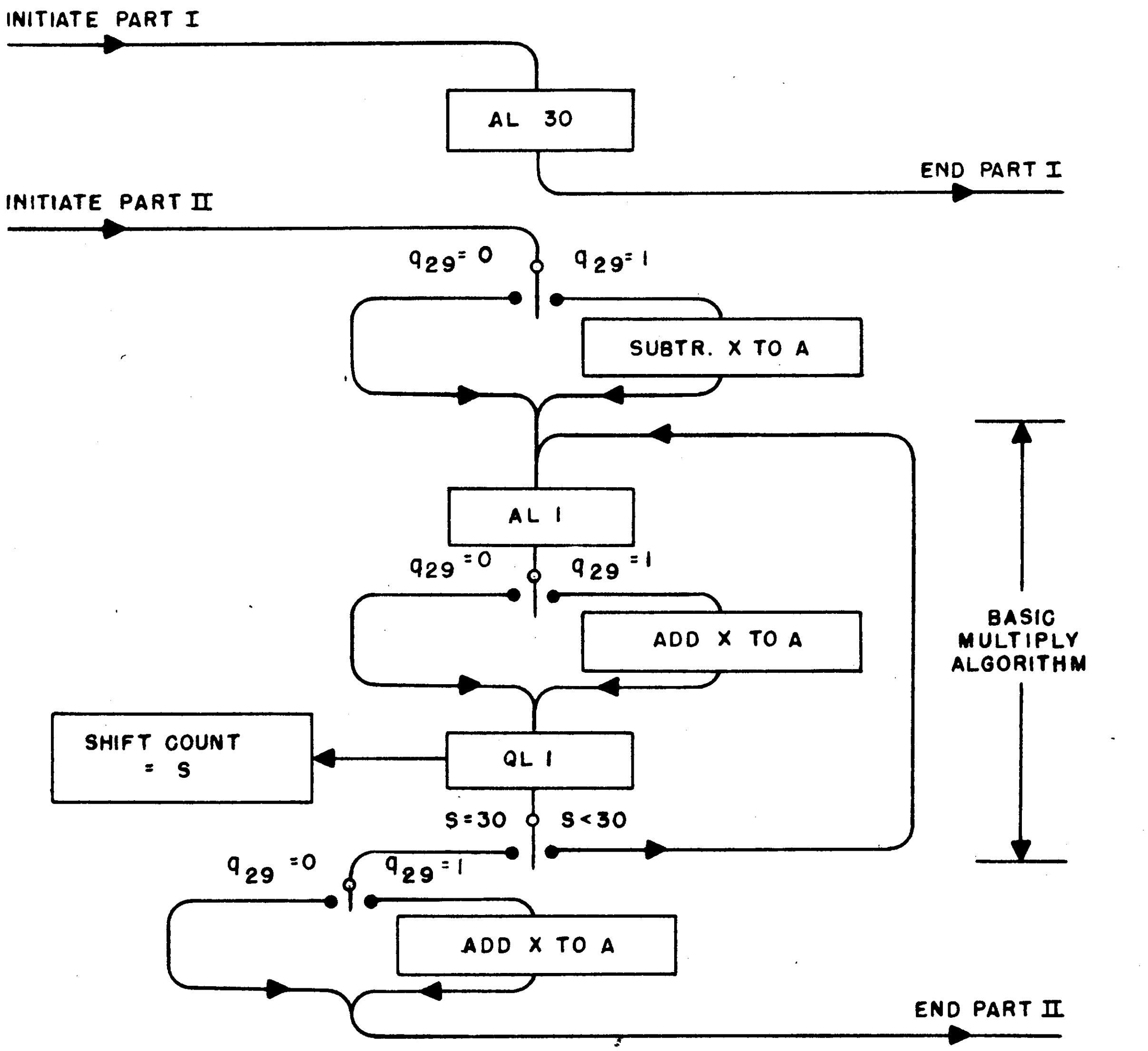


OPERATION SELECTING MATRIX
OPERATION TIMING MATRIX
G
CONTROL PULSES TO ALL PARTS OF COMPUTER
63 LINES
1001 (ED)
1011 (CA)
1101 (IA)
1111 (UNASSIGNED)
12 OTHER COMBS (NOT 1XX1)
CLOCK PULSE GENERATOR AND DISTRIBUTOR
INTERRUPT PULSE DISTRIBUTION
RESUME PULSE DISTRIBUTION
COMMAND TRANSLATOR SWITCH
TAPE TRANSLATOR SWITCH
CLOCK PULSES
RD
RESYNCHRONIZING DELAY
4
6
4 G
6 G
FEED PULSES
CLEAR (000000)
FROM STORAGE
CLEAR (0000)
REMAINS IN CLEARED STATE DURING INPUT OPERATIONS
TAPE CONTROL AMPLIFIERS
FEED PULSE (SPROCKET HOLE) AMPLIFIER
TAPE DATA AMPLIFIERS



UNITS MARKED (✱) ARE USED EXCLUSIVELY FOR INPUT


STORAGE ADDRESS REGISTER
STORAGE INSERTION REGISTER
CLEAR, ON END-SIGNAL FROM STORAGE
12 G
XMIT EPK TO SAR
END-BORROW SIG.
COMPLEMENT
END POINT COUNTER (SUBTRACTIVE)
CLEAR (O'S)
ADVANCE EPK (SUBTR. ONE)
AWC - INPUT (ADDS WITHOUT CARRY)
XMIT Q TO EPK
12 G
30 G
XMIT Q TO SIR
COMPLEMENT
R Q
Q-REGISTER
SINGLE-SHIFT Q (FROM ARITHMETIC SEQUENCE CONTROL)



MApy USES PARTS I AND II.
MPpy USES PART II ONLY.

INITIAL CONTENTS:

X: MULTIPLICAND
Q: MULTIPLIER
A: INITIAL NUMBER (ZERO FOR MPpy)

FINAL CONTENTS:

X: MULTIPLICAND
Q: MULTIPLIER
A: INITIAL NUMBER PLUS PRODUCT

Figure 4. MULTIPLY ALGORITHM (portion of MApy and MPpy)

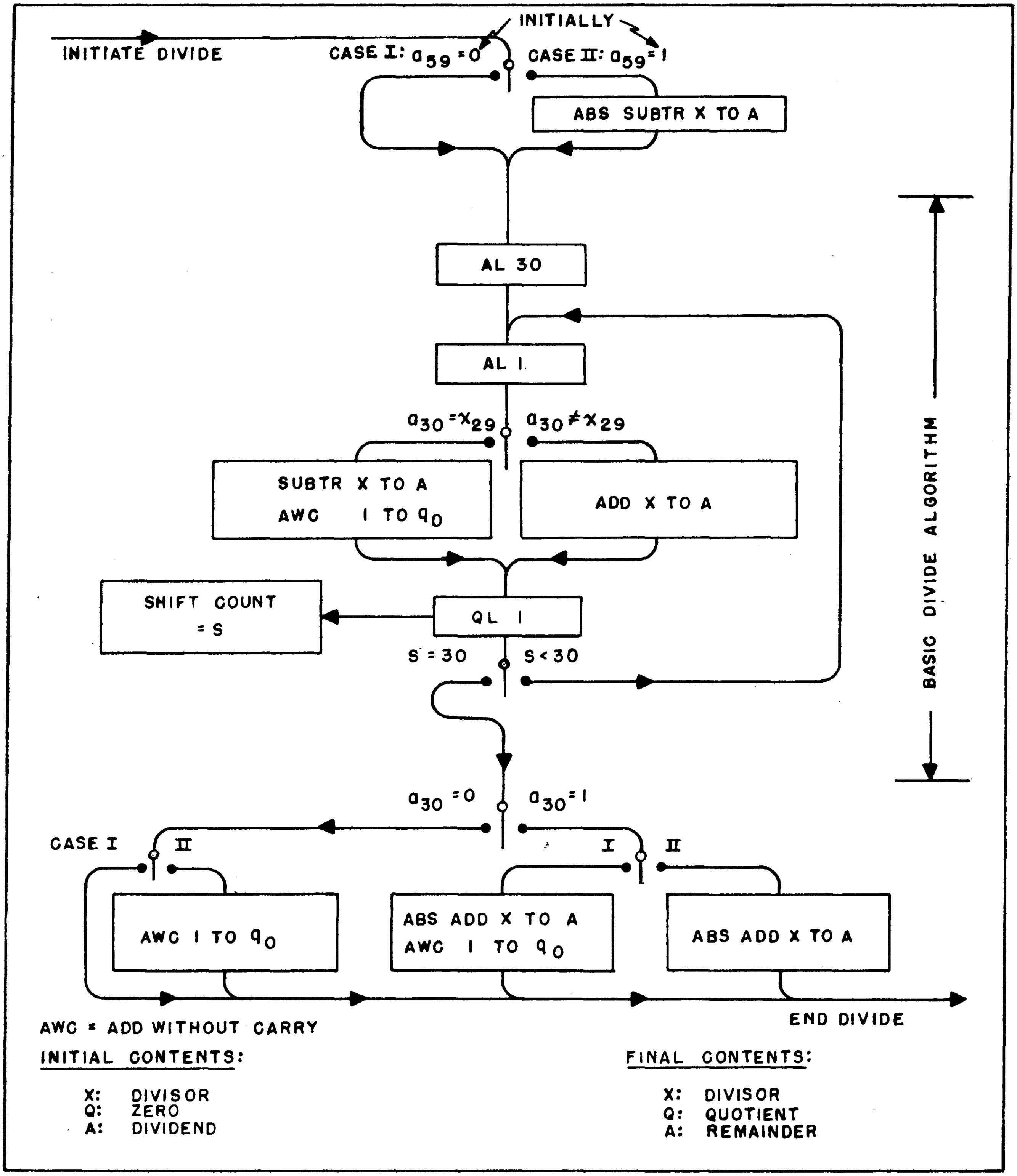


Figure 5. DIVIDE ALGORITHM (portion of DPpy)

TABLE I

TENTATIVE PHYSICAL CHARACTERISTICS OF STORAGE DRUM

Number of storage tracks	60
Number of locating or address tracks	11
Number of control or timing tracks	2
Digits per inch of track	80
Tracks per axial inch of drum	8
Diameter of drum, inches	8.5
Length of drum, inches	10
Period of revolution, nominal, milliseconds	16
Surface speed, inches per second	1600
Scanning rate, digits per second	128,000
Number of magnetic heads	73

TABLE II

NUMBERS SUBTRACTED INTO ACCUMULATOR IN THE BASIC ARITHMETIC OPERATIONS

Operation	Accumulator Digits $a_{59}a_{58}\cdots a_{30}a_{29}a_{28}\cdots a_1a_0$	Where Used
Add X to A	$x'_{29}x'_{29}\cdots x'_{29}x'_{29}x'_{28}\cdots x'_1x'_0$	OApy OPpy NPpy NMpy MApy MPpy DPpy
Subtract X to A	$x_{29}x_{29}\cdots x_{29}x_{29}x_{28}\cdots x_1x_0$	OSpy ONpy DPpy MApy MPpy
Absolute Add X to A		AApy APpy DPpy
If $x_{29} = 0$	$1\ 1\ \cdots 1\ 1\ x'_{28}\cdots x'_1x'_0$	
If $x_{29} = 1$	$1\ 1\ \cdots 1\ 1\ x_{28}\cdots x_1x_0$	
Absolute Subtract X to A		ASpy ANpy DPpy
If $x_{29} = 0$	$0\ 0\ \cdots 0\ 0\ x_{28}\cdots x_1x_0$	
If $x_{29} = 1$	$0\ 0\ \cdots 0\ 0\ x'_{28}\cdots x'_1x'_0$	
Split Add X to A	$1\ 1\ \cdots 1\ x'_{29}x'_{28}\cdots x'_1x'_0$	SApy SPpy
Split Subtract X to A	$0\ 0\ \cdots 0\ x_{29}x_{28}\cdots x_1x_0$	SSpy SNpy
Add Q to A	$q'_{29}q'_{29}\cdots q'_{29}q'_{29}q'_{28}\cdots q'_1q'_0$	QAp- QPp-

Table III. COMMANDS AND OPERATING TIMES

TABLE III. COMMANDS AND OPERATING TIMES (LIS

TITLE (DEFINITIONS IN SEC. 2.2)		CODE	MNEMONIC SIGNIFICANCE
1.	HOLD ADD	OApy	Ordinary Add
2.	CLEAR ADD	OPpy	Ordinary Positive
3.	HOLD SUBTRACT	OSpy	Ordinary Subtract
4.	CLEAR SUBTRACT	ONpy	Ordinary Negative
5.	ABSOLUTE HOLD ADD	AApy	Absolute Add
6.	ABSOLUTE CLEAR ADD	APpy	Absolute Positive
7.	ABSOLUTE HOLD SUBTRACT	ASpy	Absolute Subtract
8.	ABSOLUTE CLEAR SUBTRACT	ANpy	Absolute Negative
9.	SPLIT HOLD ADD	SApy	Split Add
10.	SPLIT CLEAR ADD	SPpy	Split Positive
11.	SPLIT HOLD SUBTRACT	SSpy	Split Subtract
12.	SPLIT CLEAR SUBTRACT	SNpy	Split Negative
13.	CLEAR ADD PLUS ONE	NPpy	Notch Plus
14.	CLEAR ADD MINUS ONE	NMpy	Notch Minus
15.	HOLD LOGICAL MULTIPLY	LApy	Logical Add
16.	CLEAR LOGICAL MULTIPLY	LPpy	Logical Positive
17.	SUBSTITUTE DIGITS	SDpy	
18.	SUBSTITUTE EXECUTION ADDRESS	SEpy	
19.	STORE A	AYpy	A to memorY
20.	FILL Q	YQpy	memorY to Q
21.	TRANSMIT A TO Q	AQp-	A to Q
22.	HOLD ADD FROM Q	QAp-	Q to A, Add
23.	CLEAR ADD FROM Q	QPp-	Q to A, Positive
24.	STORE Q	QYpy	Q to memorY
25.	SHIFT A LEFT	ALpk	
26.	SHIFT Q LEFT	QLpk	
27.	HOLD MULTIPLY	MApy	Multiply Add
28.	CLEAR MULTIPLY	MPpy	Multiply Positive
29.	DIVIDE	DPpy	Divide, Positive rem.
30.	SCALE FACTOR SHIFT	SFpy	
31.	PRESET END POINT COUNTER	PEpt	
32.	TEST END POINT	TEpy	
33.	TEST FULL ACCUMULATOR	TFpy	
34.	TEST HALF OVERFLOW	THpy	
35.	TEST SIGN	TSpy	
36.	PRINT ONLY	POpy	
37.	PRINT AND PUNCH	PPpy	
38.	INTERMEDIATE STOP	ISp-	
39.	FINAL STOP	FS--	

LE III. COMMANDS AND OPERATING TIMES (LIST E-4)

CODE	MNEMONIC SIGNIFICANCE	MINIMUM ALLOWABLE TIME INTERVALS, IN CELL-PERIODS (SEE NOTE 1)			APPLICABLE NOTES
		$C_{p'y}$	C_{yp}	$C_{p'p}$	
OApy	Ordinary Add	8	8	-	
OPpy	Ordinary Positive	8	8	-	
OSpy	Ordinary Subtract	8	8	-	
ONpy	Ordinary Negative	8	8	-	
AApy	Absolute Add	8	8	-	
APpy	Absolute Positive	8	8	-	
ASpy	Absolute Subtract	8	8	-	
ANpy	Absolute Negative	8	8	-	
SApy	Split Add	8	8	-	
SPpy	Split Positive	8	8	-	
SSpy	Split Subtract	8	8	-	
SNpy	Split Negative	8	8	-	
NPpy	Notch Plus	8	8	-	
NMpy	Notch Minus	8	8	-	
LApy	Logical Add	8	8	-	
LPpy	Logical Positive	8	8	-	
SDpy		8	32	-	2
SEpy		8	32	-	2
AYpy	A to memorY	8	32	-	2
YQpy	memorY to Q	8	8	-	
AQp-	A to Q	-	-	8	
QAp-	Q to A, Add	-	-	8	
QPp-	Q to A, Positive	-	-	8	
QYpy	Q to memorY	8	32	-	2
ALpk		-	-	8,16,24	3
QLpk		-	-	8,16	3
MApy	Multiply Add	8	48	-	
MPpy	Multiply Positive	8	40	-	
DPpy	Divide, Positive rem.	8	56	-	
SFpy		24	32	-	2
PEpt		-	-	8	
TEpy		-	-	8	
TFpy		-	-	8	
THpy		-	-	8	
TSpy		-	-	8	
POpy		8	8	-	4
PPpy		8	8	-	4
ISp-		-	-	-	
FS--		-	-	-	

Note 1

Note 2

Note 3

Note 4

MINIMUM ALLOWABLE TIME INTERVALS, IN CELL-PERIODS (SEE NOTE 1)			APPLICABLE NOTES
$C_{p'y}$	C_{yp}	$C_{p'p}$	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	8	-	
8	32	-	2
8	32	-	2
8	32	-	2
8	8	-	
-	-	8	
-	-	8	
-	-	8	
8	32	-	2
-	-	8,16,24	3
-	-	8,16	3
8	48	-	
8	40	-	
8	56	-	
24	32	-	2
-	-	8	
-	-	8	
-	-	8	
-	-	8	
-	-	8	
8	8	-	4
8	8	-	4
-	-	-	
-	-	-	

Note 1:

One cell-period is approximately 8 microseconds.

Note 2:

The minimum allowable time interval between <u>two</u> storage writing operations is 256 cell-periods. Should a shorter interval be programmed, the interval will automatically be lengthened by the duration of a complete drum revolution, or 16 milliseconds. For a <u>single</u> storage writing operation, only the time intervals tabulated here need be provided.

Note 3:

For shifts of 1 to 20 places, $C_{p'p}$ is 8; for 21 to 40 places, 16; for 41 to 60 places, 24.

Note 4:

The minimum allowable time interval between <u>two</u> print/punch operations is of the order of 125 milliseconds. Should a shorter interval be programmed, it will automatically be lengthened to this value. For a <u>single</u> print/punch operation, only the time intervals tabulated here need be provided.

APPENDIX A

LIST OF REFERENCES

a. Letter, A. A. Cohen of ERA to S. N. Alexander of NBS (13 August 1948).

b. Letter, A. A. Cohen of ERA to H. D. Huskey of NBS (30 September 1948).

c. Letter, A. A. Cohen of ERA to H. D. Huskey of NBS (19 October 1948).

d. Report, "Selective Alteration of Digital Data in a Magnetic Drum Computer Memory", Office of Naval Research contract N6onr-240, Engineering Research Associates, Inc. (1 December 1947).

e. I.R.E. National Convention paper, A. A. Cohen and W. R. Keye, same title as Reference (d), (25 March 1948). Abstracted in Proc. I.R.E. 36, 379 (1948). Printed transcripts of this paper are available from ERA on request.

f. I.R.E. National Convention paper, N. Rochester and D. R. Brown, "Rectifier Networks for Multiposition Switching " (25 March 1948). Abstracted in Proc. I.R.E. 36, 379 (1948).